Synthetic Data Generation for Deep Learning Models

Christoph Petroll[1,2], Martin Denk[2], Jens Holtmannspötter[1,2], Kristin Paetzold[3], Philipp Höfer[2]

[1] The Bundeswehr Research Institute for Materials, Fuels and Lubricants (WIWeB)
[2] Universität der Bundeswehr München (UniBwM)
[3] Technische Universität Dresden

** Korrespondierender Autor:*
Christoph Petroll
Institutsweg 1
85435 Erding
Germany
Telephone: 08122/9590 3313
Mail: christophpetroll@bundeswehr.org

Abstract

The design freedom and functional integration of additive manufacturing is increasingly being implemented in existing products. One of the biggest challenges are competing optimization goals and functions. This leads to multidisciplinary optimization problems which needs to be solved in parallel. To solve this problem, the authors require a synthetic data set to train a deep learning metamodel. The research presented shows how to create a data set with the right quality and quantity. It is discussed what are the requirements for solving an MDO problem with a metamodel taking into account functional and production-specific boundary conditions. A data set of generic designs is then generated and validated. The generation of the generic design proposals is accompanied by a specific product development example of a drone combustion engine.

Keywords

Multidisciplinary Optimization Problem, Synthetic Data, Deep Learning

 | DOI: https://doi.org/10.35199/dfx2021.11

1. Introduction and Idea of This Research

Due to its great design freedom, additive manufacturing (AM) shows a high potential of functional integration and part consolidation [1]-[3]. For this purpose, functions and optimization goals that are usually fulfilled by individual components must be considered in parallel. This leads to multidisciplinary optimization problems (MDO). One way to counter this is direct mathematical coupling or an iterative mathematical overall structure of such problems [4][5] This approach requires difficult algorithms and complex numerical development processes [6]. In connection with complicated boundary conditions in form of functions, this can lead to non-generic design processes with unsystematic optimization goals [7][8]. In reality, this complexity is encountered by iterative cycling during the different product development phases. This can entail long development cycles, incomplete part design optimization, and represents a correspondingly great challenge to use the design freedom of AM [2][9].

In this work the authors present the design basis for an approach to meet this challenge of MDO and AM constraints. The goal is to summarize all requirements from physics optimization, production constraints and function integration in a design proposal for a deep learning metamodel. In contrast to mathematically complex and restricted existing metamodel approaches [10][11], an indirect coupling via a deep learning model is examined. An overview of the structure of the entire approach is shown in Figure 1.The described process is developed and applied to a problem in which it is a matter of developing a new design of an existing engine mount of a drone internal combustion engine. Among other requirements, this holder has to thermally dissipate the heat of the engine, be structurally stable and offer a low aerodynamic resistance. In addition, the structure geometry should be able to be manufactured additively as one single component.

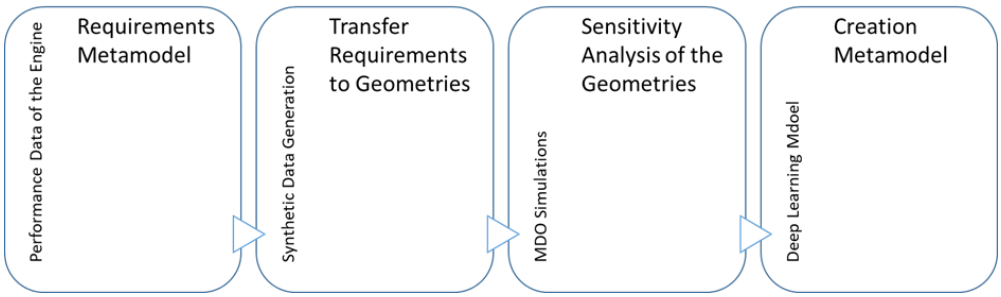

Figure 1: 4 Steps to Generate a MDO Metamodel

One of the main challenges for the approach is to generate a correspondingly large and qualitatively meaningful data set on the basis of which the metamodel can learn the wanted behavior. The generation of a corresponding number of synthetic design proposals as data sets is the subject of this research [Figure 1: Step 2].

