CREATING TRAINING DATA FOR SURROGATE MODELS USING FE DRAPING SIMULATION

S. Keller[1], F. Maier[1], P. M. Blies[2] and R. M. Hinterhoelzl[1]

[1] Lightweight Design and Composite Materials, University of Applied Sciences Upper Austria, Stelzhamerstraße 23, 4600 Wels, Austria, sophia.keller@fh-wels.at
[2] enliteAI GmbH, Wollzeile 24/12, 1010 Vienna, p.blies@enlite.ai

Keywords: Surrogate modelling, Process simulation, Fabric draping, Automated Quality Assessment, Training data

ABSTRACT

In the last years, Finite Element (FE) simulations became increasingly important for the development and optimization of manufacturing processes. They enable a reduction of trial-and-error experiments in the development phase, saving costs and time. Despite the benefits, FE simulations, especially for fiber reinforced composites, can take a considerable amount of time to compute. A promising approach to address this issue is the approximation of the process behavior, using computationally inexpensive, artificial-intelligence-based surrogate models. When training such models with sufficient training data, they can predict the outcome for a given input accurately. However, a common issue for this approach is the lack of training data. Therefore, this work focuses on the efficient generation of training data for a surrogate model, using FE analysis. Towards this goal, we developed an automated routine for data generation. A Python script generates input files, post-processes the output files and exports relevant data. The underlying parametrized draping simulation model is called with different input parameters, to simulate various draping cases. This approach can generate training data sets, covering around 200 different parameter variations, within reasonable time. To evaluate the suitability of the FE results as training data, we trained a U-Net based surrogate model. The resulting surrogate model predictions were compared with the respective simulation results. Additionally, to determine the minimum amount of training data necessary for the surrogate model to predict results correctly, we conducted a parametric study with varying training set sizes.

1 INTRODUCTION

Due to their outstanding potential, fiber reinforced composites (FRC) find more and more applications in various fields, such as the aerospace, automotive and sports industry. A large amount of FRC components, however, are still manufactured manually, requiring experienced workers to achieve sufficient quality. Especially in the aerospace and automotive industry, with their high demand, manual manufacturing reaches its limits. Towards these needs, automatization of the process becomes inevitable. An important part to develop automated processes (being either robot-assisted or fully automated) is the process simulation. A common method for simulating manufacturing processes is the use of numerical methods, such as the Finite Element Method (FEM). That way, accurate predictions of the process result, for a given parameter set, can be realized. Despite increasing computational power, simulations can require several days, especially for the preforming process of slack fabrics (as it is often the case for FRC) [1]. With the simulation being that time-consuming, virtual optimization of the process can become quite inefficient. Thus, the simulation forms the bottleneck for the development process.

A promising approach to resolve this issue is to create machine learning (ML) based surrogate models. The idea behind surrogate modeling is to replace the time-consuming FE simulation model by a computationally inexpensive surrogate, which approximates the original simulation model. The surrogate model then enables a computationally efficient optimization of manufacturing process parameters. In the field of metal forming, there are already various examples in the literature where ML-surrogate models are used for an effective process optimization (see e.g. [2] and [3]). In recent studies, surrogate modeling is also used for textile forming processes ([4-6]), opening new possibilities to advance the composite industry by enhancing the efficiency of the parameter optimization.

For a machine learning model to accurately predict the outcome for a given input, it must be trained with enough data. A common problem encountered when training a surrogate model is the lack of sufficient training data. The objective of this study is to efficiently create FE training data for surrogate models. Therefore, we modeled a fabric draping process in the commercial software package Abaqus/CAE. Using the Python API of Abaqus, we created a Python script containing a parametrized version of the model. Then, we built a routine around the scripted model, which enables the automated generation of training data. The generated data was then used to train a U-Net based surrogate model, to determine the suitability of the chosen parameter variations. Additionally, we conducted a parametric study to determine the amount of training data that is necessary to correctly predict the simulation result for a given parameter set.

2 METHODS

To create FE training data for a surrogate model in reasonable time, two preconditions must be met: (1) the FE model must be parametrizable, and (2) the computation time of one simulation run must be as low as possible (ideally in the range of seconds to a few minutes). Additionally, if the surrogate model should not only recreate the result from a given input, but also find the optimal processing window, a metric for an automated quality assessment of the simulation result must be defined. The following subsections show a detailed description of the FE model and the surrogate model, as well as the architecture of the training data and the chosen metric for the automated draping quality assessment.

2.1 Material

For dynamic, explicit simulations (required for the draping process), the stable time increment Δt_{stable} defines the computation time. A larger stable time increment results in a reduced time to obtain the simulation result. The stable time increment (Eq. (1)) is defined by the smallest element length L^e of the FE mesh and divided by the wave speed c_d of the material. The element length is a purely geometric parameter; however, the wave speed depends on the Young's modulus E and the mass density ρ of the material (Eq. (2)).

$$\Delta t_{stable} = \frac{L^e}{c_d}, \tag{1}$$

$$c_d = \sqrt{\frac{E}{\rho}}. \tag{2}$$

A smaller ratio of Young's modulus and mass density results in a larger stable time increment and faster computation time. Therefore, the material (*Interglas 92125* 2x2 twill woven fabric) was chosen based on the ratio value of E and ρ. To generate initial training data (computationally efficient), material properties were estimated (Table 1).

