HAL open science

Data-driven surrogate modelling of residual stresses in Laser Powder-Bed Fusion

L. Lestandi, J.C. Wong, G.Y. Dong, S. Kuehsamy, J. Mikula, G. Vastola, U. Kizhakkinan, C.S. Ford, D.W. Rosen, M.H. Dao, et al.

To cite this version:

L. Lestandi, J.C. Wong, G.Y. Dong, S. Kuehsamy, J. Mikula, et al.. Data-driven surrogate modelling of residual stresses in Laser Powder-Bed Fusion. International Journal of Computer Integrated Manufacturing, 2023, pp.1-23. ⟨10.1080/0951192X.2023.2257628⟩. ⟨hal-04543550⟩

HAL Id: hal-04543550
https://hal.science/hal-04543550v1

Submitted on 12 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Autorisation HAL

HAL Authorization

Data-driven surrogate modelling of residual stresses in Laser Powder-Bed Fusion

ARTICLE HISTORY

Compiled June 21, 2023

ABSTRACT

In order to enable the industrialization of additive manufacturing, it is necessary to develop process simulation models that can rapidly predict part quality. Although multi-physics simulations have shown success at predicting residual stress, distortion, microstructure and mechanical properties of additively manufactured parts, they are generally too computationally expensive to be directly used in applications such as optimization, controls, or operational digital twinning. In this study, a critical evaluation is made for how data-driven surrogate models can be used to model the residual stress of parts fabricated by Laser Powder-Bed Fusion. Residual stress data is generated by using an inherent-strain based process simulation for two families of part geometries. Three different models using varying levels of sophistication are compared: a multilayer perceptron (MLP), a convolutional neural network (CNN) based on the U-Net architecture, and a interpolation-based method based on mapping geometries onto a reference. All three methods were found to be sufficient for part design, providing mechanical for a CPU time below 0.2s, representing a runtime speed-up of at least 3900x. Neural network based models are significantly more expensive to train compared to using interpolation. However, the generality of models based on the U-Net architecture is attractive for applications in optimization.

KEYWORDS

Laser Powder Bed Fusion ; Additive Manufacturing ; Geometry parametrization ; surrogate models ; mapping ; RBF ; Neural Network ; CNN ; MLP

1. Introduction

Additive manufacturing (AM) processes such as Laser Powder-Bed Fusion (LPBF) are potentially transformative technologies due to their ability to create complex parts that are difficult to form through other, more conventional means. However, AM processes tend to be very complex, often exposing the material to conditions very different from conventional processing. The digital simulation of AM processes is therefore an essential tool to control the LPBF process and enable its full potential.

The LPBF process is inherently multiscale and multiphysics, where microscopic-scale phenomena such as the laser-powder interaction can lead to significant changes in the residual stresses and distortion at the part-scale (Markl and Carolin 2016). Consequently, although a variety of research groups (Promoppatum, Onler, and Yao 2017; Joshi et al. 2021; Mikula et al. 2021) and companies (e.g. Simufact, ANSYS Additive, Siemens NX) have developed process simulations to study the LPBF process, these tools are generally not computationally efficient enough to be used for optimization or process control. Simulation of a single complex build can take hours depending on the level of fidelity of the computational tool.

A typical approach to overcome this difficulty is to develop surrogate models that are computationally less expensive to evaluate, but still are able to capture the correct predictions of the full-order model (FOM). Data-driven surrogate models such as neural networks or response surfaces allow most of the computational effort to be done off-line. Such machine learning (ML) methods have been applied to AM processes as discussed in (Qi et al. 2019; Meng et al. 2020). Surrogate models have been applied to a variety of problems in AM, including estimating mechanical properties (Ansell et al. 2020; Baturynska 2019; Popova et al. 2017), surface roughness (Wu, Wei, and Terpenny 2019), melt pool structure (Kizhakkinan et al. 2023), microstructure (Gan et al. 2019) and thermal history (Li and Polydorides 2022).

Surrogate modeling has enabled the integration of simulation, data and AM to deal with several aspects impacting printability. For example, strategies have been developed to optimize build costs utilizing Long-Short Term Memory (Ulu et al. 2019), Generative Adversarial Networks, and Convolutional Neural Network architectures

(Nie, Jiang, and Kara 2019; Nie et al. 2020). Gaussian Process based modeling has been used to predict porosity distributions due to printing (Tapia, Elwany, and Sang 2016) as well as predicting robust process parameters for printing (Tapia et al. 2018).

In this paper, strategies for developing surrogate models for AM for the purposes of component design are critically evaluated. In particular, focus is placed on accelerating the calculation of process-induced residual stresses and distortions at the part-scale (Mercelis and pierre Kruth 2006). This problem has been studied with dimension reduction (Vohra et al. 2020) and applied to geometry compensation using techniques such as Artificial Neural Networks (ANN) (Chowdhury 2016) and reduced order modeling (Quaranta et al. 2019). This study focuses on this area since the full-order model is well established: the inherent strain approximation (Ueda et al. 1975) is generally accepted as valid for calculating residual stresses and distortions at the part-scale during the LPBF process.

Component design offers particular challenges to surrogate modelling, as it is necessary to be able to rapidly assess a large number of geometries. To develop such surrogate models, it is useful to be able to describe a geometric parameterization (Staten et al. 2011) that can efficiently describe a variety of potential component shapes. Several schemes have been developed to describe intricate shapes using a few parameters. Free form deformation (FFD) uses a small number of volumetric control points on the surface to define the remainder of the component geometry (Manzoni, Quarteroni, and Rozza 2012; Demo et al. 2018). Interpolation methods such as inverse distance weighting (Sen, De Nayer, and Breuer 2017; Ballarin et al. 2019) and radial basis functions (RBF) (Botsch and Kobbelt 2005; de Boer, van der Schoot, and Bijl 2007; Ammar et al. 2013; Sieger, Menzel, and Botsch 2014; Biancolini et al. 2020) have similarly been used to morph between different mesh geometries. Another approach consists of finding a low order representation of complex shapes that describe the geometry using a small number of spatial modes (Chinesta et al. 2017) obtained by applying any low order approximation on the position of the surface.

Families of simple geometries (such as those studied in this paper) can be described using a feature-based approach (Rosen and Grosse 1992) using explicit analytical description of simple shapes (cylinders, spheres, cubes, etc.) combined with geometric operations (addition, subtraction, rotation) which can be rendered in parametric CAD software such as FreeCAD (Team 2020). This method limits the complexity of the component, but facilitates analysis. For example, one can very easily determine if an arbitrary point is inside or outside the surface.

The remainder of this paper is organized as follows. The next section formally describes the problem. Three different data-driven surrogate modeling methods for part-scale mechanical simulation are presented: a standard neural network based on a multilayer perceptron (MLP); a convolutional neural network (CNN); and a map-decompose-interpolate (MDI) approach that exploits geometry parameterization. In section 5 these methods are applied to data produced with a voxel based inherent strain simulation. Finally, each strategy is critically evaluated and the relative merits of each method are discussed.

2. Statement of the problem

The goal of this work is to develop a data-driven method that allows the computationally efficient evaluation of the mechanical properties in a martensitic steel component fabricated using laser powder bed fusion (LPBF). To this end, the residual stress developed upon printing two groups of simple geometries with no support structure was determined. The first group consists of a thin plate containing an elliptical hole with various shapes and sizes. The second group consists of trapezoidal prisms with different heights and base sizes, representing bulky structures.

The geometry of each build is parametrically generated using FreeCAD, then exported as an STL file. This geometry is then voxelized using a mesh and voxel size standardized for each geometry group. The plate geometries use a voxel size of $0.5mm \times 6mm \times 0.5mm$ and are embedded in a $100 \times 1 \times 200$ mesh. The wedge geometries use a voxel size of $0.5mm \times 5mm \times 0.5mm$ and are embedded in a $40 \times 10 \times 60$ mesh. The residual stress and strain are calculated using a modified inherent strain method (Mikula et al. 2021). The part scale simulation step is referred to in this paper as the Full Order Model (FOM), and is discussed in more details in (Dong et al. 2022). Simulations are performed using the materials properties as presented in table 1. A

Mechanical Property	Value
Young's Modulus E (GPa)	160
Ultimate tensile strength (MPa)	1050
Poisson ratio ν	0.23
Hardening coefficient h	69.4
Inherent strain ε_{inh}	-0.008

Table 1.: Materials properties for process simulation used in all numerical experiments of this paper.

single simulation run of the FOM takes 1h or less on a high performance workstation with two Intel® Xeon® Gold 6230 CPUs with 32 cores @ 2.10GHz.

The key outputs that will be modeled in this study are the residual stress (measured through the scalar von Mises stress), and the distortion (measured through the scalar displacement and the displacement vector). The FOM outputs the values of these fields at the nodes of the voxelized mesh. These outputs are taken as representative parameters for LPBF process simulations: since these surrogate models are data-driven, they are suitable for any relevant simulation outputs such as local strength or microstructure, as long as the FOM is appropriately extended.

The surrogate modeling problem can be formally be stated as follows. Considering a group of geometries, they may describe using a vector of parameters $\boldsymbol{\mu} = (\mu_1, ..., \mu_p) \in P$ where P is the set of possible parameters. The process model predicts the quantity of interest, $u(\boldsymbol{x}; \boldsymbol{\mu})$, using the geometry as an input. $u(\boldsymbol{x}; \boldsymbol{\mu})$ can be either a scalar or vector valued function and is defined at all points $\boldsymbol{x} \in \Omega(\boldsymbol{\mu})$ the parametrized geometry.

In this study, purely data driven surrogate model $\hat{u}$ are developed to offers fast and accurate approximation of u,

$$\hat{u}(\boldsymbol{x}; \boldsymbol{\mu}) \approx u(\boldsymbol{x}; \boldsymbol{\mu}),\ \forall(\boldsymbol{x}; \boldsymbol{\mu}) \in \Omega(\boldsymbol{\mu}) \times P. \tag{1}$$

While only geometric parameters are used, the surrogate could be extended to include other variables such as process parameters as long as sufficient data is sampled. The challenge of developing a surrogate model is that the relationship between geometry and u is complex and difficult to evaluate. However, this surrogate may be utilized in applications such as design or optimization of structures

The performance of the different surrogate models will be compared using the relative error,

$$E(\hat{u}, u) = \frac{||u - \hat{u}||}{||u||} \tag{2}$$

Different definitions of the norm are used to characterize different aspects of the error. The L_2 norm is a measure of the Euclidean length of the vector $u - \hat{u}$, and so measures the overall accuracy of the model. The L_∞ norm finds the largest element of $u - \hat{u}$, and so measures the maximum error of the model.

3. Deep neural networks-based surrogate models

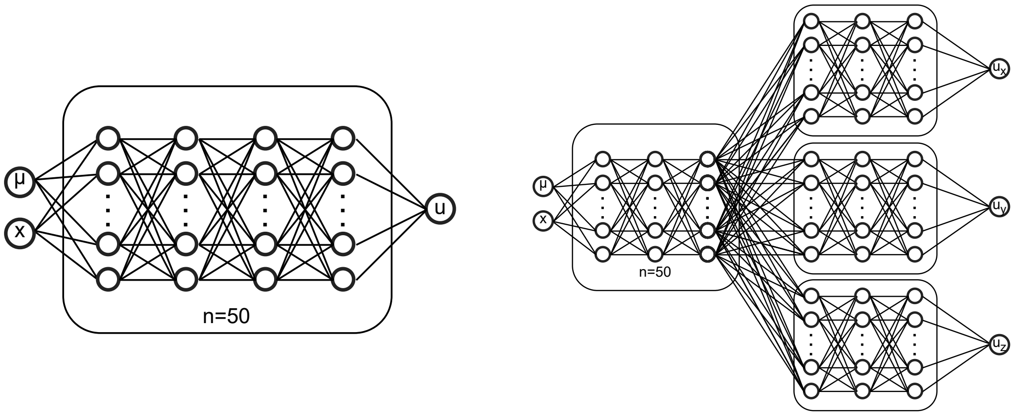
In order to leverage the strength of deep neural networks at regression tasks, $u(\boldsymbol{x}; \boldsymbol{\mu})$ in Eq. (1) is approximated by $\hat{u}(\boldsymbol{x}; \boldsymbol{\mu}; \boldsymbol{w})$, where $\boldsymbol{w}$ is a set of neural network parameters to be optimized. The intermediate states of the build process are excluded from the training data and only the final spatial data is used for training and testing. While many neural network architectures could be utilized, the scope of this work is limited to two well-known architectures. The first architecture considered in this work, the multilayer perceptron (MLP), is a feedforward network made up of input, hidden and output nodes in which each layer is fully connected. MLP is a universal function approximator (Hornik, Stinchcombe, and White 1989), thus should be suitable for the non-linear interpolation problem studied here. The second architecture considered is the convolutional neural network (CNN), which has shown great success in visual recognition tasks (Lecun, Bengio, and Hinton 2015) such as object classification (Krizhevsky, Sutskever, and Hinton 2017), object segmentation(Shelhamer, Long, and Darrell 2017) and medical image analysis (Yamashita et al. 2018). This method utilizes a sparsely connected network which accounts for local connectivity, and is considered well suited for data including spatial relationships. Unlike MLPs, CNNs can be directly applied to voxelized geometries generated by FOM (Dong et al. 2022), without the need to introduce a parameterization factor.