2. State of the Art

The determination of “good data sets” and the preparation of the data represent an essential achievement when using deep learning models. The core issue here is to have the right quality and quantity. It is important that the wanted information and relationships from the data set can be transferred to the deep learning model. The consideration of how much effort is required for the creation of a certain model can essentially be measured by the question of how much usable data has already been observed or collected experimentally with the right information in it. For numerous special applications, it has therefore already proven to be useful to generate so-called synthetic. It is not actually collected data in order to train corresponding models [12][13]. The primary task of this data is to bring a certain flexibility and data richness into the

training of the model. Important are continuous, discrete or categorical (ordinal or non-ordinal) properties. Arbitrarily distributed data with a wide spread in the statistical distribution should be preferred. This has already been considered extensively for data records that can only be collected with great effort or that violate possible personal rights in large numbers [14].

Generative deep learning is one common field addressing this kind of problem using architectures such as autoencoders or generative adversial neural networks (GAN)[17]-[20]. The GAN architecture is constructed by using a generator and a discriminator. The discriminator tries to distinguish between the generated and the real data set. The generator is trained and monitored so that to a certain extent it can generate further real, similar data that the discriminator classifies in binary form. If the accuracy of the predictions between fake and real data is 50 percent, the generator is able to generate "real" data and the discriminator can no longer predict which data is real or synthetic. One research area which uses this ability, is the data augmentation research field. As mentioned earlier, there can be various difficulties in deploying datasets. Data augmentation tries to increase therefore the diversity of the training data set through realistic, random transformation of data. This means, for example, turning, scaling, rotating, translating or adding artificial noise to data. The use of such methods has proven to be a robust way of augmenting data without suffering great loss of accuracy when training neural networks. With respect to synthetic data, the use case from GAN is to generate additional data that can no longer be distinguished from the real data [19]. In addition to data replication, another well-known application of GAN is for example the generation of fake images or videos from a few sample images.

In the case of product development, the minimum amount of data required for many approaches is precisely the difficulty, the design of just a few different geometries as a basis is the challenge. Even the GAN or the data augmentation approach needs a start data set to provide more training data. But it is not only difficult to find geometries which fits to individual problems, it is even more difficult to connect geometries into functions or to optimize them further [16]. Usually, simple geometric features or only very reduced, smaller functional features are considered. This is regularly achieved by varying the geometry parameters. Using deep learning models, these parameters are then optimized as input, compared to performance data as output. The limitation remains the description of the geometry using parametric descriptions. The model learns just what you can describe as a geometry.

So far, it has been difficult to implement a first, free design draft of the “white sheet of paper” as required in the introduction, taking into account all the physical, production-specific and functional boundary conditions mentioned.

3. Generation of sufficiently consistent generic designs for deep learning

In order to implement the required parametric independence, designs are generated that have generative properties in relation to the physical and functional description of the engine mounting.

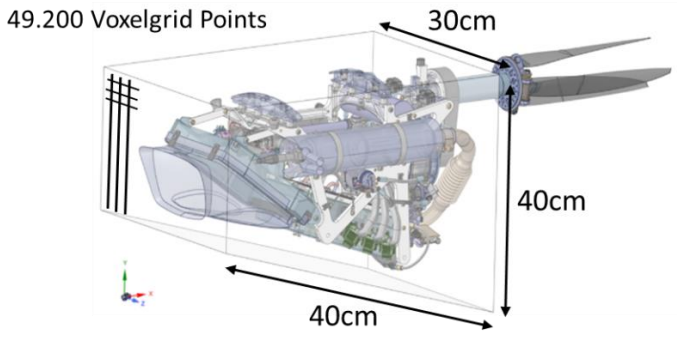

Figure 2: Drone Engine in the Design Space

First a design space is fixed and is defined by a corresponding number of points to be calculated. In view of the size of the engine unit and the currently available computing power and the data to be processed in bulk, this is set to 30x40x41cm [Figure 2].
Due to the intended goal of a first rough “white sheet of paper” draft, only one design point is set per cm, so that a material distribution for 49,200 points has to be calculated. For the calculation of the generic designs for the creation of a metamodel, the procedure is divided into 4 steps.