$E_1 = E_2$ $[MPa]$	E_3 $[MPa]$	$\nu_{12} = \nu_{13}$ $[-]$	ν_{23} $[-]$	$G_{12} = G_{13}$ $[MPa]$	G_{23} $[MPa]$	ρ $[ton/mm^3]$
520	50	0.3	0.42	67	30	$1.95 \cdot 10^{-7}$

Table 1: Material parameters of the artificial (computationally efficient) material

These material properties were derived from mechanical properties of the S2-Glass-Epoxy2 (see [7]) and slightly modified. The modified S2-Glass-Epoxy2 estimates the draping behavior of the woven fabric, that will be implemented for the future calculations. Modifications include reducing the Young's moduli as well as the shear moduli. The density remained unchanged.

2.2 FE Model setup

For the finite element simulations, the draping process is modeled using the commercial software package Abaqus/Explicit. We created two models of the draping process: the baseline model and a parametrized model. The baseline model (high level of detail, time-expensive) includes all relevant process properties, while the parametrized model is a numerically efficient, simplified version of the baseline model.

Baseline Model. Figure 1 shows the model setup for the baseline model as it was created in Abaqus/CAE. It contains all relevant process parameters to predict the draping result accurately. The customized blank cut single layer (see Figure 1b) is lowered onto a rib tool (see Figure 1c). To hold and manipulate the blank cut, six clamps are attached to the lamina. The clamps are modeled by connecting the lamina nodes affected by a clamp with a reference point using a kinematic coupling constraint (clamping areas are shown in Figure 1b). The reference point is connected to the frame (modeled as coupled reference points) using SPRINGA elements with a spring stiffness of 0.6 N/mm. The draping procedure is modeled as a one-step process, where the fabric is initially lowered onto the tool and then draped into shape, using a uniform pressure load, on the top surface of the lamina.

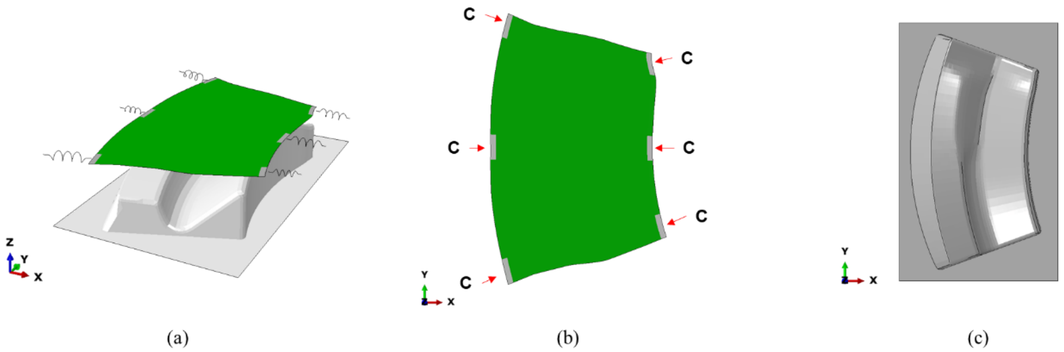


Figure 1: (a) Isometric view of the draping process model and top view of the (b) blank cut with highlighted clamping areas denoted with “C” and the (c) modified rib tool

The tool is meshed using 2840 elements of the type R3D3 and R3D4 (see Figure 2a) while the blank cut is meshed using 5307 elements of the type S4 (see Figure 2b). To simplify the localization of distinct areas on the blank cut (such as clamping areas and pressure areas), we structured the mesh with an equal number of elements on opposite sides.

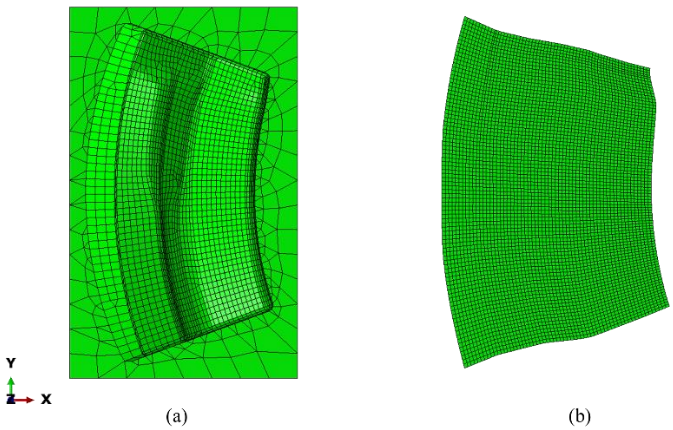


Figure 2: FE mesh of the (a) tool using R3D3 and R3D4 elements and (b) the blank cut using S4 elements

The estimated material was modeled in Abaqus using an orthotopic material model with the engineering constants specified in Table 1. Between tool and blank cut, a general contact with a friction coefficient of 0.3 is assumed.