3.1. *Multilayer perceptron (MLP) surrogate model*

In this work, the MLP is utilized by directly taking the geometry parameter $\boldsymbol{\mu}$ and spatial coordinate $\boldsymbol{x}$ as inputs, and returning the prediction $\hat{u}_{MLP}(\boldsymbol{x};\boldsymbol{\mu};\boldsymbol{w})$. The MLP is trained using backpropagation (Goodfellow, Bengio, and Courville 2016), where the gradient of neural network weights $\boldsymbol{w}$ is computed with respect to a loss function $\mathcal{L}$ and updated based on the errors of the previous epoch. For a given dataset consisting of n samples (input–output pairs), the loss function is defined using mean squared error (MSE),

$$\mathcal{L} = L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(u_i - \hat{u}_{MLP}(\boldsymbol{x}_i;\boldsymbol{\mu}_i;\boldsymbol{w})\right)^2 . \tag{3}$$

The MLP architecture and training settings used in this study are described as follows: The network comprises of 4 hidden layers with 50 neurons plus Swish activation, before connecting to a non-negative scalar output $u \in \mathbb{R}^+$, e.g., von Mises stress and displacement magnitude, through a ReLU activation in the output layer. For a vector output $u \in \mathbb{R}^m$, e.g., displacement vector, the network branches out to m sets of 3 hidden layers with a linear activation in the last hidden layer that predicts each of vector components. Their network weights are initialized by the Xavier method (Glorot and Bengio 2010). These models were created using the Keras API as packaged with TensorFlow 2.5 (Abadi et al. 2016). The ADAM optimizer (Kingma and Ba 2015) is used for optimizing the network weights with respect to MSE loss. Table 2 provides an overview of the MLP architecture used for each specific sub-problem. The architecture was tuned for the specific data of each regression problem. Although the shape of the input data varies depending on the specific component geometry, tuning the architecture of the hidden layers was not needed. However, predicting scalar and vector outputs requires different network architectures as shown in Fig. 1. For example, to model the displacement vector, a set of common "trunk" layers is used, followed by branches specific to each component of the vector.



(a) Scalar output MLP, 4 layers, denoted $(\theta, M, x, y, z) - 50 - 50 - 50 - 50 - \hat{u}$

(b) Vector output MLP, 3 trunk layers followed by 3 branches of 3 layers, denoted $(\theta, M, x, y, z) - 50 - 50 - 50 - \left[\begin{array}{l} 50 - 50 - 50 - (\hat{u}_x), \\ 50 - 50 - 50 - (\hat{u}_y), \\ 50 - 50 - 50 - (\hat{u}_z) \end{array} \right.$

Figure 1.: Architecture of MLP depending on output. All hidden layers have 50 neurons.

3.2. *Convolutional Neural Network (CNN) surrogate model*

The voxelized input geometry is represented as binary matrices where 0 represents empty space and 1 represents a solid voxel. For training a CNN model, the voxelized geometries are padded with 0s such that the size of the matrices in the training set are all the same size, e.g., 256x128 for the 2D plate geometries and 24x64x32 for 3D wedge geometries.

In the present study, a CNN architecture known as U-Net (Ronneberger, Fischer, and Brox 2015; Özgün Çiçek et al. 2016; Iglovikov and Shvets 2018) is implemented, which is suited for the field prediction task because it assigns output value(s) to each input pixel / voxel for 2D / 3D inputs (Yao et al. 2018; Le, Chiu, and Ooi 2021; Chen, Viquerat, and Hachem 2019). Fig. 2 shows the detailed architecture implemented for 3D data. Briefly, the U-Net is comprised of separate contraction and expansion networks making up of successive layers of convolution, batch normalization, nonlinear activation, and pooling / up-sampling operations. The contraction network is designed to gradually map the input into increasing number of advanced feature maps and also to reduce the feature map sizes. The expansion network is symmetric but acts in the opposite direction of the contraction half, where the pooling operations are replaced by the up-sampling operations, to gradually map from the advanced features

Table 2.: MLP architectures have been fine tuned for the specific problems at hand.

	von Mises stress	displacement magnitude	displacement vector
MLP architecture –no. neurons–	$(\theta, M, x, y, z)-$ $50-50-50-50-\hat{u}$	$(\theta, M, x, y, z)-$ $50-50-50-50-\hat{u}$	$(\theta, M, x, y, z)-$ $50-50-50$ $-\left[\begin{array}{l} 50-50-50-(\hat{u}_x), \\ 50-50-50-(\hat{u}_y), \\ 50-50-50-(\hat{u}_z) \end{array}\right.$
Activation	Swish, ReLU	Swish, ReLU	Swish, linear
Loss type	MSE	MSE	MSE
Max. training epoch	500	500	1000

(a) Plate with hole data

	von Mises stress	displacement magnitude	displacement vector
MLP architecture –no. neurons–	$(L_x, L_x, H, x, y, z)-$ $50-50-50-50$ $-\hat{u}$	$(L_x, L_x, H, x, y, z)-$ $50-50-50-50$ $-\hat{u}$	$(L_x, L_x, H, x, y, z)-$ $50-50-50-$ $-\left[\begin{array}{l} 50-50-50-(\hat{u}_x), \\ 50-50-50-(\hat{u}_y), \\ 50-50-50-(\hat{u}_z) \end{array}\right.$
Activation	Swish, ReLU	Swish, ReLU	Swish, linear
Loss type	MSE	MSE	MSE
Max. training epoch	500	500	1000

(b) Wedge data

extracted by the contraction network to the final output. An important feature of the U-Net is the skip connections from layers of equal resolution in the contraction half to its expansion counterpart, which allows the network to more easily propagate the localization information to later layers, and also back propagate the loss gradients to earlier layers.

The performance of the U-Net model is more sensitive to the network architecture and training hyper-parameters compared to MLP. The U-net design is based on the authors previous work (Dong et al. 2022) and have been fine-tuned for each specific prediction tasks resulting in different network architectures where the training hyper-parameters are reported in table 3. All these models are initialized by the Xavier method (Glorot and Bengio 2010) , and trained by the ADAM (Kingma and Ba 2015) optimizer with respect to MAE (L_{MAE}) or MSE (L_{MSE}) loss for at least 1k epochs.

$$\mathcal{L} = L_{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left| u_i - \hat{u}_{MLP}(\boldsymbol{x}_i; \boldsymbol{\mu}_i; \boldsymbol{w}) \right| . \tag{4}$$

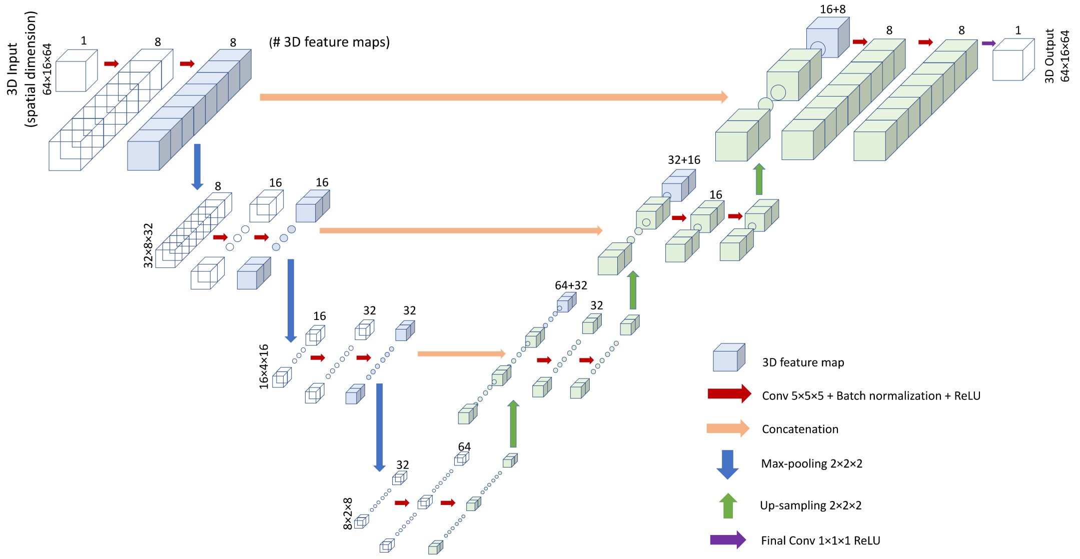


Figure 2.: Scalar output Unet-CNN architecture predicting displacement magnitude for 3D wedge geometries.

Table 3.: U-Net architecture and training settings for each dataset studied.

	von Mises stress	displacement magnitude	displacement vector
CNN architecture –no. filter–	$(X_{256\times128})-$ $16-16-32-32-64-$ $64-128-128-64-$ $64-32-32-16-16$ $-(\hat{U}_{256\times128})$	$(X_{256\times128})-$ $16-16-32-32-64-$ $64-128-128-64-$ $64-32-32-16-16$ $-(\hat{U}_{256\times128})$	$(X_{256\times128})-$ $16-16-32-32-$ $64-64-128-128-$ $64-64-32-32-[$ $16-16-\hat{U}_{x,256\times128},$ $16-16-\hat{U}_{y,256\times128},$ $16-16-\hat{U}_{z,256\times128}$
Activation	ReLU	ReLU	ReLU, linear
Loss type	MAE	MAE	MSE
Max. training epoch	1000	1000	2000

(a) Plate with hole data

	von Mises stress	displacement magnitude	displacement vector
CNN architecture –no. filter–	$(X_{24\times64\times32})-$ $16-16-32-32-64-$ $64-128-128-64-$ $64-32-32-16-16$ $-(\hat{U}_{24\times64\times32})$	$(X_{24\times64\times32})-$ $16-16-32-32-64-$ $64-128-128-64-$ $64-32-32-16-16$ $-(\hat{U}_{24\times64\times32})$	$(X_{24\times64\times32})-$ $16-16-32-32-$ $64-64-128-128-$ $64-64-32-32-[$ $16-16-\hat{U}_{x,24\times64\times32},$ $16-16-\hat{U}_{y,24\times64\times32},$ $16-16-\hat{U}_{z,24\times64\times32}$
Activation	ReLU	ReLU	ReLU, linear
Loss type	MAE	MAE	MSE
Max. training epoch	1000	1000	2000

(b) Wedge data

4. Map-Interpolate (MI) surrogate model

In this section, a surrogate model explicitly incorporating geometry parameterization is presented. It is named Map-Interpolate (MI). In this strategy, during an *offline* pre-computing phase described in Algorithm 1, the input data is mapped onto a reference geometry where one can use any off-the-shelf method to interpolate in the parametric space. Next, the *online* phase consists in interpolating at the target parameters followed by an inverse mapping of the fields to the target geometry, described in Algorithm 2.

Algorithm 1: MI Offline phase

---

input : Precomputed samples : $\{u(\boldsymbol{x};\boldsymbol{\mu}_i)\}_{i=1}^{N_{samples}}$

1 Map samples to reference: eq. (6) $\forall i \leq N_{samples},\ \forall x \in \Omega(\boldsymbol{\mu})$,

$$\bar{u}\left(\bar{\boldsymbol{x}}, \boldsymbol{\mu}_i\right) = u(\mathcal{M}(\bar{x};\boldsymbol{\mu}_i);\boldsymbol{\mu}_i) = u(\boldsymbol{x};\boldsymbol{\mu}_i)$$

2 Store snapshots $\{\bar{u}_1, \cdots, \bar{u}_{N_{samples}}\}$ with $\forall i \leq N_{samples},\ U_i \in \mathbb{R}^n_{ref}$ the discrete representation of $\bar{u}_i$

3 Precompute any purely offline step of the interpolation method (e.g. HOSVD)

Algorithm 2: MI Online phase

---

input : Precomputed snapshots and algorithm 1:line 3 result

1 Interpolate the snapshots at target $\hat{\boldsymbol{\mu}} = (\hat{\mu_1}, ..., \hat{\mu}_{N_\mu})$ and obtain

$$\forall x \in \Omega_{ref},\ \bar{u}(x, \hat{\boldsymbol{\mu}})$$

2 Map back to physical space,

$$\forall \boldsymbol{x} \in \Omega(\hat{\boldsymbol{\mu}}), \quad u_{MI}(x;\hat{\boldsymbol{\mu}}) = \hat{u}(\mathcal{M}^{-1}(\bar{\boldsymbol{x}};\hat{\boldsymbol{\mu}}), \hat{\boldsymbol{\mu}})$$

output: Approximation at target parameter $\hat{\mu}$: $\forall \boldsymbol{x} \in \Omega(\boldsymbol{\mu}),\ \hat{u}_{MDI}(x;\hat{\mu})$

4.1. *Mapping*

The training data is mapped onto a topologically equivalent reference domain $\Omega_{ref} = \Omega(\boldsymbol{\mu}_{ref})$ using a parametrized map $\mathcal{M} : \Omega_{ref} \times P \longrightarrow \Omega(\boldsymbol{\mu})$ defined such that

$$\Omega(\boldsymbol{\mu}) = \mathcal{M}(\Omega_{ref}, \boldsymbol{\mu}) \tag{5}$$

Let $\bar{u} : \Omega_{ref} \longrightarrow \mathbb{R}^d$ such that $\forall \bar{\boldsymbol{x}} \in \Omega_{ref}$ where $\mathcal{M}(\bar{\boldsymbol{x}}; \boldsymbol{\mu}) = \boldsymbol{x}$, $\bar{u}$ verifies

$$\bar{u}\,(\bar{\boldsymbol{x}}, \mu) = u(\mathcal{M}(\bar{x}; \boldsymbol{\mu}); \boldsymbol{\mu}) = u(\boldsymbol{x}; \mu) \tag{6}$$

In other words, for any field variable $u(\cdot; \mu)$, $\bar{u}(\cdot, \mu)$ is the same field mapped onto Ω_{ref}, the reference space. Figure 4 illustrates the definition of geometry mapping.