First, it is mathematically ensured that the repetition of mathematical patterns within the desired amount of data does not occur. In addition, it is guaranteed that the material distribution corresponds to bodies that represent uniform, organic geometries within the design space. Second, it is assured that the geometries are structurally connected and that air can flow through, as well as contain the connection points and interfaces for the engine and the drone structure. Third, without actively intervening in the design generation, designs will be preferred by 2 different criteria that have favorable manufacturing properties for metallic additive manufacturing. Finally, the entire procedure with regard to the use of design space, mesh capability and repeatability of the procedure is evaluated and validated.

3.1. Generation of functional material distribution with random noise

Using random functions, it is basically possible to generate a result that is similar but different from one another. Only a few parameters need to be adjusted for this. The challenges here are the generally continuous behavior and the possible avoidance of recurring patterns. One possibility for this is to apply noise to the function. For the description of a geometry, it is the goal that a so-called smooth noise is used. It means in the stochastic replacing individual data points with local averages of the surrounding data points. In addition, if viewed continuously over time, there is a relationship to the previous point within noise. The result is a “smooth”, steady curve. For our example, a smooth curve of the function means the natural, organic imitation of structures such as landscapes, water or bionicle optimized geometries. It can be described as a pseudo-random continuous function of the noise.

$$noise(x)\colon \mathbb{R}\rightarrow\mathbb{R} \qquad (1)$$

This form of noise is based on what is known as gradient noise. In contrast, there is discontinuous noise, such as white noise or value noise processes. Gradient based means the tangent slopes g_x at the discrete points $x \in \mathbb{Z}$.

$$g_x = grad(x) \in [-1,1] \qquad (2)$$

The discrete points P are generated as integer grid points at which the noise function $noise\ (x)$ has its zeros. This results in:

$$noise\ (x) = 0 \qquad noise(x)' = \frac{\partial\ noise(x)}{\partial\ x} = g_x \qquad (3)$$

A point P is generated using a hash function with different heuristic techniques, which are known from cryptology, but which will not be discussed further here. For our 3-dimensional example in this work, a hash value is generated for every $P(x, y, z, t)$ at a continuously point in time (t). The points are linked with an interpolation taking into account the local gradients. This leads to a 3D Perlin Noise which is used in the following. Therefore, a polynomial of the 4th degree is used as the classic interpolation function in this case. Ken Perlin invented the

Algorithm to make computer images look more natural. Instead of the interpretation of contour lines or images of fire, the fluctuations are interpreted as differences in density in this work [Figure 3]. Perlin uses several scaled versions of the same noise function in his algorithm. He uses a combination of frequency f and amplitude modulation AM with different frequencies and amplitudes $\hat{u}$.

$$noise_{AM}(x) = \sum_{i=0}^{M-1} \hat{u}_i * noise(f_i * x)$$
$$\text{with,} \quad \hat{u}_{i+1} = \hat{u}_i * \phi \tag{4}$$

ϕ is called persistence and represents a special constant that links the amplitude with the amplitude of the previous step. Finally, Perlin used several octaves to make it appear less sine-like. For this work this means that the adjustable parameters are the amplitude, the frequency, the persistence and the number of octaves. The ratio of frequency and amplitude ultimately determines the level of detail in the geometry. For this reason, the amplitude and frequency (noise scale) are then varied until the desired shape is available as the basis for the material distribution.