Parametrized Model. The parametrized model is a scripted version of the baseline model, where predefined parameters can be varied within a range to receive the training data set. Within the script, the meshed tool and blank cut are imported as an orphan mesh, to assure identical meshes and thus, nodal positions, for each simulation run. For the parametrization, the magnitude F_i of the uniform loads per section i as well as the size of the section (defined by division lines y_n) on which the loads are applied, are defined as variable parameters (see Figure 3). The script can be executed in the Abaqus command prompt window, enabling the user to generate input files without opening the graphical user interface (GUI) of Abaqus/CAE. Since the material data is already computationally efficient, the parametrized model is merely a scripted clone of the baseline model, which is able to automatically create Abaqus input files with varying values for the specified parameters.

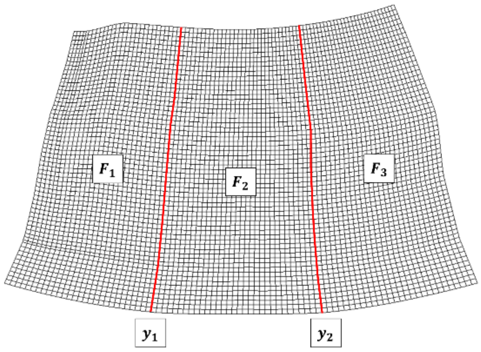


Figure 3: Variable parameters for the parametrized model

2.3 Automated Draping Quality Assessment

To supply the surrogate model with a complete training data set to learn from, the quality of each simulation results must be assessed automatically. Therefore, a metric for measuring the draping quality is needed. On a macroscopic scale, the quality of a draping result is mainly determined by the fiber orientation and the number and type of defects that occur. Besides fiber misalignment, possible geometrical draping defects are wrinkles, folds, bridging and cracks. One common feature of wrinkles, folds and bridging is the lack of contact with the tool. In Abaqus the *COPEN* parameter determines if the nodes of two collision objects (i.e., in our case the blank cut and the tool) are in contact. The ratio between the nodes in contact and all nodes provides a value between 0 and 1 which indicates the quality of the simulation result (with 0 being the worst and 1 being the best result). For the proof-of-concept of the surrogate model, this simple assessment is sufficient. However, for further studies, the quality assessment will be extended by instance segmentation (to differentiate between folds, wrinkles and bridging) and by considering possible fiber misalignment.

2.4 Training Data Generation

Automated Training Data Generation. First, all input files for a training data set are created using the main python script. Within that python script, the parametrized model is modified using a for-loop, to iterate through different combinations of the individual parameter values. For each parameter combination, the parametrized model is executed to create an input file. The input file names are then inserted into a batch file. By executing that batch file, the training data set is generated by executing each simulation using the CPU parallelization (with a predefined number of CPUs). The batch file repeats two steps in a for loop: first, the simulation is calculated; second, the generated output file (in the closed file format *.odb*) is opened and relevant output data is exported to a *.csv* file for further processing. For large scale training data, the main python script is also able to assign input files to multiple computers. In that case, a batch file is created for each computer and calculations are divided equally between them. Thus, the parallelization is only limited by the available machines and software licenses.

Training Data Architecture. The *.csv* files, which are exported from the simulation results, contain the input for the U-Net surrogate model. All variables defined in the *Field Output Request* of the simulation can be exported to the *.csv* file. In the case of the initial training data sets presented for this paper, we only used the *COPEN* parameter and the stresses in the local 1-direction *S11*.

Training Data pressure magnitude. For the training of the surrogate model, we created two different data sets. For the first training data set, the variable parameters are the pressure magnitudes. The blank cut is divided into three approximately equal-sized areas (c.f. A1, A2 and A3 in Figure 4), on which the pressure magnitudes of each area varied between 0.01 and $0.11\,N/mm^2$ in increments of $0.02\,N/mm^2$. Considering all combination of pressure loads, this leads to a data set containing 216 simulation runs.

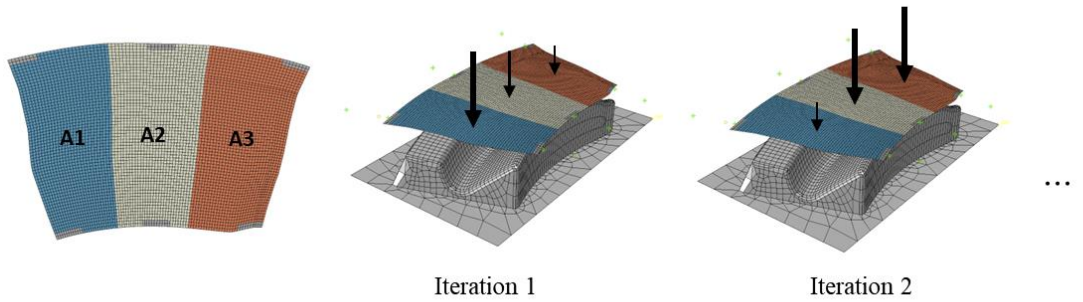


Figure 4: Variations of the model parameters for creating the training data set *pressure magnitude*