The radial basis function (RBF) mapping method is used to map the volume of the geometry using control points placed on the surface (de Boer, van der Schoot, and Bijl 2007). The `multiquadratic_biharmonic_spline` RBF kernel is used as it showed good accuracy and stability for all choices of radius and step size. The control points are placed regularly on the coupon surfaces according to the constructive solid geometry used to generate the CAD files. The mesh is mapped to the reference geometry using `pygem`(Tezzele et al. 2021), after which the data is interpolated at the nodes of the reference mesh.

4.2. *Interpolation methods*

After the mapping, all the snapshots share the same geometry (discrete and continuous). Consequently, any off-the-shelf interpolation method can be used. In this work, the nearest neighbour interpolation (NNI) is used as baseline. Linear interpolation is tested as well since it scales well even for large parametric spaces. Finally, a model based on high order singular value decomposition (HOSVD) is proposed to reduce the model size when the parametric space dimension is large i.e. $\#P > 2$.

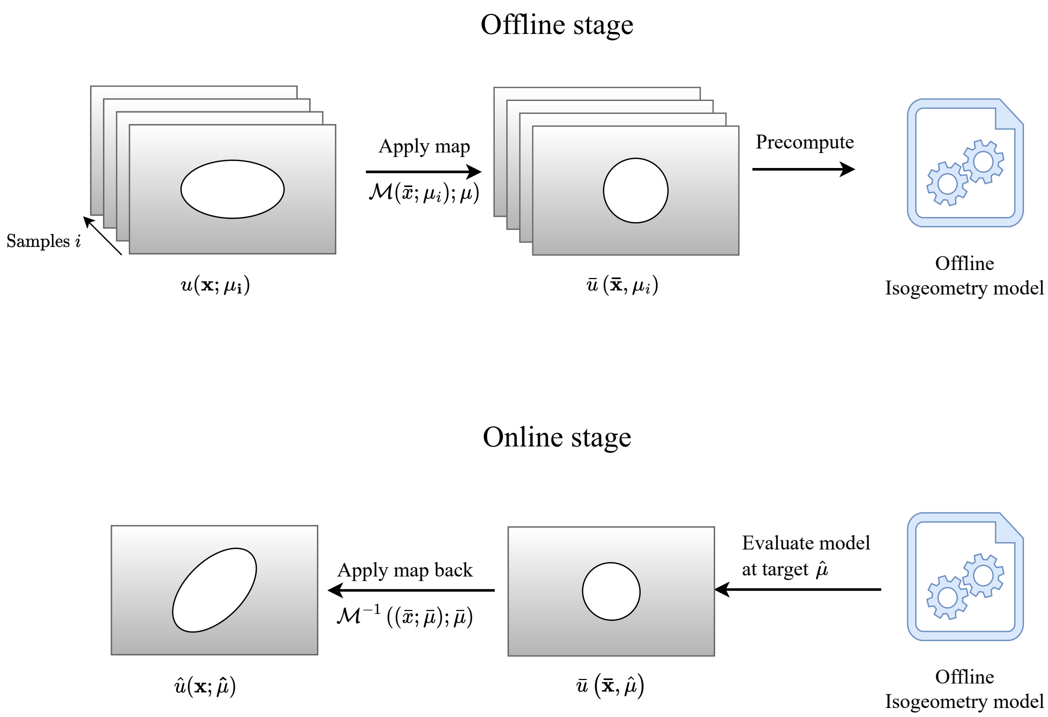


Figure 3.: Graphical representation of the Map Interpolate method where the Isogeometry model is the snapshots $\bar{u}$ for standard interpolation method or the HOSVD decomposition when the parametric space dimension is large.

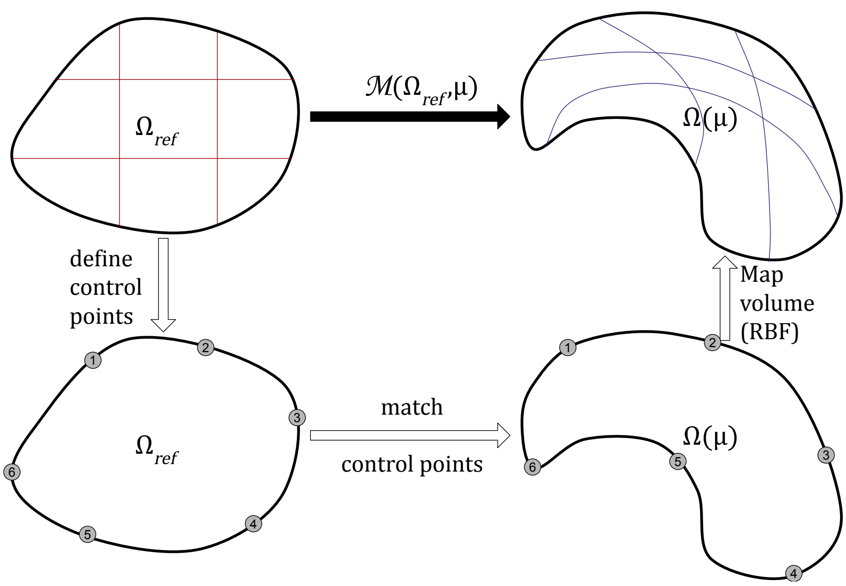


Figure 4.: Illustration of a map $\mathcal{M}$ between reference geometry Ω and parametrized geometries $\Omega(\boldsymbol{\mu})$.

4.2.1. Nearest Neigbour Interpolation (NNI)

Note that even for this basic model, the mapping stage was necessary to "project" the nearest neighbour $\hat{\boldsymbol{\mu}}_{nn}$ of $\hat{\boldsymbol{\mu}}$ onto the actual geometry $\Omega(\hat{\boldsymbol{\mu}})$. Within the reference geometry, the model reads:

$$\forall \bar{\boldsymbol{x}} \in \Omega_{ref}, \quad \hat{u}_{NNI}(\bar{\boldsymbol{x}}, \hat{\boldsymbol{\mu}}) = \bar{u}(\bar{\boldsymbol{x}}, \boldsymbol{\mu}_{NN}) \tag{7}$$

4.2.2. Linear interpolation

As for NNI, one can use a standard multivariate piece-wise linear interpolation within the reference geometry. For compatibility with the next method (subsection 4.2.3), the study is restricted to rectilinear parametric grids. Then, one can use a sequence of 1D linear interpolation e.g. ND-linera interpolation ((Weiser and Zarantonello 1988)). Finally,

$$\forall \bar{\boldsymbol{x}} \in \Omega_{ref}, \quad \hat{u}_{lin}(\bar{\boldsymbol{x}}, \hat{\boldsymbol{\mu}}) = \texttt{Nlinear\_Interpolation}\left(\{\bar{u}_i(\bar{\boldsymbol{x}},)\boldsymbol{\mu}_i\}_i, \bar{\boldsymbol{x}}, \hat{\boldsymbol{\mu}}\right) \tag{8}$$

4.2.3. HOSVD+Interpolation (MDI)

Inspired by the surrogate modeling method of Proper Orthogonal Decomposition with Interpolation (PODI) (Bui-Thanh, Damodaran, and Willcox 2004) a data decomposition is performed on the mapped training data. In this method, a set of data samples, often referred as snapshots, are decomposed into modes to build a low order approximation of the underlying manifold of the data. This allows the application of 1D interpolation techniques (e.g. piece-wise polynomial, Lagrange or splines) to simplify parametric interpolation over the solution space. This non-intrusive surrogate modelling strategy is similar to recent work (Hesthaven and Ubbiali 2018; Ghnatios et al. 2021; Demo, Tezzele, and Rozza 2019) where the model does not need the PDE once the training data has been acquired, even when dealing with multiple parameters at once (Duong et al. 2020).

As all the samples have been mapped onto Ω_{ref}, the snapshots have exactly the same mesh and thus the data can be well described by constraining the data sampling on the tensor product grid. This means that this model is a tensor $\tilde{U}$ of size $N_x \times N_{\mu_1} \times \cdots \times N_{\mu_p}$ representing $\tilde{u}$ at grid points. The entries of the tensor are defined as follows:

$$\forall\, i_0 \leq N_x, ..., i_p \leq N_{\mu_p}, \quad \bar{U}_{i_0 i_1 \ldots i_p} = u(\bar{\boldsymbol{x}}^{i_0}, \mu_1^{i_1}, ..., \mu_p^{i_p})$$

Note that computational limitations coming from the size of tensor may be alleviated using more advanced sparse representation and decomposition techniques (Ballani, Grasedyck, and Kluge 2010; Dahmen et al. 2015; dung Nguyen, Abed-meraim, and trung Nguyen 2016). The trajectories are centered by removing the point-wise average,

$$\langle \bar{U} \rangle = \frac{\sum_{i_1,...,i_p} \bar{U}_{:i_1 \ldots i_p}}{N_{samples}} \tag{9}$$

Finally, tensor U is defined as:

$$\forall i_0 \leq N_x, \quad U_{i_0} = \bar{U}_{i_0} - \langle \bar{U} \rangle \tag{10}$$

Any tensor decomposition method can be used in the algorithm. In this case, the Tucker tensor decomposition was applied to U, specifically the Sequentially Truncated Higher Order Singular Value Decomposition (ST-HOSVD) (Vannieuwenhoven, Vandebril, and Meerbergen 2012), since it provides easy representation and is efficient for reasonably low dimension ($d < 5$)(Lestandi 2021). The decomposition U then reads,

$$U \approx \widetilde{U} = \sum_{k_0=1}^{r_0} \cdots \sum_{k_p=1}^{r_p} w_{k_0 \ldots k_p} X^{k_0} \otimes \varphi_1^{k_1} \otimes \cdots \otimes \varphi_p^{k_p} \tag{11}$$

Where X^{k_0} is the k_0-th spatial mode (singular vector), φ_q^i is the i-th mode of parameter $1 \leq q \leq p$, $w_{k_0 \ldots k_p}$ is the weight associated with the specific combination of modes $(k_0, ..., k_p)$. Finally, $\boldsymbol{r} = (r_0, ..., r_p)$ is the truncation rank. This separated form also enables *efficient storage* of the training data if $N = \prod_i r_i << N_{samples}$ with a controlled approximation error. This is particularly useful when the number of parameters is large, then, the exponential growth of sample data is not reflected in the surrogate model which can retain a low rank. Hence, the comparison between NN surrogates and MI is focused on this HOSVD + interpolation approach that is called Map-Decompose-Interpolate (MDI).

Interpolation and Reconstruction This is the first *online* step of the surrogate model. In order to evaluate the model at target parameter $\hat{\boldsymbol{\mu}} \in P$, is evaluated

$$\begin{aligned} \tilde{u}((\bar{\boldsymbol{x}}, \hat{\boldsymbol{\mu}}) &= \tilde{u}((\bar{\boldsymbol{x}}, \hat{\mu}_1, ..., \hat{\mu}_p)) && (12) \\ &= \sum_{k_0=1}^{r_0} \cdots \sum_{k_p=1}^{r_p} w_{k_0 \ldots k_p} X^{k_0}(\bar{\boldsymbol{x}}) \varphi_1^{k_1}(\hat{\mu}_1) \cdots \varphi_p^{k_p} \hat{\mu}_p) && (13) \end{aligned}$$

for $\bar{\boldsymbol{x}} \in \Omega_{ref}$. The modes $\varphi_q^i(\hat{\mu}_q)$ evaluated at the target parameter can be interpolated using any 1D interpolation technique.

Finally, the mean field is added back,

$$u_{HOSVD+I}(\bar{\boldsymbol{x}}), \hat{\boldsymbol{\mu}}) = \tilde{u}_{MI}((\bar{\boldsymbol{x}}, \hat{\boldsymbol{\mu}}) + \langle \bar{u} \rangle$$

where $\langle \bar{u} \rangle$ is the mean field on snapshots i.e. the function associated with $\langle \bar{U} \rangle$ from eq. 9.

The following methods were tested: piece-wise polynomials, splines and Lagrange interpolation and found they all performed well with similar efficiency. For the results reported the paper, piece-wise linear interpolation was used.

5. Results

In this section, the MLP, CNN, and MI surrogate models are applied to familes of part geometries consisting of 1) a plate with a hole of varying shape and 2) a set of wedge-like geometries. The performance of each model is evaluated for the prediction of 1) the von Mises residual stresses (scalar), 2) the displacement magnitude (scalar), and 3) the displacement vector (3 component vector). In all figures in this paper, the units for residual stress are taken to be Pa and the units for displacement are m.