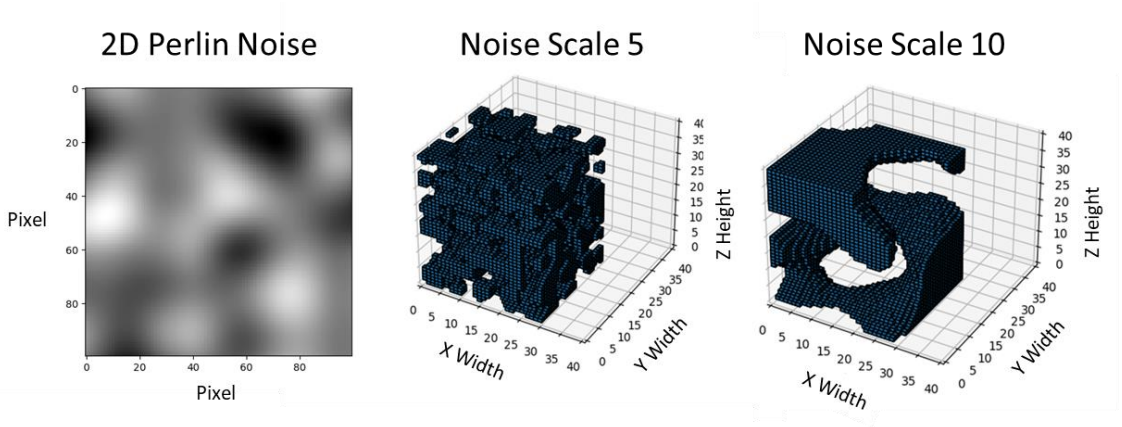

Figure 3: Perlin Noise Material Distribution with different Noise Scale from 2D to 3D

In a final step, the calculated points are then used as the center point for voxels in order to translate the points into a voxel-based geometry. This will later be used especially for convolutional neural networks, in which a fixed grid size reduces the computing power required for data processing. With a fixed grid size, the scanning functions used there can also be applied directly to the data.

3.2. Safeguarding physical and functional boundary conditions for simulations

Now that there is an even continuous, generic material distribution, the design space is trimmed according to the specifications for the engine mount. In order to ensure the inclination of the engine, for the correct angle of attack mounted on the drone, a corresponding area is declared as a non-design space using the coordinates of the voxels. In addition to the area to be kept free, interfaces are then also defined at which material must be available for each generic design. In the example, there are the attachment points of the motor and the screw points on the drone structure. In this way, any functionally required geometries for the operation of the engine can be added.

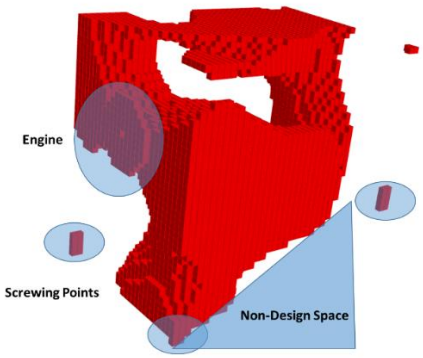

Figure 4: Example of one Voxel-Based Generic Design with Geometric Interfaces

For the basic material distribution that has now been adapted to the wanted engine mount, it now has to be ensured that static, thermal and aerodynamic evaluation of all generic designs are possible in the context of simulations. Therefore, for the static integrity, a continuously connected body has to be present without isolated voxels. For the static load case this means a connection of the screwing points to the engine, as well as a connection to the surface on the front of the engine holder. The engine brings thrust loads to the structure. On the front side, aerodynamic pressure leads to drag loads against the structure, depending on the inlet surface [

Figure 5]. In order to guarantee these connections, a search algorithm is used that is able to identify the isolated areas in three dimensions. The procedure used is based on the 2D image connected components algorithm presented by Rosenfeld and Pflatz. William Silversmith developed this process further for the 3-dimensional case [21]. The advantage of the algorithm is high efficiency and a good performance. After finding the separated areas, the heavy point of the largest connected area is determined as middle point. There the individual areas are connected via a 3-dimensional straight-line equation. The result are sustainable geometries in the desired directions of loading [

Figure 5].