Training Data patch size. The variable parameters for the second training data set are the patch sizes, on which the pressure was applied. The pressure magnitudes remained constant at values of $F_1 = 0.05\,N/mm^2$, $F_2 = 0.01\,N/mm^2$ and $F_3 = 0.05\,N/mm^2$. The patch sizes were varied by moving both division lines of the three patches along the y-direction (Figure 5). Since the mesh is structured, the elements could be arranged in a matrix, in which the element numbers match with their position relative to their neighboring elements. Thereby, patches could be divided with quasi-continuous lines, by defining the columns in the matrix, at which the patches should be divided. In y-direction, the mesh has 87 elements. Therefore, the first division line was within the range of 15 to 56 increasing by increments of 1, and division line 2 varied between 30 and 70, increasing by increments of 1, with a minimum distance of 15 elements. This parametrization results in a total of 198 simulation runs.

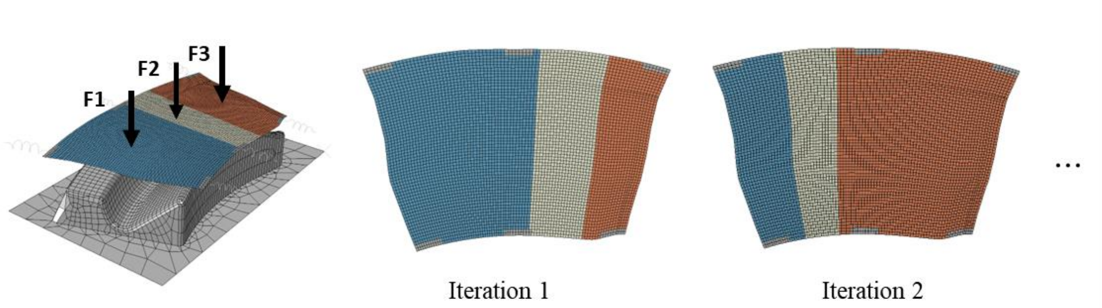


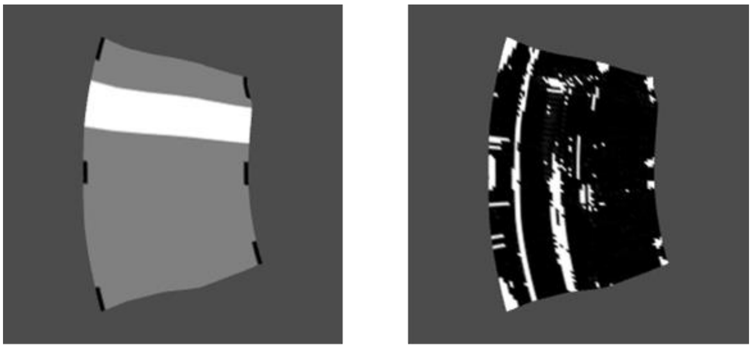
Figure 5: Variations of the model parameters for creating training data set *patch size*

2.5 Surrogate Model

Derived from the FE model setup, the problem that we want to solve can be stated as: given the particular state s of the fabric, find the action a that produces the best draping result. For our purposes, "best" means the draping result with the smallest global *COPEN* value. This problem can be formulated as a Markov decision process (MDP). One component of an MDP are the transition probabilities $p(s'|s,a)$, which determine how the state of the system changes when a particular action is applied. In our problem, the true transition probabilities are calculated from the FEM simulations, telling us exactly how the fabric behaves, when specific forces are applied at specific locations. Since FEM is computationally expensive, this is the part that we would like to replace with a surrogate model. In other words, the transition probabilities $p(s'|s,a)$ should be calculated with the surrogate model. In this case, the transition probabilities are deterministic, so the form $p(s'|s,a)$ can actually be written in the simpler form $s' = f(s,a)$. Now, since we have a 1-step MDP, the initial state is the same for all actions, resulting in a further simplification of $s' = f(a)$. So we need to learn the mapping f from given actions a and resulting states s'. This can be cast as a simple supervised learning problem.

We investigated several approaches for the class of surrogate model. In this paper, we present the results for an approach that is common for segmentation tasks: a U-Net. It was originally developed for biomedical image segmentation tasks by [8]. We will not provide too many details on this architecture itself since it is widely used in literature. The input to a U-Net is an image, which means that we have to encode the state and action of the tool-fabric system into an image. We will demonstrate the process for training data set *patch size*. For the other training data set, the process is analogous.

Encoding Input Data. The post-processing script for the FEM simulation provides numerical input data in a .csv file. To be able to use a U-Net, the input data (in our case just the action a) as well as the resulting state must be encoded as image data. For both sets of training data the pressure magnitudes are encoded as single-channel color values on top of the fabric geometry (Figure 6a). The target (*S11* or *COPEN*) is also encoded as single-channel color value on top of the geometry (Figure 6b).