5.1. *2D Plate geometry*

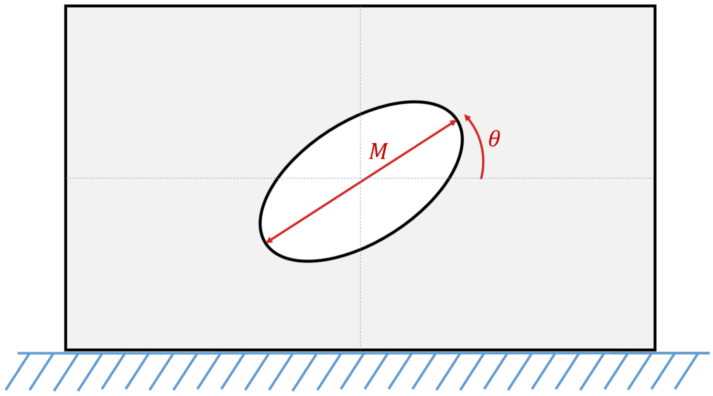


Figure 5.: A rectangular plate geometry with an ellipsoidal hole parameterized by the major axis of the ellipse M and the tilt angle from the horizontal axis θ.

Data description

The plate geometry considered consists of a rectangular plate with an ellipsoidal hole in its center, as illustrated in Fig. 5. As the thickness ($l_y = 6$mm) is very small compared to the other dimensions $l_x = 100$ and $l_z = 50$ mm, only 1 voxel is used in this dimension and are able to treat the problem as 2D using MLP and CNN. However, MI treats the plate geometries as a 3D model and so treats this problem as quasi-2D.

The hole is parameterized using $M \in [10, 20]$ mm, the major axis length, and $\theta \in [0, \pi[$, the tilt angle of the major axis relative to axis x. The minor axis is fixed to be 10mm, so that $M = 10$mm corresponds to a circular hole of radius 10mm. The data is systematically sampled over the parameter space $\mu = (M, \theta) \in [10 : 20 : 1] \times [0 : \pi : \pi/12]$ for a total number of 144 snapshots as shown in 6.

The training and validation strategy is different for the three methods because of their different requirements.

- As MLP is a point-wise model, 50% of the point data of the snapshots is a large enough dataset for training. Hence, for each snapshot, half of the point data is randomly sampled for training, while the remaining half is used for validation of the model.
- CNN uses the entire snapshot data for training. No extra validation set is used in this study and the hyperparameter tuning was performed based on training loss.

Consequently, the validation set in table 4 and 5 refers the training snapshots.

- MI also uses the entire snapshot data for training. Validation is unnecessary, as this model is deterministic. Consequently, the validation set in Table 4 and 5 refers the training snapshots.

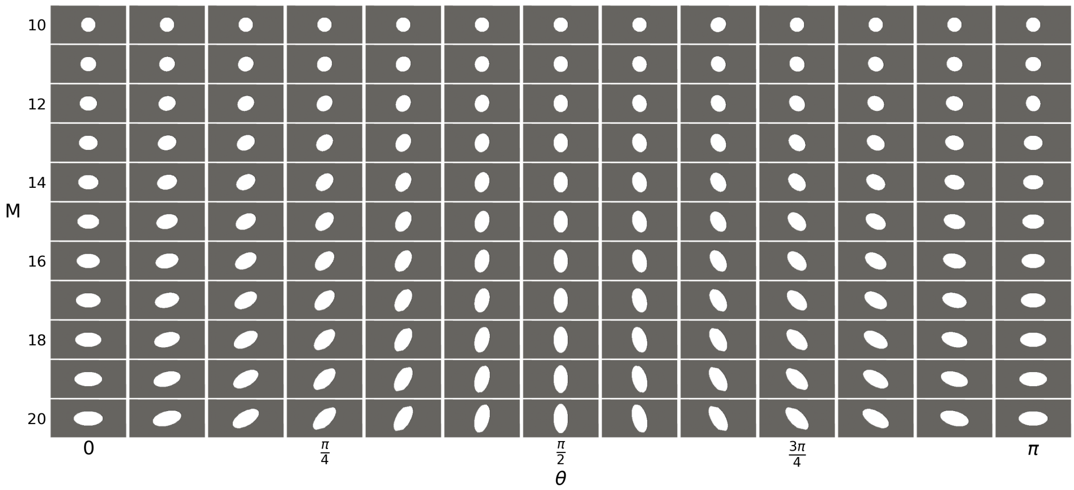
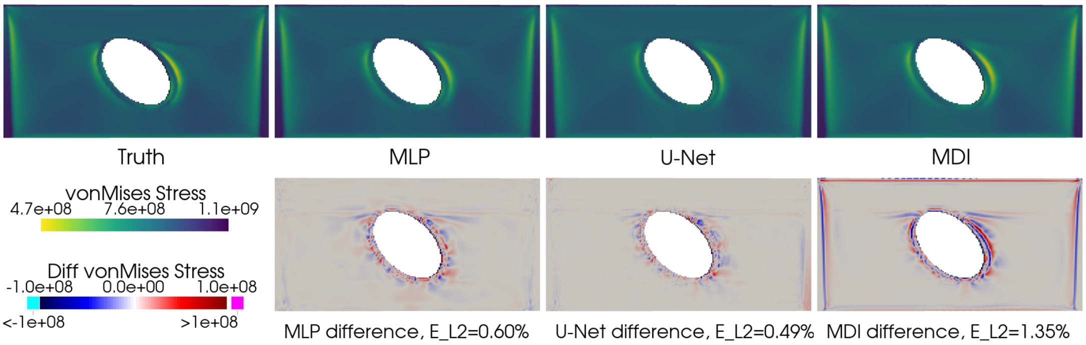


Figure 6.: Overview of the training set for plate with hole problem with parameters $(M, \theta) \in [10 : 20 : 1] \times [0 : \pi : \pi/12]$

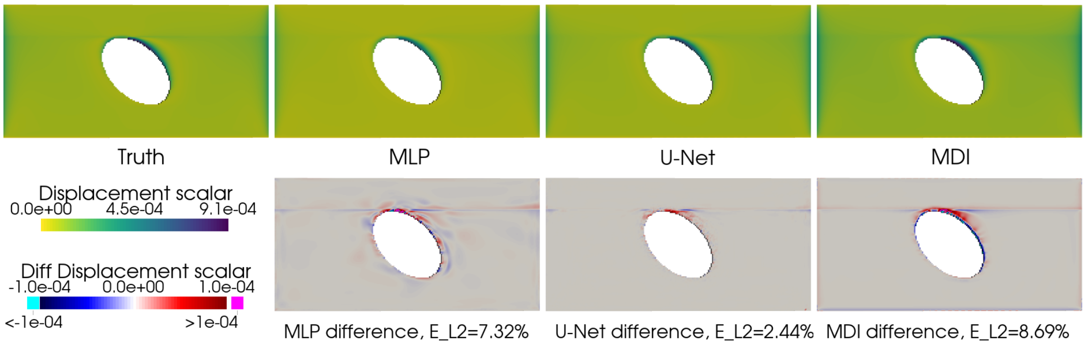
Surrogate modeling

Fig. 7 presents a comparison of all three methods for $M = 15.6$ and $\theta = 2.3736$. The FOM fields on the top line followed by the difference ($u_{diff} = u - \hat{u}$) plot for each model in the associated field. All three surrogate modeling methods perform similarly well in terms of prediction accuracy for the von Mises stresses. However, displacement proves more challenging and U-Net CNN architecture performs notably better. In all cases, it is difficult to distinguish differences between the predictions and the exact solution (FOM). Tab. 4, reports the training (validaiton) and test set average error. They confirm the visual impression from Fig. 7 unique sample.

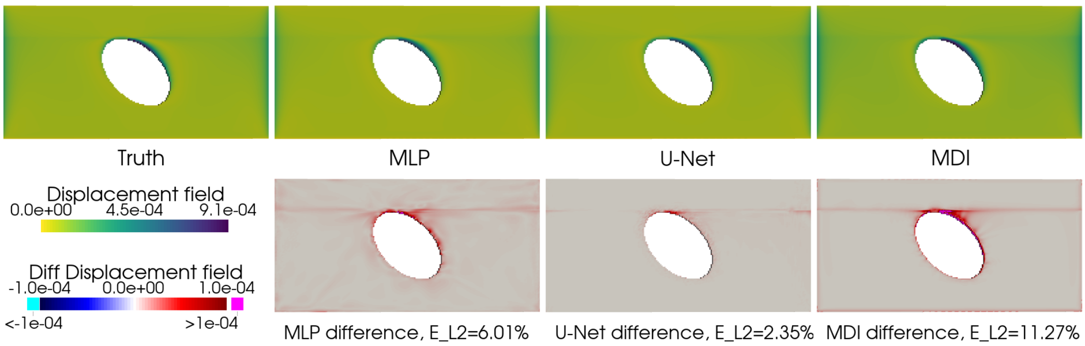
MLP surrogate model. The MLP is trained by retaining 50% of the point data for one field e.g. von Mises stresses. Training is performed over 500 epochs with a learning rate of 5e-3 and reduced upon plateauing (Bengio 2012) until a minimum value of 1e-6 is reached. Training required minutes of computational time (see Table



(a) von Mises stresses surrogate models



(b) Displacement magnitude surrogate models



(c) Displacement vector surrogate models

Figure 7.: Comparison of results for surrogate models of plate geometries. In each subfigure, the top line presents the models outputs alongside the corresponding "truth" with the same colorscale. The bottom line depicts the difference between the predicted values and the truth solution. The L_2 relative error of each prediction is indicated below each frame.

8). As shown in Table 4, this model performs well for all fields in L_2 relative error while L_∞ error is much larger because a few (surface) points have locally large error. This is a problem found with all models and can be attributed to the voxel structured mesh, which can exhibit non-physical steps on smooth surfaces. Similarly, the error on stresses ($\approx 0.7\%$) is one order of magnitude smaller than that of displacement ($\approx 8\%$) both vector and magnitude. This will be observed for all surrogates since these two fields have very different smoothness properties. Fig. 7 shows the stress field is well captured by this method, except for a region showing (mild) discrepancy is above the hole. Here, the surrogate stress field is smoother than the one from the FOM, resulting in both positive and negative error in this area.

Table 4.: Comparison of the MLP, CNN and MDI surrogate models error on three fields for both training/validation and test set on **plate** geometry.

		MLP		CNN		MDI	
Variable	Set	E_{L_2} (%)	E_{L_∞} (%)	E_{L_2}(%)	E_{L_∞} (%)	E_{L_2} (%)	E_{L_∞} (%)
Stress	Val.	0.71	25.3	0.49	28.3	0.91	22.7
	Test	0.68	11.68	0.5	9.81	1.36	23.3
Disp. scalar	Val.	8.30	44.9	3.21	49.9	8.98	40.2
	Test	7.98	47.8	3.62	23.3	11.5	55.7
Disp. vector	Val.	9.03	121	4.50	50.7	11.39	42.0
	Test	7.85	52.1	3.53	23.2	14.1	56.4

U-Net CNN. The U-Net CNN also provides a low error approximation with $E_{L_\infty} = \mathcal{O}(10^{-1})$ and $E_{L_2} \approx 0.5\%$ for residual stresses as shown in Table 4. Likewise, the performance error for displacement fields is higher by an order of magnitude. The more accurate results from U-Net CNN are very encouraging although this method requires more effort during training, particularly in hyper-parameter optimization (see Table 3 for complete description of the architecture). Another peculiarity of this method is the lack of smoothness of the output fields, which is particularly visible on the error distribution for displacement field. Indeed, the displacement is very concentrated in the upper portion of the hole, leading to large local error. Additionally, one can notice that the U-net is the only model which reasonably captures the “linelike” feature that is seen through the part at the top of the hole level.

MI surrogate model This section is focused on the more involved MDI (HOSVD+interpolation) version of MI as it is the most accurate and adaptable. In addition, it uses full rank decomposition in eq. 11 which preserves all the information of the training set with machine precision. Tab. 4 shows that MDI prediction accuracy follows the same pattern as MLP and Unet-CNN. Yet, one can see that its test set error is the largest of the surveilled models. In particular, one can see that displacement prediction error is above 10% which may be considered too high for use in production.

To further illustrate this method, Fig. 8 is provided. This figure shows the two states of prediction that are used for this method. Indeed, panel (a) shows the actual output of MDI which is a stretched mesh as one can see at the hole edges. In this case, the prediction closely matches the ground truth and prediction error is very small. However, to have a fair comparison with the other methods presented here, the surrogate model output is projected onto the same mesh as the FOM solution panel (c). This introduces extra error which is of the same magnitude as the total error for the model reported in this paper. The poor performance of this model is further studied in section 6.1.

Finally, one can observe in Fig. 7 that the error distribution of MDI is very different from MLP and Unet-CNN while retaining the same difficulty to capture displacement above the hole. Stresses error is concentrated near the edges of the geometry as clearly visible in subfigure (a).

5.2. *3D wedge geometry*

Compared to the pseudo-2D plate geometry studied in the previous section, the data structures of 3D geometries have higher complexity. Consequently, a family of bulky parts is introduced. They are used to test for potential differences in the accuracy and computational efficiency of the three methods.