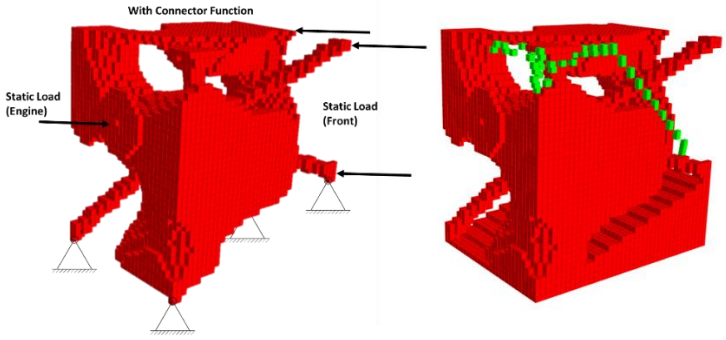

Figure 5: Visualization of the Static Load Case, the Connector Function and the Path Finding Algorithm

The next step now includes securing the flow of air through the engine bracket for later implementation of passive cooling of the engine. This is guaranteed by means of a path finding algorithm. In our example, the use of Dijkstra's algorithm turned out to be efficient. It calculates the shortest distance via a permanent start node and temporary nodes to the destination point. If the result of the search is positive, the generic design could be considered suitable for a fluid dynamic simulation.

This now resulted in all of the generic designs having the required geometry interfaces and physical properties. In order to further process, the generic design in voxel format, is transferred into an STL file using a marching cube algorithm. The voxel geometries are

converted into polygonal surface models for transfer to a simulation as volume body. In Order to avoid unwanted stress peaks in the transmission of forces and a higher aerodynamic resistance within the voxel's edges, the surface is smoothed. The voxel STL is scanned and a limit value is used to decide what is inside and what is outside the desired final geometry. The limit value can be understood as a percentage of how much overlap in the volume should or should not be included. Depending on the limit value, a smooth voxel structure is created. In the work used here, a 1-voxel range smoothing is performed in order to have no sharp 90° edges for the further simulations.

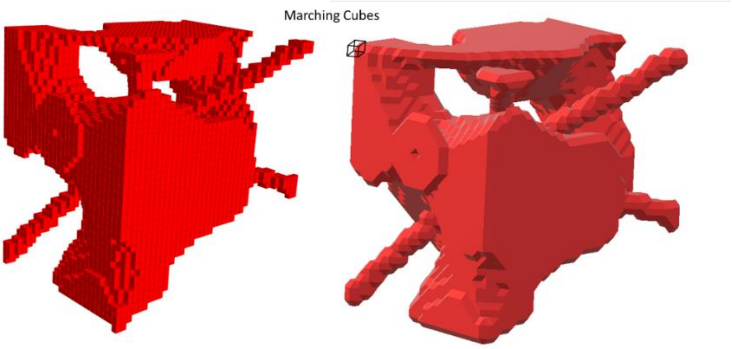

Figure 6: STL File Smoothed with Marching Cubes Algorithm

3.3. Implementation of mathematically independent manufacturing constraints

The integration of the manufacturing boundary conditions from additive manufacturing is deliberately drawn from the basic design creation and towards further design evaluation. This means that the selection of designs created by the Perlin Noise should not be influenced by a systematic criterion. The reason is that every recognition of a pattern in the generation of the basic material arrangement, except where it is intended, can lead to a restriction in the deep learning model. The model should decide afterwards through as diverse inputs as possible where material has to be in the sense of the learned specifications. Therefore, the restrictions of additive manufacturing are included in the process as kind of a preferred selection of fully generated designs afterwards. This means that a design is later transferred to the metamodel only according to the preferred manufacturing criteria.