(a) (b)

Figure 6: Exemplary encoded single-channel color values on the top surface of the lamina for the (a) pressure magnitudes and the (b) target variable

Training the Surrogate Model. The training procedure of the U-Net is straightforward. We feed the encoded action images into the U-Net and use the *COPEN* as well as the *S11* images as targets. As loss function, we use a custom cross entropy loss adopted for regression tasks. The implemented optimizer uses the stochastic gradient descent (SGD) with a learning rate of 0.001. Since the focus of this paper is the data generation process, we are interested in finding out how much training data this problem needs at a minimum until the surrogate model can reproduce the behavior of the FEM simulation to a sufficient degree. Therefore, we conducted a parametric study on training data set *pressure magnitude*. For the parametric study we have used the following training setup for 20 different seeds:

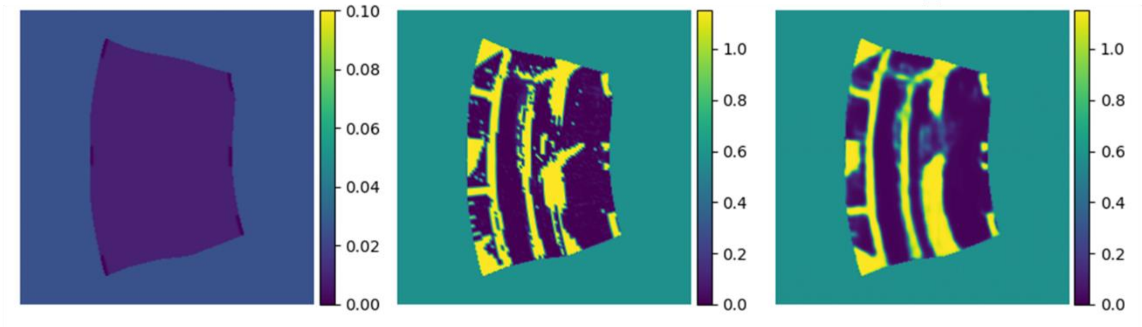
1. Perform a (seeded) random 80/20 train/validation split
2. Train a new model on a subset of the training set (2%, 4%, 8%, 16%, 32%, 64% and 100% of the training data respectively)
3. Evaluate the trained model on the complete validation set with a simple image similarity metric (in this case with the Euclidean norm between FE results and surrogate model prediction)

This way, we get an estimate of how much training data is needed to achieve a sufficient performance on the validation set. Of course, it is expected that the performance of the surrogate model increases with an increasing training data set size. To amount for the limited number of training data, it is of interest to see if we can cut down on the number of simulations needed.

3 RESULTS

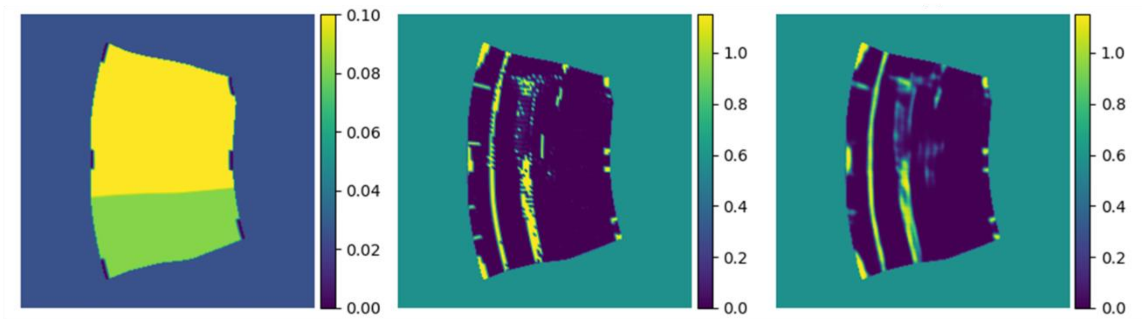
3.1 Comparison of FE and Surrogate Model Results

Results Training Data Set pressure magnitude. With a runtime per simulation of about 3.2 minutes, the first training data set (containing 216 simulation results) was created within 11.5 hours. Before training the surrogate model, we manually checked whether there was enough variation of the *S11* and *COPEN* values in the simulation results, throughout the entire data set. Then, the worst and the best draping result (according to the *COPEN* parameter), i.e., the draping results with the least and most nodes in contact, were determined. The parameters leading to the worst result were $F1 = 0.01\ N/mm^2$, $F2 = 0.01\ N/mm^2$, $F3 = 0.01\ N/mm^2$, while the parameters leading to the best result were $F1 = 0.11\ N/mm^2$, $F2 = 0.11\ N/mm^2$, $F3 = 0.09\ N/mm^2$. Figures 7 and 8 show a comparison of the FE and surrogate model result for the worst and best parameter combination.