Geometry description. The 3D geometry analyzed in this study is relatively simple and can be described as a wedge or a trapezoidal prism as shown in Fig. 9. The base of the prism is printed directly on the baseplate, where its length along y axis is fixed at $L_y = 50mm$. The base width $l_x \in [4, 20, 4]$, height $H \in [10, 30, 4]$ and top surface

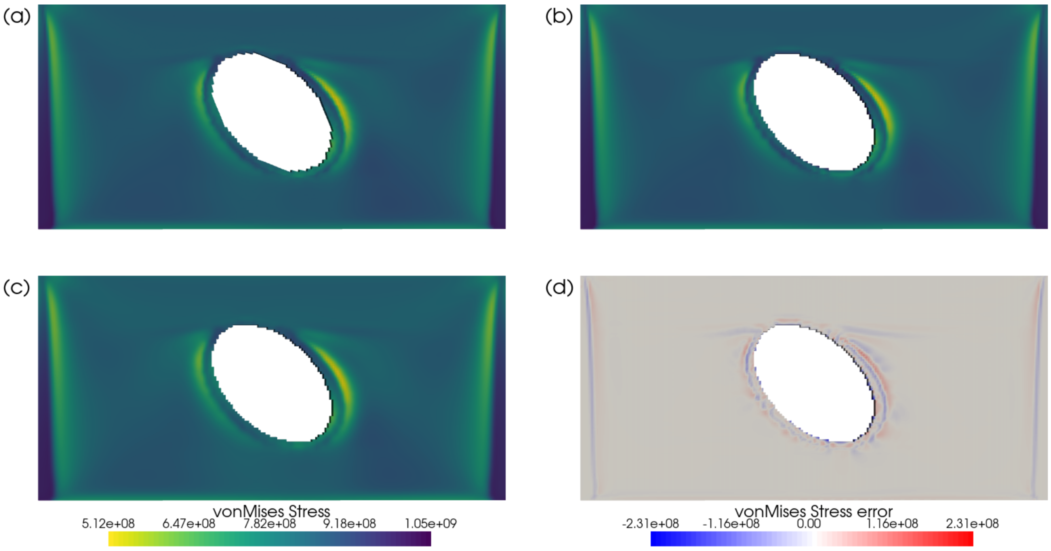


Figure 8.: Comparing MDI surrogate model output (a) with a test set sample computed with FOM (b) at $\mu = (15.6, 2.3736)$ for the von Mises stresses field. Panel (c) shows the projection of (a) onto FOM mesh (b) to compare results. Panel (a), (b) and (c) share the same vonMises stress colorbar. Panel (d) shows the difference map.

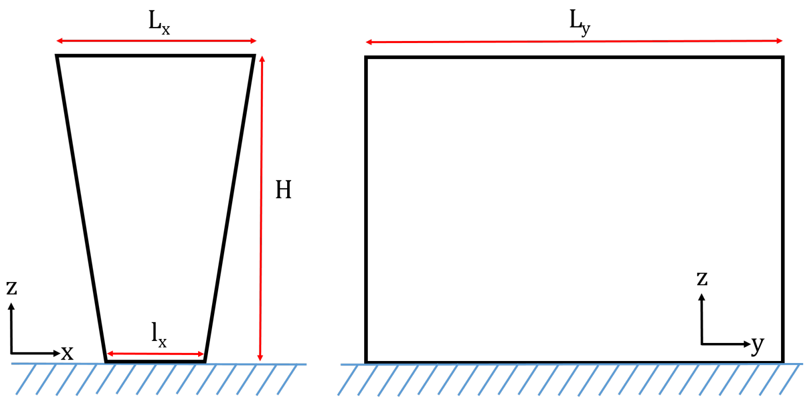


Figure 9.: Schematic view of the wedge parametrized geometry. Caution: L_y is not to scale.

width $L_x \in [4, 20, 4]$ are the 3 geometric parameters varied in this model. The range chosen for each parameter allows the study of a wide range of bulky geometries. A representative subset of the sampled geometries is shown in Fig. 10.

The parametric sampling is taken to be uniform since it enables easy implementation of MDI and does not affect the other methods. The total number of samples is $5 \times 5 \times 6 = 150$. The voxels are set with a uniform size on the whole set of samples with physical dimensions of (0.5mm $\times$ 5mm $\times$ 0.5mm) which was selected to describe efficiently the shape variations along x and z axis while limiting the total number of voxels using

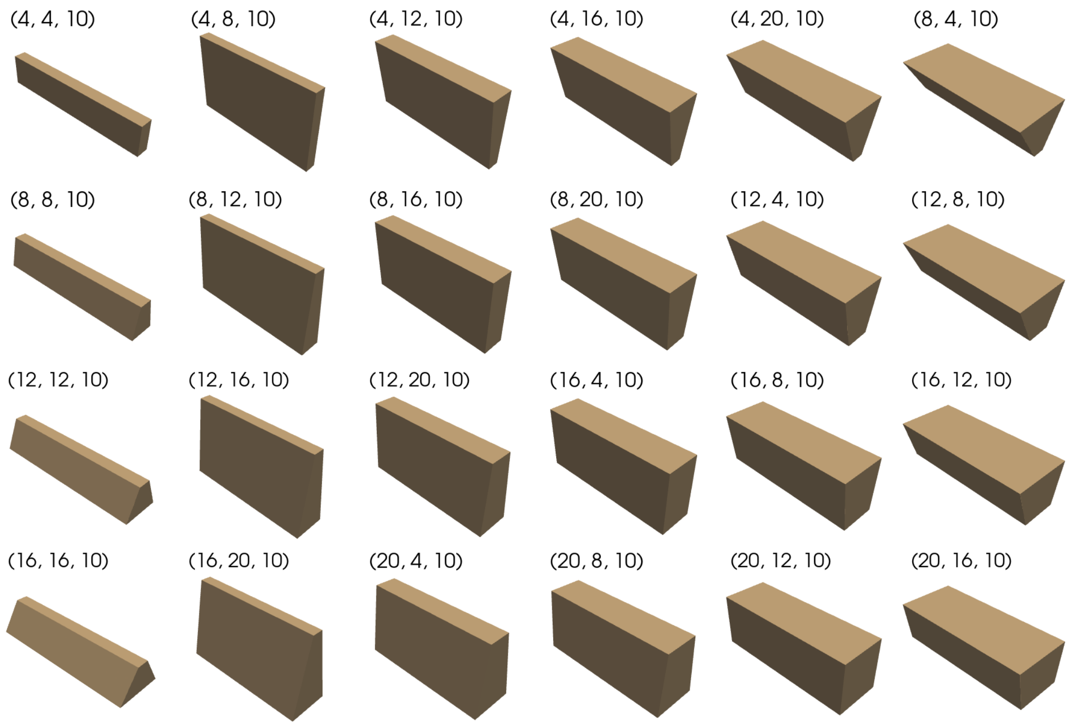


Figure 10.: Subset of the wedge geometry samples that are precomputed with the FOM.

coarse mesh along y axis. Mechanical properties are uniform for the whole part and can be found again in Table 1 with base plate stiffness that corresponds to half the printed material Young's modulus.

Surrogate modeling

MDI. After parametrizing the position of the control points on the surfaces and edges of the geometry, it was possible to apply the model seamlessly for all fields of interest as mapping has been directly implemented in 3D while decomposition (HOSVD) and interpolations methods are not affected by spatial dimension which they process as a single variable $\boldsymbol{x}$. Fig. 11 presents a comparison of MDI prediction (a) and the ground truth (b) for von Mises stresses. There, one can see that the surrogate model captures well the von Mises stress. To enable comparison with the ground truth as computed by FOM, MDI prediction is projected onto FOM mesh as shown in panel (c), thus introducing additional error as for the 2D case. The non-physical stepping observed on MDI output (a) is masked from the projected prediction while still generating sizeable error as the local error plot highlights in panel (d). This banded error is difficult to address for such voxel meshes which create increased error on the exterior of the domain on the FOM and translates into extra error in MDI. Additionally, some of the oscillations that are observed here are also visible in the interior of the domain on panel (d).

In Fig. 12, one can see that MDI produces the least accurate prediction for all fields surveyed, where errors are concentrated near the oblique surfaces of the wedge. In Table 5, one can notice that the displacement error fields are significantly larger than in the 2D example. This is likely because of the range of sloped surfaces exhibited by the wedge samples. Because of the voxel geometry, sloped surfaces are approximated by steps. Unlike the plate samples, where the sloped surfaces were centered around the hole feature, the wedge samples exhibit extended sloped features. This poses a challenge for MDI, where the smooth, ideal geometry must be mapped onto the voxel mesh. In contrast, MLP and CNN are inherently discrete, which means they better handle the stepped surface.

MLP proves very efficient at handling this set of 3D geometries. Training is efficient, as discussed in Sec. 6, while preserving a very high prediction accuracy. It performs better than MDI across all fields while remaining close to U-Net CNN as shown in Fig. 12. One can see that the error map is very different from MDI with error being less acute on the edges. Overall, results presented in Table 5 show that on both validation and test sets the relative error is very similar to U-Net CNN, while clearly being lower than MDI.

U-Net CNN difference map in Fig. 12 shows yet another error topology with higher accuracy near the edges of the part and very small error in the interior. Overall, for an homogeneous train/test set such as the one studied here, one can confidently say that U-net CNN produces reliable predictions so long as the voxelized grid is provided. This suggest that one can use this method for more complex geometries involving varying thicknesses and, more importantly, implicit parametrization since this network architecture only requires voxels. Finally, one can see in Table 5 that U-

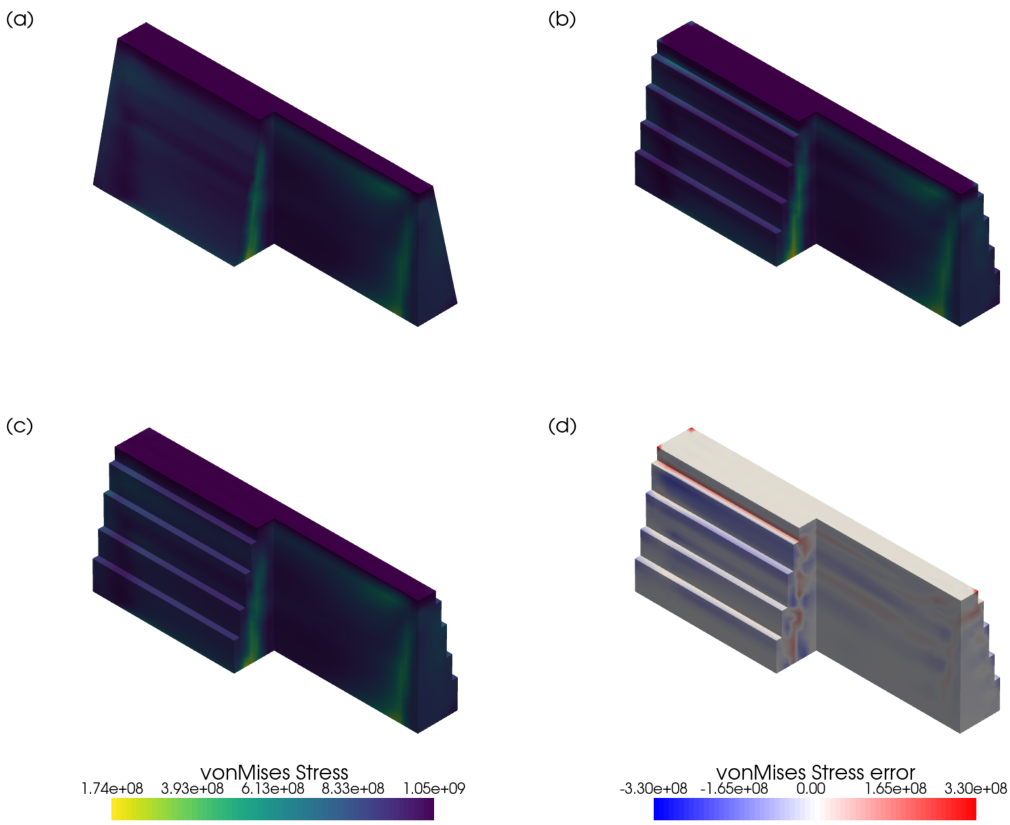
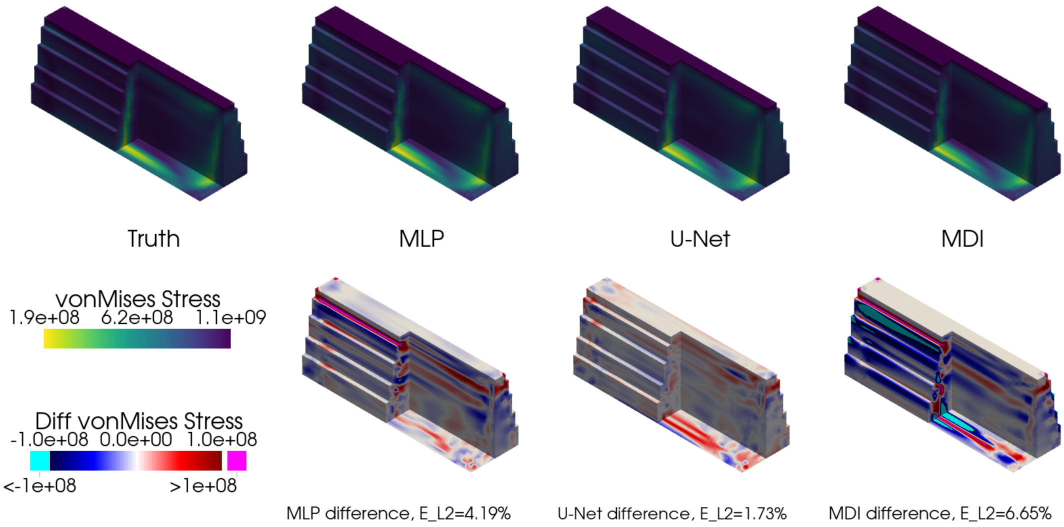


Figure 11.: Comparing raw MDI surrogate model (a) with a test wedge sample (14,5,20) computed with FOM (b) for the von Mises stress (scalar) field. Panel (c) shows the projection MDI output onto the FOM grid. Panel (a), (b) and (c) share the same von Mises stress colorbar. Finally, panel (d) shows the point-wise difference between (c) and (d) which corresponds to a 6.6% L2 error.