One well known boundary condition that is considered in this work is the overhang angle of approx. 43 °. This angle is significantly limiting additive manufacturing for metal components through the use of massive support structure. This means the overhang of a structure in relation to the printing direction in relation to the X-Y plane. For a triangle of the generated STL files, this is calculated with the angel of the surface normal to the X-Y plane. For evaluation the figure of how many percent of the triangles within the generated geometry have an angle greater than 43° is used. The second manufacturing boundary condition within this thesis is the average energy jump per layer in the Z direction per design. Beside the need for support structure through overhang angles, the energy distribution is decisive. For this purpose, it is evaluated how many elements are exposed per layer in the Z-direction and how high the jump to the next layer is. This criterium can be better described as the uniformity of the structure. With these two properties for additive manufacturing the designs are now sorted by the program code. In this way, a data set is created that contains the preferred properties with regard to production for every generic design of the previous steps.

3.4. Validation of data consistency

For validation it is not only important that each individual design contains the required properties. In the context of a coherent data set, it must also be validated in total. The first thing that is evaluated is the spatial consistency of the entries across all designs. This means how often is each of the 49,200 calculation points used respectively how many times has each point played a role in a generic design. This is done with an arithmetic mean over the entries in the matrix of the voxel geometries. For example, the required geometric interfaces for screw-on points and the motor should, occur across the entire set of generic designs.

The automatic transferability of the design into a static simulation is also tested. This is tried with the use of the automatic mesh function in the simulation program itself. It is important that the meshes could be generated over all bodies of the generic designs with the same settings of the program automatically. Not only the mesh ability is verified, it is also tested if useful simulations can be performed with the generated geometries. Therefore, a single static, a thermal and an aerodynamic simulation is carried out.

4. Results and Discussion

The geometry generation, the use of the linking algorithm and the pathfinding algorithm showed no errors over 100,000 designs created. The code runs until the desired number of designs in terms of quality and quantity have been generated and takes less than 5 minutes on a standard workstation.

The integration of the manufacturing boundary conditions showed a distribution between 38% and 52% of the triangles per design with angles over 43°. Overall, based on the basic distribution of the Perlin Noise, it can probably be said that the angle alone only offers a limited suitability. In order to be more precise based on the angle it should be examined on functional surfaces or on enclosed surfaces.

The energetic entry or the change from layer to layer on average via design, promises a better suitability as an assessment. There are differences between 32 and 350 average exposed elements per layer. This shows a good range for evaluation purposes. As an example, if the design is printed from the inlet in the direction of flow to the back of the engine mount. It is imaginable that it is rather inconvenient and worthy of reworking if you need a lot of material around the motor, but only have thin branches to the front.

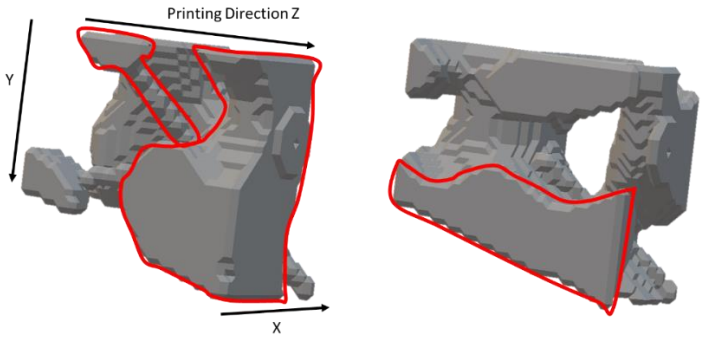

Figure 7: Design with a High (left) and Low (right) Rate of Material Change

In Figure 7 two designs are shown, the design on the left that requires a lot of support and the right design is easy to print according to this criterion. The right design has a smooth transition of material and an accompanying uniform heat input across the layers.

The data set as a whole offers a high degree of consistency for the statements and conclusions made. After already 100 examples, the datapoints are already normally distributed across the design space. For each design point to be determined by the metamodel, there are approx. 40-50% (light blue to green, Figure 8) inputs to the subsequent deep learning model

as a proportion of the total examples. A desired 100% (red) material distribution is only present in the geometrically desired areas (engine, screwing points).