(a) (b) (c)

Figure 7: (a) Encoded load case for the worst draping result according to *COPEN* values and the respective (b) FE result and (c) surrogate model prediction



(a) (b) (c)

Figure 8: (a) Encoded load case for the best draping result according to *COPEN* values and the respective (b) FE result and (c) surrogate model prediction

For Figures 7a and 8a, the continuous color map from blue to yellow displays the different pressure magnitudes acting on the patches. The color map for Figures 7 and 8 (b) and (c) displays the distance of the lamina nodes to the tool, with blue being no distance (i.e., in contact) and yellow being maximum distance (i.e., not in contact). It can be seen that the surrogate model prediction of both load cases matches the FE results quite well. Some small yellow areas (nodes not in contact) in the FE results do not appear in the surrogate model prediction. This effect is slightly stronger for the worst draping result (Figure 7b and c), which means the surrogate model has more difficulties to predict poor draping results in detail. In general, however, there is good compliance between FE results and surrogate model prediction.

Results Training Data Set patch size. The second training data (containing 198 simulations) was created within 10.5 hours. Similar to the procedure for the first training data, the set was checked on variations of the *S11* and *COPEN* parameter. We found that there was less variability compared to varying the pressure magnitude. Nevertheless, we trained the surrogate model to explore the surrogate quality, for less variable training data. The worst patch divisions (according to *COPEN*) are at element columns 23 and 38 while the best patch divisions are at element columns 21 and 71. The prediction and the FE simulation results for those parameters are shown in the following figures:

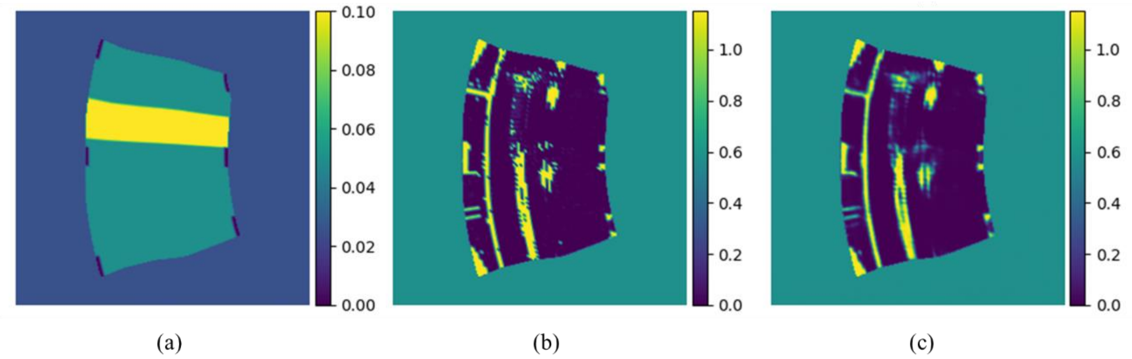


Figure 9: (a) Patch division for the worst draping result according to *COPEN* values and the respective (b) FE result and (c) surrogate model prediction

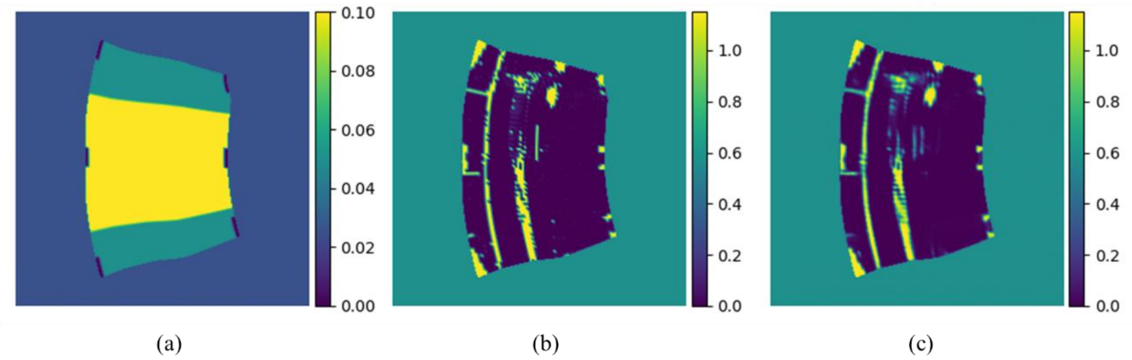


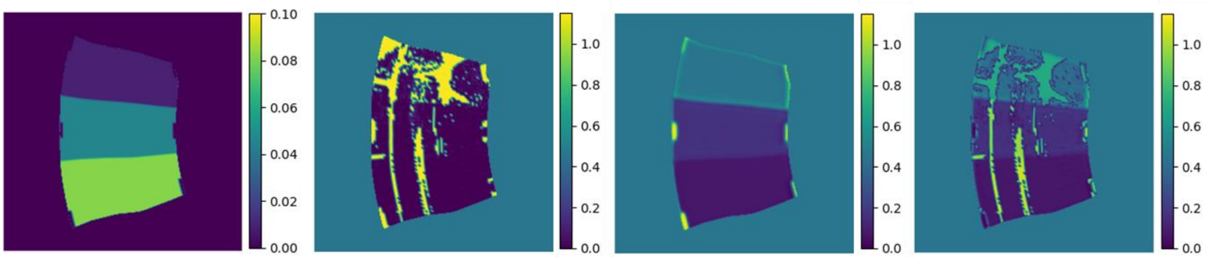
Figure 10: (a) Patch division for the best draping result according to *COPEN* values and the respective (b) FE result and (c) surrogate model prediction

The color code for Figures 9 and 10 is identical to the one of the first training data set. Again, the surrogate model prediction matches the FE result well. However, the *COPEN* images for the worst and best case look quite similar. Since the difference between best and worst result is small, the surrogate model has no problems with predicting the output for a certain patch division. Again, the best draping result is predicted slightly more precisely than the worst draping result.