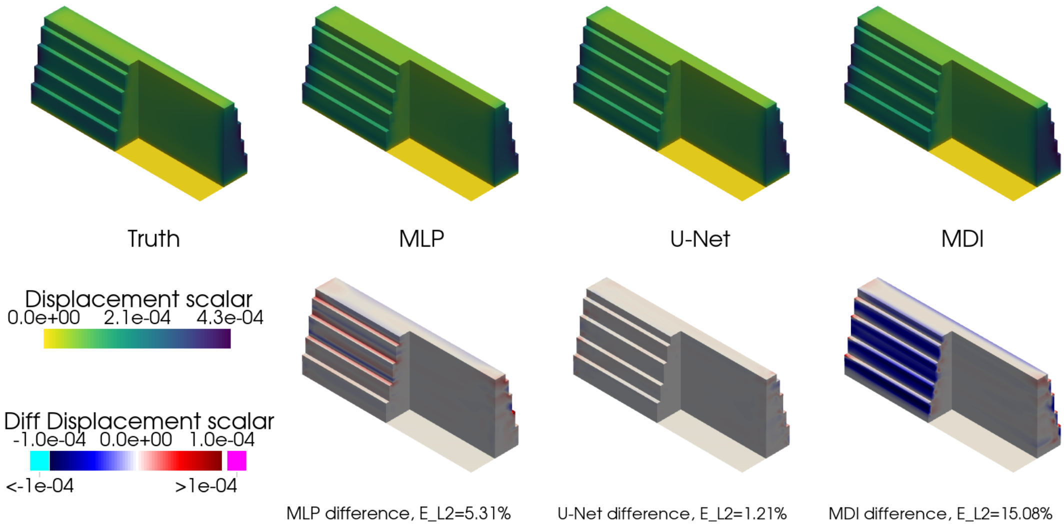
net CNN produces the lowest error on stress field at $E_{L2} = 3.25\%$ and more strikingly on the diplacement field at 3.68%. This can be attributed to the ability, by design, of the architecture to capture different scales of the volumetric data. Hence, the singular values at the bottom plate cause little to no interference with the rest of the domain.

6. Discussion

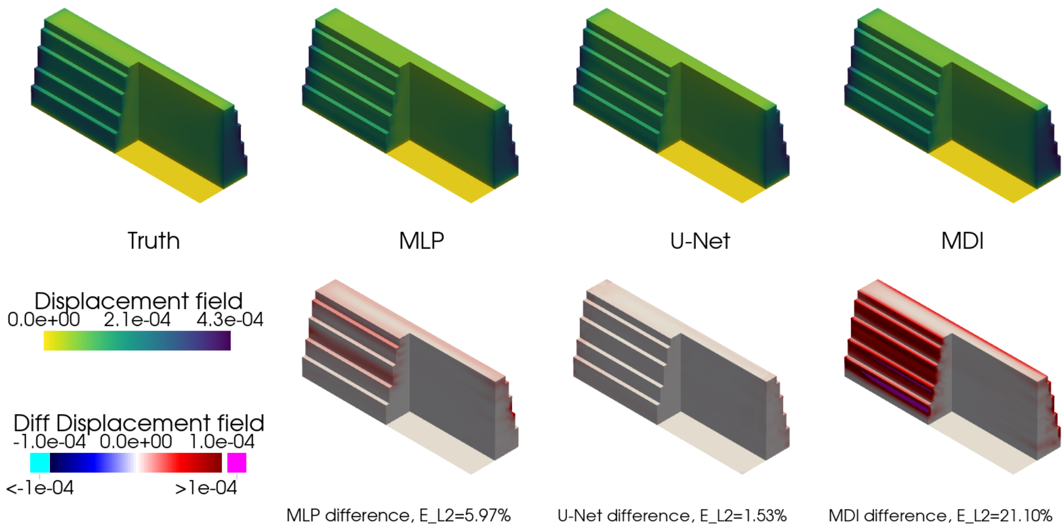
In the previous section, the three surrogate models have exhibited good predictive power with relative error within a few percent for the von Mises stresses. However, the displacement field has proved to be more difficult to predict, in particular for MDI. In section, a deeper analysis of the surrogate models capabilities is presented.



(a) von Mises stresses surrogate models



(b) Displacement magnitude surrogate models



(c) Displacement vector surrogate models

Figure 12.: Wedge test geometry (14,5,20) surrogate model comparison with point-wise difference between the surrogate models studied with FOM (truth) result shown for reference on the first column

Table 5.: E_{L_2} and E_{L_∞} error (%) comparison for the MLP, CNN and MDI models on three fields for both training/validation and test set on **wedge** problem.

		MLP		CNN		MDI	
Variable	Set	E_{L_2}(%)	E_{L_∞}(%)	E_{L_2}(%)	E_{L_∞}(%)	E_{L_2}(%)	E_{L_∞}(%)
Stress	Val.	1.83	18.7	1.25	50.0	3.94	24.7
	Test	3.41	14.1	3.25	14.38	5.63	25.2
Disp. magnitude	Val.	1.43	7.43	2.15	20.0	13.63	19.56
	Test	5.60	20.47	3.06	5.54	16.02	22.39
Disp. vector	Val.	1.18	11.3	1.97	22.2	19.40	21.5
	Test	7.65	22.4	3.68	9.06	22.75	23.68

6.1. *Error analysis of MI models*

In general, isolating individual error sources is an efficient approach to reduce the overall error of the surrogate model. In this section, a detailed analysis of error in the MI methods is performed on the *plate geometry only* (although the extension to the wedge geometry is straightforward), as it can be decomposed into clearly-identified steps of the algorithm and then analyzed mathematically.

6.1.1. Comparing submodels for MI

Table 6 shows the relative L2 error of three variants of MI presented in section 4 . MI models produce acceptable accuracy levels von Mises stress. As expected, nearest neighbors has the lowest accuracy, but only by a small margin. The displacement is only studied for scalar case as it is representative of the vector field. This one shows much larger error around 11% with large variation around the mean value, for instance displacement scalar error ranges from 7% to 16% for all models. The maximum error is observed for sample $\mu = (16.4, 2.34)$ for all methods which indicates that this particular geometry is difficult to handle irrespective of the interpolation scheme. This is attributed to the mapping procedure since sampling is fine and no major correlation has been observed between test error and distance to nearest training sample. It can be

variable	MDI	piecewise linear	nearest neigbors
von Mises stress	1.10	1.12	1.20
Displacement scalar	11.41	10.29	11.59

Table 6.: Comparison of the relative L2 error (%) for different interpolation methods in Map-Interpolate surrogate models. MDI method is computed with $\varepsilon = 10^{-2}$.

concluded that from an accuracy standpoint, the decomposition step does not improve the accuracy of the model significantly. However, decomposition is expected to become more useful when the number of dimensions of the data grows, as truncation may help filter high frequencies which makes the model smaller and so enables interpolation through standard libraries, similarly to PODI (Bui-Thanh, Damodaran, and Willcox 2004). For this reason, a detailed error analysis of MDI is proposed.

6.1.2. MDI error analysis

The total error, denoted as E_{tot}, is initially decomposed into three distinct components: the error arising from the mapping step, denoted as E_{Map}, the error due to low rank approximation, denoted as E_{RB}, and the interpolation error, denoted as E_{interp}.

$$E_{tot} = E_{Map} + E_{RB} + E_{interp} \tag{14}$$

The relative influence of each component is compared by evaluating the error generated at the respective step, excluding compensation. Therefore, Table 7 presents the results depicting e_{Map}, e_{RB}, and e_{interp}. These values represent the best estimates of the mapping error, low rank approximation error, and interpolation error achievable by instrumenting the code. The analysis was conducted on all test samples, and two representative samples, namely $\mu_{t1} = (15.6, 2.37)$ and $\mu_{t2} = (13.4, 1.71)$, are extracted and displayed in Table 7 for illustrative purposes.

Following section 4, a few notations are introduced. For μ, a test geometry (parameter), assume $u(\cdot;\mu)$ is known and its discrete representation U_μ^{truth} by FOM on a dedicated grid. The projection on the reference grid is $\bar{U_\mu}$ and $U_\mu = \bar{U_\mu} - \langle\bar{U}\rangle$ where $\langle\bar{U}\rangle$ is the mean field of the training data.

The values given in this detailed analysis of MDI are reported for the displacement (scalar) field since the error is larger. This makes the interpretation of the relative weight of the various contributions easier.

Mapping error. Mapping to and from the reference space is very important in terms of model quality since errors from this step are carried onto the model afterwards. Additionally, mapping is performed twice on the data, 1) on the training samples to build the model, 2) on the model in the reference space to the physical space. In order to get a finer understanding of the mapping error E_{Map}, it can be split into three contributions corresponding to the algorithm

$$E_{Map} \propto e_{Map} = e_{proj} + e_{mesh} + e_{STL} \tag{15}$$

where e_{proj} is the error associated with projecting data from one mesh to another with a Gaussian kernel; e_{mesh} is the error due to the mesh deformation quality; and e_{STL} is the error associated with voxelizing the free-form STL geometry

- **Projection:** The projection error e_{proj} can be evaluated by applying projection of data back and forth between the test sample FOM mesh and the reference reference mesh. Specifically,

$$e_{proj} = \frac{||U_\mu^{truth} - U_\mu^{proj\times 2}||_{L2}}{||U_\mu^{truth}||_{L2}} \tag{16}$$

  where $U_\mu^{proj\times 2}$ projects back onto U_μ^{truth} mesh. Doing so, its evaluation yields $e_{proj}(\mu_{t1}) = 6.82\%$. This indicates that the projection step accounts for a significant share (close to half) of the error observed on the MDI method despite the absence of modelling. The values reported here have been obtained after systematic optimization of the Gaussian kernel based interpolation parameters available through `VTK` API.
- **Mesh:**

  The mesh error e_{mesh} increases with the deformation of the mesh due to the use of the RBF algorithm, which can produce unphysical meshes with points outside the physical domain. To minimize this error, the mesh deformation is made stable by employing substepping of the deformation, and using the multi quadratic biharmonic spline kernel. As this surrogate model is data-driven, no equation is solved on the RBF mesh and e_{mesh} is not significant in this study as long as a physically coherent deformation is obtained. This has been demonstrated by systematic study of the radial basis set.

- **STL:** The voxelization of the geometry introduces a number of points outside the solid region generating extra interpolation/extrapolation error. To counteract this effect, the implementation projects these points onto the STL surface of the solid domain before the projection step. While it allows for cleaner and physically coherent mesh projection, no difference was observed in L2 error when skipping this STL projection step. This is attributed to the dominance of the mesh projection in the mapping error.

From this analysis, the authors conclude that the mapping error is dominated by the data projection i.e. $e_{map} \approx e_{proj}$.

In order to analyze E_{RB} and E_{interp}, in situ evaluation is performed by varying the tolerance of the HOSVD. However, since these error sources are coupled, it is not possible to account for these error sources independently. Instead, the evaluation is restricted to each step individually i.e. measuring e_{RB}, E_{HOSVD} and e^{x}_{interp} separately.

Low rank approximation error. The approximation error of the HOSVD, E_{HOSVD}, is displayed in Table 7. It is given by the L^2 norm of the reconstruction error, $E_{HOSVD} = ||U - \widetilde{U}||/||U||$ as defined in eq. (11). It can be seen that although varying the truncation parameter ε strongly affects E_{HOSVD}, E_{tot} is orders of magnitude smaller. This is due to the prominent weight of the mean field in the overall model. Additionally, the denominator of E_{HOSVD} is different from the one used to compute E_{tot}. Consequently, a better way to evaluate the weight of the low rank approximation is selected. Here, the focus is on the ability of the HOSVD reduced basis (RB) to represent the truth data U_μ and use the same normalization factor as for E_{tot}. Hence, the following RB error is used,

$$E_{RB} \propto e_{RB} = \frac{||U_\mu - U^{RB}_\mu||_2}{||\bar{U}_\mu||_2} \tag{17}$$

where $U^{RB}_\mu = \boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}} U_\mu$ is the projection of U_μ onto the spatial HOSVD basis $\boldsymbol{X}$.

In Table 7, one can see that this estimate of the error fits well with the evolution of the total error. For both geometries, one can see that e_{RB} is driving the error when ε

is large while it plateaus at $e_{RB} \approx [3\%, 2.5\%]$ for $\varepsilon \leq 10^{-4}$, thus limiting the overall accuracy of the MDI model. This is in stark contrast with the direct measure of the HOSVD error, which can be reduced to machine error when full rank is reached. Note that here, because of data augmentation, the actual number of non-collinear snapshots is 121, meaning full rank is reached for $\varepsilon = 10^{-6}$ as shown in the Tucker rank column with (121,11,13) corresponding to (space, M,θ) modes. The last column of Table 7 reports the model size relative to the training data set. For instance, it shows that for μ_{t1} a MDI model obtained with $\varepsilon = 10^{-2}$ requires 9% of the storage/memory of the full data set while being able to predict the displacement scalar field with $E_{tot}^{lin} = 7.8\%$ (1.08% for von Mises stresses) . This is close to the best accuracy for this test sample, despite suboptimal RB representation. This is attributed to interpolation error as discussed next.