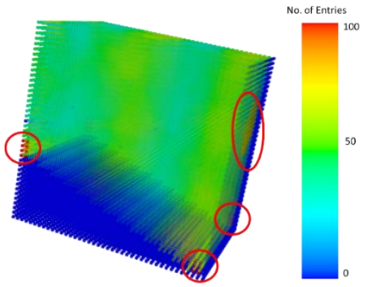

Figure 8: Distribution of the Entries over the Entire Code Generation:

The final evaluation of the suitability for simulations shows good applicability of the generic designs. The meshing process and the setting of the boundary conditions is possible in an automated manner. The finished STL files are automatically loaded into the ANSYS Workbench for testing by a script in Iron Python with ANSYS Spaceclaim. 4.8% of the designs showed problems creating an automated mesh body with different general settings. A program of specific errors occurred which could not be traced back to any systematic. In order to take this into account in the following, an over-generation of designs by approx. 5% is used to just ignore failed designs. The results with the automated meshed bodies, depending on the smoothed STL files, shows good consistency in simulations [Figure 9].

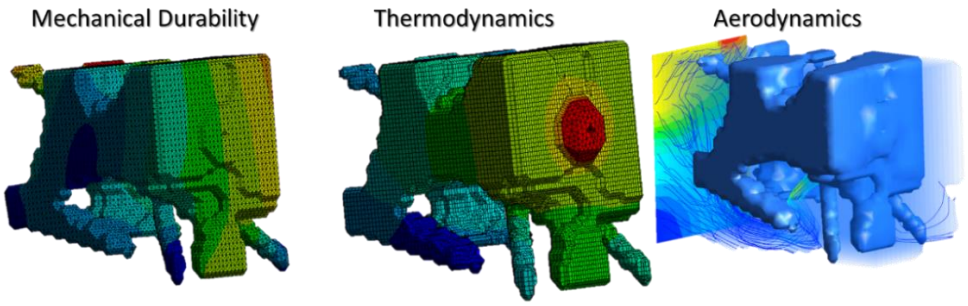

Figure 9: Example of one Simulation per Optimization Discipline

5. Conclusions

In this work it is shown how to generate data which can be used to address MDO problems over a deep learning model. The data set is validated and checked for the application described. The approach presented shows a solution for non-existent data sets for a complex product development problem. Any boundary conditions given by geometry can be incorporated into the design work as long as the function can be described generically. In this way, boundary conditions from different physical disciplines (MDO) and functional properties for production, which are particularly difficult to link mathematically, can be assessed together. Furthermore, through a systematic review of the data set, it is possible to know exactly about existing properties and data distribution in relation to your design space. The manageability of the data for adjustments, calculations or simulation is given in full through the use of a voxel-based procedure. Due to their uniform form, the data do not require any preprocessing in relation to deep learning models. Data preprocessing is already done during the systematic generation and skips one of the most demanding parts in the training of such models. Only the

schematic approaches used in the code must be followed critically, whether you have patterns or other weak points in relation to the training of deep learning models which can lead to an over- or underfitting in individual areas. Models trained with this data set are already show good accuracy in the evaluation of an optimal static and thermal design according to the procedure described. In further work, various trained deep learning models will be shown, evaluated and discussed. This is where the fundamental suitability of the data generated here becomes fully apparent.

Literature

[1] Yang, S., Tang, Y., & Zhao, Y. F. (2015). A new part consolidation method to embrace the design freedom of additive manufacturing. *Journal of Manufacturing Processes*, *20*(June), 444–449. https://doi.org/10.1016/j.jmapro.2015.06.024

[2] Kamps, T., Gralow, M., Schlick, G., & Reinhart, G. (2017). Systematic Biomimetic Part Design for Additive Manufacturing. *Procedia CIRP*, *65*, 259–266. https://doi.org/10.1016/j.procir.2017.04.054

[3] Durakovic, B. (2018). Design for additive manufacturing: Benefits, trends and challenges. *Periodicals of Engineering and Natural Sciences*, *6*(2), 179–191. https://doi.org/10.21533/pen.v6i2.224

[4] Joaquim R. R. A. Martins. (2012). A Short Course on Multidisciplinary Design Optimization. *University of Michigan*.