3.2 Minimum Required Training Data

As the variation of the *COPEN* and *S11* values was greater within data set *pressure magnitude*, this data set was used for the parametric study. The minimum required training data was determined based on the *COPEN* and the *S11* values.

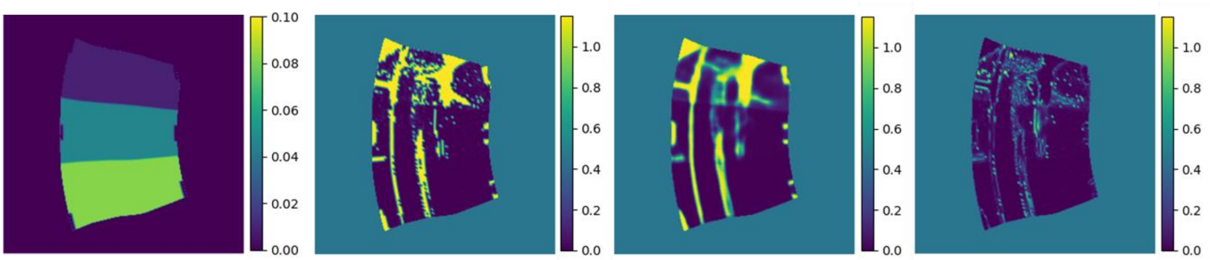
Parametric study on COPEN values. Figure 11 shows a comparison of the *COPEN* values calculated by the FE simulation (Figure 11b) and the *COPEN* values predicted by a surrogate model that has been trained on 2% of the training data (Figure 11c) for an exemplary load case (Figure 11a). It is apparent that the surrogate model cannot predict the *COPEN* values. The surrogate model prediction shows no variation with the individual lamina areas. Therefore, the difference between both results (which would ideally be zero across the entire fabric) shows major deviations in regions where the FE model calculated higher *COPEN* values (see Figure 11d) and the pattern of the surrogate model prediction can also be seen.



(a) (b) (c) (d)

Figure 11: (a) Exemplary load case and the respective (b) FE simulation result, (c) prediction of the surrogate model that is trained with 2% of the training data, and (d) difference between FE result and surrogate model prediction

As expected, the results are better when the entire available training data is used. Figure 12 shows the results for the same load combination predicted by a surrogate model that has been trained on 100% of the training data. The action space (Figure 12a) remained the same, thus, the FE result (Figure 12b) did not change. The prediction of the surrogate model, however, improved significantly and the prediction in Figure 12c looks quite similar to the FE result in Figure 12b. This is also reflected in the difference image (see Figure 12d), indicating a high similarity between the two results.



(a) (b) (c) (d)

Figure 12: (a) Exemplary load case and the respective (b) FE simulation result, (c) prediction of the surrogate model that is trained with 100% of the training data, and (d) difference between FE result and surrogate model prediction

Comparing the difference of the two results, highest deviations occur at the fringes of the areas which are not in contact with the tool. The surrogate model can therefore predict the locations with the largest *COPEN* values, with slight imprecisions at the fringes of these areas.

Minimum required training data. Naturally, the model trained on more data, yields better results. The blue curve in Figure 13 shows the average (over all seeds) difference of the *COPEN* images for different training set sizes. It is clearly visible that the average difference decreases with increasing training set size. While the error from 2% to 8% training set is quite high, there is a vast decrease in error between 8% and 16% of the training set size. A possible explanation for the decrease in the error is that all relevant modes are included in 16% of the training set, such that the generalization to the validation set is sufficient for this simple problem. Thus, a generation of 16% of the training set simulations (i.e., 28 simulations out of 216) is theoretically adequate to train a sufficiently good surrogate model for predicting the *COPEN* value. Reducing the training set would therefore reduce the time from 11.5 to 1.9 hours.

In order to analyze the learning behavior for a second quantity, the average difference of the *S11* images was also evaluated for different training set sizes (see orange curve in Figure 13). Similar to the curve for the *COPEN* images, the average difference for the *S11* values decreases with the increasing training set size. However, as opposed to the *COPEN* diagram, the overall improvement of the average difference for the *S11* values is less than for the *COPEN* values. We found that there was less variability of the *S11* data compared to the *COPEN* data. This explains, why the prediction quality for 2% of the training set is already quite high, and the improvement rate with increasing training set size is low.