Interpolation error. E_{interp}^{x} is estimated by measuring its standalone effect e_{interp}^{x}. Indeed, it enables comparison of the interpolation $\tilde{u}(\bar{x}, \mu)$ of eq. (13) (discrete $\widetilde{U}_{\mu}$) with the projection of the exact solution on the RB obtained above U_{RB} which is the best representation of U_{μ} possible,

$$E_{interp}^{x} \propto e_{interp}^{x} = \frac{||\widetilde{U}_{\mu} - U_{\mu}^{RB}||_2}{||\bar{U}_{\mu}||_2} \tag{18}$$

First, one can see in Table 7 that choosing between quadratic or linear interpolation has little effect on the MDI total error, accounting for roughly 0.1%. Looking at e_{interp}^{x} specifically, the same pattern is observed. Interestingly, the interpolation error is small for very low rank approximation and grows until it becomes the largest component of the error. This can be explained by the increasing oscillations in the M and θ modes, indicating an inability to generalize, consistent with observations in (Lestandi 2021).

In conclusion, the breakdown of the error shows that, provided adequate tuning of the MDI steps, each component has a comparable share of the global error at around few percent and always remain smaller than the mapping error. The total error E_{tot}^{x} exhibits compensation between steps, resulting in overall lower error than the sum of its individual components. This analysis suggests it is difficult to substantially improve the accuracy of MDI, in particular because mapping error which represents a hard limit to the MDI surrogate model accuracy. Despite these limitations, the explicability of the error can help refine the method for future work. In this study, it justifies using a higher value of the HOSVD truncation ε criterion e.g. 10^{-3}, to minimize the size of the model while maintaining good accuracy.

Table 7.: MDI error break-up analysis on plate with hole example with two representative geometries from the test set for *displacement (scalar) field*. Various HOSVD tolerances ε are used to evaluate the error of the MDI model E_{tot}^{X} with two interpolation methods $X \in \{$"linear", "quadratic"$\}$. An estimate of the components of the error e_{RB}, E_{map} and e_{interp}^{X} is provided as well as specific HOSVD error E_{HOSVD}, Tucker rank and the model relative size compared to the sample data set (called compression rate in (Lestandi 2021)).

(a) $\mu = (15.5, 2.37)$

ε	E_{tot}^{lin}	E_{tot}^{quad}	e_{RB}	e_{map}	e_{interp}^{int}	e_{interp}^{quad}	E_{HOSVD}	Tucker rank	Relative size
1.0e-01	9.68%	9.85%	6.67%	7.30%	3.38%	3.53%	38.39%	(4, 1, 3, 5)	2.80%
1.0e-02	7.80%	7.77%	5.67%	7.30%	3.73%	3.80%	24.44%	(13, 1, 6, 9)	9.11%
1.0e-03	7.76%	7.82%	4.42%	7.30%	5.08%	5.29%	8.89%	(41, 1, 11, 13)	28.78%
1.0e-04	7.67%	7.78%	3.15%	7.30%	5.88%	6.15%	2.54%	(93, 1, 11, 13)	65.28%
1.0e-06	7.67%	7.79%	2.93%	7.30%	6.00%	6.28%	0.00%	(123, 1, 11, 13)	86.34%

(b) $\mu = (13.4, 1.71)$

ε	E_{tot}^{lin}	E_{tot}^{quad}	e_{RB}	e_{map}	e_{interp}^{int}	e_{interp}^{quad}	E_{HOSVD}	Tucker rank	Relative size
1.0e-01	7.78%	8.07%	5.27%	6.11%	1.32%	1.67%	38.39%	(4, 1, 3, 5)	2.80%
1.0e-02	7.05%	7.34%	4.54%	6.11%	1.51%	1.90%	24.44%	(13, 1, 6, 9)	9.11%
1.0e-03	6.50%	6.64%	3.30%	6.11%	2.75%	3.01%	8.89%	(41, 1, 11, 13)	28.78%
1.0e-04	6.46%	6.66%	2.72%	6.11%	3.31%	3.63%	2.54%	(93, 1, 11, 13)	65.28%
1.0e-06	6.45%	6.64%	2.55%	6.11%	3.43%	3.75%	0.00%	(123, 1, 11, 13)	86.34%

Table 8.: Comparison of the MLP, CNN and MDI surrogate models CPU (wall) times for training and evaluation of the von Mises stress field. FOM typical CPU walltimes were 1330s for plate with hole and 270s for wedge geometries. MDI projections steps were implemented in two versions, pure CPU and combined with GPU for 3D cases.

	Plate with hole (2D), $\mu = (16.0, 0.96)$			Wedge (3D)		
	MLP	Unet-CNN	MDI (CPU only)	MLP	Unet-CNN	MDI (CPU+GPU)
Training time (Offline)	600s	480s	44s	1200s	1200s	47s
Evaluation time	25ms	33ms	133ms	24ms	38ms	68ms
Speedup (w.r.t. FOM)	53,200	40,300	10,000	11,250	7,110	3,970

6.2. *Computation time*

The three methods reported in this paper have comparable accuracy for both quasi-2D plate geometries and 3D wedge geometries. However, some differences were observed in the training and evaluation time for these surrogates, as shown in Table 8. The training time for all models is relatively short compared to the time required to run the FOM. Likewise, all the run times show speed-up of at least 1000×, reflecting the suitability of these methods as surrogate models.

Although it is difficult to directly compare the computation time for different methods due to differences in implementation, analysis of the training times gives indications for the relative suitability of the various surrogate models. In spite of the neural network models being implemented using the very efficient `TensorFlow` library, the CNN and MLP models required relatively long training times. In contrast, while the MDI model utilized relatively unoptimized code (including interpreted python scripts), the deterministic training strategy still was faster than either MLP or CNN. The efficient training process illustrates one potential advantage for interpolation-based algorithms.

6.3. *Sparse sampling*

In this subsection, the effect of sparsely sampled parametric space is investigated. Indeed, in many cases, when the number of parameters increases, it becomes prohibitively expensive to generate densely sampled training space. To emulate this behaviour, a sparse sampling grid is used with only 4, 9 and 30 samples evenly spaced in the same parametric domain for plate with hole and compare Unet-CNN, MLP and MDI surrogate models. The results are presented in Tab. 9. As one would expect, for all models the error decreases as the number of samples increases, $n = 30$ still exhibits a larger error than the full training set for MLP and Unet-CNN. MDI is proportionately the least affected by the reduction of the training set. This can be attributed, in part, to the larger final error of MDI but also to its nature, since the level of details captured by this method does not require very fine sampling. At the other end of the spectrum, MLP is the most affected by sparse sampling as it is simply difficult to train this kind of network with very few data points. Finally, the Unet-CNN performance is very good with as little as 4 samples, which is remarkable since this kind of network usually requires very large training set in the context of image processing.

Table 9.: **Plate** geometry test set average E_{L_2} error (%) of the MLP, CNN and MDI surrogate models trained with sparse sampling strategy i.e. with n training samples.

n	Variable	MLP	CNN	MDI
4	von Mises stress	5.95	2.27	2.70
	Disp. scalar	42	16	25
9	von Mises stress	3.28	1.68	1.79
	Disp. scalar	23	14	20
30	von Mises stress	1.42	0.93	1.35
	Disp. scalar	13.8	6.10	11.5

6.4. *Assessment of surrogate modeling methods*

As shown in Sec. 5, Unet-CNN method is clearly better in terms of accuracy, even for sparse training set. However, there are still differences in their training. For example, although the U-Net can be applied to data with no explicit parameterization (Dong et al. 2022), it can only be utilized on the specific voxel mesh it has been trained on. In addition, it is the most computationally expensive to train (in general). On the other hand, MLP does not require a mesh, but instead requires an explicit parameterization of the family of geometries. It may be trained much more efficiently than the U-Net surrogate, as it does not require the whole dataset (in space) to be trained and the number of epochs can be reduced.

Similarly to MLP, MDI does not require a mesh but requires the definition of control points to map the geometry onto a reference. It consequently requires the most detailed parameterization of the surrogate models in the present study. It has also proven to be the least accurate method for dense training set. However, since the low-order decomposed modes calculated in the MDI method are deterministic and have a physical interpretation (Lestandi 2018), this method can potentially lead to additional understanding of the surrogate model as demonstrated by our error analysis. In addition, it is straightforward to apply the model to different fields, scalar or vector without any additional tuning.

Finally, the storage of the model for the two problems studied is very similar with a small weight of $\approx 10Mb$ which is roughly the size of a few FOM output samples.

Each of the three surrogate strategies considered in this study are suitable for optimization - however, in general, the computational cost of training decreases with increasing complexity of the shape parameterization.

7. Conclusion

In this paper, three surrogate modeling methods for part-scale mechanical simulation of LBPF printed parts have been presented. They enable extremely fast run-time compared to the full order model. These surrogate models allow fast analysis of different part shapes by parameterizing them with various geometric features. Two of these methods are based on different neural network architectures, specificially the multilayer perceptron (MLP) and the convolutional neural network (CNN). The third method uses non-linear mapping to enable linear interpolation on the tangent space, which was described as Map-Interpolate (MI). More specifically, a variant involving tensor decomposition (MDI) has been used in this study. MLP, CNN and MDI are fully data-driven methods and can be applied to any problem requiring parametrized geometry interpolation regardless of the complexity of the underlying physics.

It was shown that all three methods provide good accuracy for part-scale LPBF with error $< 3\%$ for the von Mises stress and $< 12\%$ for displacement, and so these methods are all suitable incorporation into shape optimization processes. Surprisingly, even the relatively simple MLP architecture was able to model the simulation data well. In spite of CNN's generality, it suffers from a significantly longer training time for CNN. Lastly, conventional interpolation schemes such as MDI are trained efficiently compared to the neural network based methods while struggling with highly singular data.

Finally, the authors found that a key issue in developing surrogate models for part-scale additive manufacturing models arise from the choice of domain discretization in the LPBF simulation workflow. Specifically, while the simulation is performed on a voxel mesh, the geometry parameterization in this work utilizes smooth functions.

Although this difference can introduce significant error to the regression scheme, it can be accounted for by using mesh projection to map from one mesh to another or by using a voxel-based feature parameterization.

Conflict of interests

The authors declare they have no conflict of interest.

8. Data availability

The authors wish to contribute to the further development of surrogate models for LPBF in the community. Consequently the data is available at

Funding

Acknowledgement

References

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. "TensorFlow: A system for large-scale machine learning." *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016* https://doi.org/10.5555/3026877.3026899.

Ammar, Amine, Antonio Huerta, Francisco Chinesta, Elías Cueto, and Adrien Leygue. 2013. "Parametric solutions involving geometry: A step towards efficient shape optimization." *Computer Methods in Applied Mechanics and Engineering* 268: 178–193. https://doi.org/10.1016/j.cma.2013.09.003.

Ansell, Troy Y., Joshua P. Ricks, Chanman Park, Chris S. Tipper, and Claudia C. Luhrs. 2020. "Mechanical properties of 3D-printed maraging steel induced by environmental exposure." *Metals* 10: 1–11. https://doi.org/10.3390/met10020218.

Ballani, Jonas, Lars Grasedyck, and Melanie Kluge. 2010. "Black Box Approximation of Tensors in Hierarchical Tucker Format." *Linear algebra and its applications* 438: 639–657. https://doi.org/10.1007/s10955-007-9289-x.

Ballarin, Francesco, Alessandro D'Amario, Simona Perotto, and Gianluigi Rozza. 2019. "A POD-selective inverse distance weighting method for fast parametrized shape morphing." *International Journal for Numerical Methods in Engineering* 117: 860–884. https://doi.org/10.1002/nme.5982.

Baturynska, Ivanna. 2019. "Application of machine learning techniques to predict the mechanical properties of polyamide 2200 (PA12) in additive manufacturing." *Applied Sciences (Switzerland)* 9. https://doi.org/10.3390/app9061060.

Bengio, Yoshua. 2012. "Practical recommendations for gradient-based training of deep architectures." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7700 LECTURE NO. https://doi.org/10.1007/978-3-642-35289-8_26.

Biancolini, Marco Evangelos, Andrea Chiappa, Ubaldo Cella, Emiliano Costa, Corrado Groth, and Stefano Porziani. 2020. *Radial Basis Functions Mesh Morphing.* Vol. 1. Springer International Publishing. http://dx.doi.org/10.1007/978-3-030-50433-5_23.

Botsch, Mario, and Leif Kobbelt. 2005. "Real-time shape editing using radial basis functions." *Computer Graphics Forum* 24: 611–621. https://doi.org/10.1111/j.1467-8659.2005.00886.x.

Bui-Thanh, T., M. Damodaran, and K. Willcox. 2004. "Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition." *AIAA Journal* 42.

https://doi.org/10.2514/1.2159.

Chen, Junfeng, Jonathan Viquerat, and Elie Hachem. 2019. "U-net architectures for fast prediction of incompressible laminar flows." http://arxiv.org/abs/1910.13532.