[5] Simpson, T. W., Mauery, T. M., & Korte, J. J. (2001). Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Desgin Optimization. *AIAA*, *39*.

[6] Duddeck, F. (2015). Multidisziplinäre Optimierung im Produktentwicklungsprozess der Automobilindustrie.

[7] Vietze, B. (2018). Gekoppelte aerothermodynamische und strukturmechanische Optimierung kryogener Raketenoberstufen. Universität der Bundeswehr München.

[8] Bierdel, M., Hoschke, K., Pfaff, A., Jäcklein, M., & Schimmerohn, M. (2017). Multidisciplinary Design Optimization of a Satellite Structure by Additive Manufacturing. *68th International Astronautical Congress (IAC), Adelaide, Australia*, (September), 25–29.

[9] Bikas, H., Lianos, A. K., & Stavropoulos, P. (2019). A design framework for additive manufacturing. *International Journal of Advanced Manufacturing Technology*, *103*, 9–12. https://doi.org/10.1007/s00170-019-03627-z

[10] F. Viana, T. Simpson, V. Balabanov, Metamodeling in multidisciplinary design optimization: How far have we really come?, AIAA Journal , vol. 52, pp. 670-690, 2014

[11] G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, Journal of Mechanical Design, vol 129, pp. 370-380, 2007

[12] Han, C., Rundo, L., Araki, R., Furukawa, Y., Mauri, G., Nakayama, H., & Hayashi, H. (2020). Infinite Brain MR Images: PGGAN-Based Data Augmentation for Tumor Detection. *Smart Innovation, Systems and Technologies*, *151*, 291–303. https://doi.org/10.1007/978-981-13-8950-4_27

[13] Bhattarai, B., Baek, S., Bodur, R., & Kim, T. K. (2020). Sampling strategies for gan synthetic data. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2020-May*, 2303–2307. https://doi.org/10.1109/ICASSP40776.2020.9054677

[14] Surendra, H., & Mohan, H. S. (2015). A Review of Synthetic Data Generation Methods For Privacy Preserving Data Publishing. *International Journal of Scientific & Technology Research*, *4*(8), 95–101.

[15] W. Cao, Z. Yan, Z. He, A Comprehensive Survey on Geometric Deep Learning, IEEE Access, 2020

[16] M. Dering, C. Tucker, A Convolutional Neural Network Model for Predicting a Product`s Function, Given Its From, Journal of Mechanical Design, 2017

[17] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang, 'Deep Generative Design: Integration of Topology Optimization and Generative Models', *ArXiv190301548 Cs*, May 2019, Accessed: May 04, 2020. [Online]. Available: http://arxiv.org/abs/1903.01548

[18] Y. Yu, T. Hur, J. Jung, and I. G. Jang, 'Deep learning for determining a near-optimal topological design without any iteration', *Struct. Multidiscip. Optim.*, vol. 59, no. 3, pp. 787–799, Mar. 2019, doi: 10.1007/s00158-018-2101-5.

[19] R. K. Tan, N. L. Zhang, and W. Ye, 'A deep learning–based method for the design of microstructural materials', *Struct. Multidiscip. Optim.*, vol. 61, no. 4, pp. 1417–1438, Apr. 2020, doi: 10.1007/s00158-019-02424-2.

[20] Z. Nie, T. Lin, H. Jiang, and L. B. Kara, 'TopologyGAN: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain', *ArXiv200304685 Cs Eess*, Mar. 2020, Accessed: May 18, 2020. [Online]. Available: http://arxiv.org/abs/2003.04685

[21] W. Silversmith. "cc3d: Connected Components on Multilabel 3D Images", January 2021. https://github.com/seung-lab/connected-components-3d/