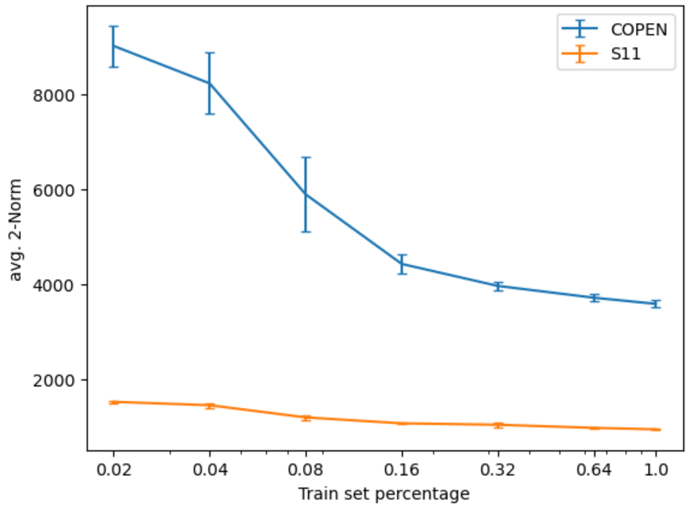


Figure 13: Average 2-norm values of the *COPEN* images (blue curve) and the *S11* images (orange curve) for different training set sizes of training data *pressure magnitudes*

4 DISCUSSION & CONCLUSIONS

Using the Python API within Abaqus, FE training data can be generated automatically and parallelized. Provided that the simulation setup allows for a fast computation of a single simulation, the created workflow can be applied to any simulation model, as long as a parametrizable version of the FE model is available. The applied U-Net surrogate model was successfully trained using the generated training data. Due to the rather simple design of surrogate model, a detailed quantitative evaluation of the model quality is not reasonable. However, for evaluating the suitability of the FE training data, a qualitative evaluation is sufficient. The qualitative comparison of different simulation results and surrogate model predictions showed that the FE data is applicable as training data. According to the parametric study on the training data set *pressure magnitude*, the initial number of training data could theoretically be reduced by 84%, saving a substantial amount of time without severe loss in prediction precision for the *COPEN* and *S11* value.

In a next step, the computationally efficient material properties will be replaced by the characterized material parameters for the *Interglas 92125 2x2 twill* woven fabric, used in the experiments. It can be expected that the characterized material will not be as computationally efficient. Therefore, different acceleration methods will be examined in a parametric study. These methods include different integration methods, modifications on the Young's modulus and the density, reducing the *Field Output* as well as using a step-time dependent amplitude for the applied gravity. Analyzing the *COPEN* parameter provides a first, but rather vague, assessment on the draping quality of the simulation result, since it only accounts for if nodes are in contact or not. Thus, the automatic quality assessment will be extended by a fiber angle evaluation and instance segmentation to differentiate between defects such as folds, wrinkles, and bridging. Since the generated FE training data was well suited for the simple U-Net surrogate model, the use of more advanced machine learning models will be investigated.

ACKNOWLEDGEMENTS

The authors want to thank the Austrian Research Promotion Agency (FFG) for the financial support of the project Jarvis4Pre (Grant No. 886871) in the scope of the "Take Off" Funding Program of the Republic of Austria, Ministry for Climate Action.

REFERENCES

[1] J. Pfrommer, C. Zimmerling, J. Liu, F. Henning and J. Beyerer, Optimisation of manufacturing process parameters using deep neural networks as surrogate models, *Proceedings of the 51st CIRP Conference on Manufacturing Systems*, Stockholm, Sweden, Vol. 72, 2018, pp. 426-431 (doi: 10.5445/IR/1000084778).

[2] H. Zhou, Q. Xu and N. Li, A study on using image based machine learning methods to develop the surrogate models of stamp forming simulations, *Journal of Manufacturing Science and Engineering*, Vol. 144, Issue 2, 2020.

[3] S. Wessing, G. Rudolph, S. Turck, C. Klimmek, S.C. Schäfer, M. Schneider and U. Lehmann, Replacing FEA for sheet metal forming by surrogate modelling, *Cogent Engineering*, Vol. 1, Issue 1, 2014 (doi: 10.1080/23311916.2014.950853).

[4] C. Zimmerling, P. Schindler, J. Seuffert and L. Kärger, Deep neural networks as surrogate models for time-efficient manufacturing process optimisation, *Proceedings of ESAFORM 2021, MS11 (Optimization & Inverse)*, Liège, Belgium, 2021.

[5] C. Zimmerling, D. Dörr, F. Henning and L. Kärger, A machine learning assisted approach for textile formability assessment and design improvement of composite components, *Composites Part A: Applied Science and Manufacturing, Volume 124*, 2019 (doi: https://doi.org/10.1016/j.compositesa.2019.05.027).

[6] C. Zimmerling, C. Poppe and L. Kärger, Estimating optimum process parameters in textile draping of variable part geometries - A reinforcement learning approach, *Procedia manufacturing, Proceedings of the 23rd International Conference on Material Forming*, ESAFORM 2020, Cottbus, Germany, Elsevir, 2019, pp. 426–431.

[7] A. K. Pickett. *Process and mechanical modelling of engineering composites*, IFB, University of Stuttgart, 2018, pp. 263-271.

[8] O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation". *Computer Vision and Pattern Recognition*, 2015, pp. 234-241 (doi: https://doi.org/10.48550/arXiv.1505.04597).