Chinesta, Francisco, Adrien Leygue, Felipe Bordeu, Elias Cueto, David Gonzalez, Amine Ammar, and Antonio Huerta. 2017. "PGD-Based Computational Vademecum for Efficient Design , Optimization and Control." *Archives of Computa- tional Methods in Engineering* 20: 31–59. https://doi.org/10.1007/s11831-013-9080- x.

Chowdhury, Sushmit. 2016. "Artificial Neural Network Based Geometric Compensation for Thermal Deformation in Additive Manufacturing Processes." .

Dahmen, Wolfgang, Ronald DeVore, Lars Grasedyck, and Endre Süli. 2015. "Tensor-Sparsity of Solutions to High-Dimensional Elliptic Partial Differential Equations." *Foundations of Computational Mathematics* https://doi.org/10.1007/s10208-015-9265-9.

de Boer, A., M. S. van der Schoot, and H. Bijl. 2007. "Mesh deformation based on radial basis function interpolation." *Computers and Structures* 85: 784–795. https://doi.org/10.1016/j.compstruc.2007.01.013.

Demo, Nicola, Marco Tezzele, Andrea Mola, and Gianluigi Rozza. 2018. "An efficient shape parametrisation by free-form deformation enhanced by active subspace for hull hydrodynamic ship design problems in open source environment." *Proceedings of the International Offshore and Polar Engineering Conference* 2018-June: 565–572.

Demo, Nicola, Marco Tezzele, and Gianluigi Rozza. 2019. "A non-intrusive approach for the reconstruction of POD modal coefficients through active subspaces." .

Dong, Guoying, Jian Cheng Wong, Lucas Lestandi, Jakub Mikula, Guglielmo Vastola, Mark Hyunpong Jhon, My Ha Dao, Umesh Kizhakkinan, Clive Stanley Ford, and David William Rosen. 2022. "A part-scale, feature-based surrogate model for residual stresses in the laser powder bed fusion process." *Journal of Materials Processing Technology* 304. https://doi.org/10.1016/j.jmatprotec.2022.117541.

dung Nguyen, Viet, Karim Abed-meraim, and Linh trung Nguyen. 2016. "Fast Tensor Decompositions for Big Data Processing." 215–221.

Duong, Pham Luu Trung, Shaista Hussain, Mark Hyunpong Jhon, and Nagarajan Raghavan. 2020. "Data Driven Prognosis of Fracture Dynamics Using Tensor Train and Gaussian Process Regression." *IEEE Access* 8. https://doi.org/10.1109/ACCESS.2020.3042830.

Gan, Zhengtao, Hengyang Li, Sarah J. Wolff, Jennifer L. Bennett, Gregory Hyatt, Gregory J. Wagner, Jian Cao, and Wing Kam Liu. 2019. "Data-Driven Microstructure and Microhardness Design in Additive Manufacturing Using a Self-Organizing Map." *Engineering* 5: 730–735. https://doi.org/10.1016/j.eng.2019.03.014.

Ghnatios, Chady, Elias Cueto, Antonio Falco, Jean Louis Duval, and Francisco Chinesta. 2021. "Spurious-free interpolations for non-intrusive PGD-based parametric solutions: Application to composites forming processes." *International Journal of Material Forming* 14: 83–95. https://doi.org/10.1007/s12289-020-01561-0.

Glorot, Xavier, and Yoshua Bengio. 2010. "Understanding the difficulty of training deep feedforward neural networks." Vol. 9.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*.

Hesthaven, Jan S, and Stefano Ubbiali. 2018. "Non-intrusive reduced order modeling of nonlinear problems using neural networks." *J. Comput. Phys.* 363: 55–78. https://doi.org/10.1016/j.jcp.2018.02.037.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. "Multilayer feedforward networks are universal approximators." *Neural Networks* https://doi.org/10.1016/0893-6080(89)90020-8.

Iglovikov, Vladimir, and Alexey Shvets. 2018. "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation." https://arxiv.org/abs/1801.05746v.

Joshi, Kartikey, Siu Sin Quek, Yingzhi Zeng, and David T. Wu. 2021. "An efficient implementation for the solution of auxiliary composition fields in multicomponent phase field models." *Computational Materials Science* 197. https://doi.org/10.1016/j.commatsci.2021.110608.

Kingma, Diederik P., and Jimmy Lei Ba. 2015. "Adam: A method for stochastic optimization." .

Kizhakkinan, Umesh, Pham Luu Trung Duong, Robert Laskowski, Guglielmo Vastola, David W. Rosen, and Nagarajan Raghavan. 2023. "Development of a surrogate model for high-fidelity laser powder-bed fusion using tensor train and gaussian process regression." *Journal of Intelligent Manufacturing* 34 (1): 369–385. https://doi.org/10.1007/s10845-022-02038-4, https://doi.org/10.1007/s10845-022-02038-4.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. "ImageNet classification with deep convolutional neural networks." *Communications of the ACM* 60. https://doi.org/10.1145/3065386.

Le, Tuyen Quang, Pao Hsiung Chiu, and Chinchun Ooi. 2021. "U-Net-Based Surrogate Model for Evaluation of Microfluidic Channels." *International Journal of Computational Methods* https://doi.org/10.1142/S0219876221410188.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep learning." .

Lestandi, Lucas. 2018. "Low rank approximation techniques and reduced order modeling applied to some fluid dynamics problems." https://tel.archives-ouvertes.fr/tel-01947210/.

Lestandi, Lucas. 2021. "Numerical Study of Low Rank Approximation Methods for Mechanics Data and Its Analysis." *Journal of Scientific Computing* 87: 14. https://doi.org/10.1007/s10915-021-01421-2, http://link.springer.com/10.1007/s10915-021-01421-2.

Li, Xiaohan, and Nick Polydorides. 2022. "Time-efficient surrogate models of thermal modeling in laser powder bed fusion." *Additive Manufacturing* 59: 103122. https://doi.org/https://doi.org/10.1016/j.addma.2022.103122, https://www.sciencedirect.com/science/article/pii/S2214860422005115.

Manzoni, Andrea, Alfio Quarteroni, and Gianluigi Rozza. 2012. "Shape optimization for viscous flows by reduced basis methods and free-form deformation." *International Journal for Numerical Methods in Fluids* https://doi.org/10.1002/fld.2712.

Markl, Matthias, and K Carolin. 2016. "Multiscale Modeling of Powder Bed – Based Additive Manufacturing." *Annu. Rev. Mater. Res. 2016.* 46: 93:123. https://doi.org/10.1146/annurev-matsci-070115-032158.

Meng, Lingbin, Brandon McWilliams, William Jarosinski, Hye Yeong Park, Yeon Gil Jung, Jehyun Lee, and Jing Zhang. 2020. "Machine Learning in Additive Manufacturing: A Review." *Jom* 72: 2363–2377. https://doi.org/10.1007/s11837-020-04155-y.

Mercelis, Peter, and Jean pierre Kruth. 2006. "Residual stresses in selective laser sintering and selective laser melting." 5: 254–265. https://doi.org/10.1108/13552540610707013.

Mikula, Jakub, Robert Laskowski, Ling Dai, Wenjun Ding, Ming Wei, Kewu Bai, Kun Wang, et al. 2021. "Advanced "Digital Twin" platform for powder-bed fusion additive manufacturin." *Hitachi Metals Technical Review* 37: 10. https://www.hitachi-metals.co.jp/rad/pdf/2021/vol37_r01.pdf.

Nie, Zhenguo, Haoliang Jiang, and Levent Burak Kara. 2019. "Stress field prediction in cantilevered structures using convolutional neural networks." *Proceedings of the ASME Design Engineering Technical Conference* 1. https://doi.org/10.1115/1.4044097.

Nie, Zhenguo, Sangjin Jung, Levent Burak Kara, and Kate S. Whitefoot. 2020. "Optimization of Part Consolidation for Minimum Production Costs and Time Using Additive Manufacturing." *Journal of Mechanical Design* 142. https://doi.org/10.1115/1.4045106.

Popova, Evdokia, Theron M. Rodgers, Xinyi Gong, Ahmet Cecen, Jonathan D. Madison, and

Surya R. Kalidindi. 2017. "Process-Structure Linkages Using a Data Science Approach: Application to Simulated Additive Manufacturing Data." *Integrating Materials and Manufacturing Innovation* 6: 54–68. https://doi.org/10.1007/s40192-017-0088-1.

Promoppatum, Patcharapit, Recep Onler, and Shi Chune Yao. 2017. "Numerical and experimental investigations of micro and macro characteristics of direct metal laser sintered Ti-6Al-4V products." *Journal of Materials Processing Technology* 240. https://doi.org/10.1016/j.jmatprotec.2016.10.005.

Qi, Xinbo, Guofeng Chen, Yong Li, Xuan Cheng, and Changpeng Li. 2019. "Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives." 8.

Quaranta, Giacomo, Eberhard Haug, Jean Louis Duval, Elias Cueto, and Francisco Chinesta. 2019. "Parametric numerical solutions of additive manufacturing processes." *AIP Conference Proceedings* 2113. https://doi.org/10.1063/1.5112640.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. "U-net: Convolutional networks for biomedical image segmentation." Vol. 9351.

Rosen, D. W., and I. R. Grosse. 1992. "A feature based shape optimization technique for the configuration and parametric design of flat plates." *Engineering with Computers* 8. https://doi.org/10.1007/BF01200104.

Sen, Shuvam, Guillaume De Nayer, and Michael Breuer. 2017. "A fast and robust hybrid method for block-structured mesh deformation with emphasis on FSI-LES applications." *International Journal for Numerical Methods in Engineering* 111: 273–300. https://doi.org/10.1002/nme.5465.

Shelhamer, Evan, Jonathan Long, and Trevor Darrell. 2017. "Fully Convolutional Networks for Semantic Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39. https://doi.org/10.1109/TPAMI.2016.2572683.

Sieger, Daniel, Stefan Menzel, and Mario Botsch. 2014. "RBF morphing techniques for simulation-based design optimization." *Engineering with Computers* 30: 161–174. https://doi.org/10.1007/s00366-013-0330-1.

Staten, Matthew L, Steven J Owen, Suzanne M Shontz, G Andrew, and Todd S Coffey. 2011. "A Comparison of Mesh Morphing Methods for 3 D Shape Optimization." *Proceedings of the 20th International Meshing Roundtable* 293–311.

Tapia, G., A. H. Elwany, and H. Sang. 2016. "Prediction of porosity in metal-based additive manufacturing using spatial Gaussian process models." *Additive Manufacturing* 12.

https://doi.org/10.1016/j.addma.2016.05.009.

Tapia, Gustavo, Saad Khairallah, Manyalibo Matthews, Wayne E. King, and Alaa Elwany. 2018. "Gaussian process-based surrogate modeling framework for process planning in laser powder-bed fusion additive manufacturing of 316L stainless steel." *International Journal of Advanced Manufacturing Technology* 94: 3591–3603. https://doi.org/10.1007/s00170-017-1045-z.

Team, FreeCAD. 2020. "FreeCAD: Your own 3D parametric modeler." .

Tezzele, Marco, Nicola Demo, Andrea Mola, and Gianluigi Rozza. 2021. "PyGeM: Python Geometrical Morphing." *Software Impacts* 7. https://doi.org/10.1016/j.simpa.2020.100047.

Ueda, Yukio, Keiji Fukuda, Keiji Nakacho, and Setsuo Endo. 1975. "A New Measuring Method of Residual Stresses with the Aid of Finite Element Method and Reliability of Estimated Values." *Journal of the Society of Naval Architects of Japan* 1975. https://doi.org/10.2534/jjasnaoe1968.1975.138_499.

Ulu, Erva, Runze Huang, Levent Burak Kara, and Kate S. Whitefoot. 2019. "Concurrent Structure and Process Optimization for Minimum Cost Metal Additive Manufacturing." *Journal of Mechanical Design, Transactions of the ASME* 141. https://doi.org/10.1115/1.4042112.

Vannieuwenhoven, Nick, Raf Vandebril, and Karl Meerbergen. 2012. "A New Truncation Strategy for the Higher-Order Singular Value Decomposition." *SIAM Journal on Scientific Computing* 34: A1027–A1052. https://doi.org/10.1137/110836067.

Vohra, Manav, Paromita Nath, Sankaran Mahadevan, and Yung-Tsun Tina Lee. 2020. "Fast surrogate modeling using dimensionality reduction in model inputs and field output: Application to additive manufacturing." *Reliability Engineering & System Safety* 201: 106986. https://doi.org/https://doi.org/10.1016/j.ress.2020.106986, https://www.sciencedirect.com/science/article/pii/S0951832020304877.

Weiser, Alan, and Sergio E. Zarantonello. 1988. "A Note on Piecewise Linear and Multilinear Table Interpolation in Many Dimensions." *Mathematics of Computation* 50 (181): 189–196. Accessed 2023-06-09. http://www.jstor.org/stable/2007922.

Wu, Dazhong, Yupeng Wei, and Janis Terpenny. 2019. "Predictive modelling of surface roughness in fused deposition modelling using data fusion." *International Journal of Production Research* 57: 3992–4006. https://doi.org/10.1080/00207543.2018.1505058.

Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. 2018. "Convolutional neural networks: an overview and application in radiology." .

Yao, Wei, Zhigang Zeng, Cheng Lian, and Huiming Tang. 2018. "Pixel-wise regression using U-Net and its application on pansharpening." *Neurocomputing* 312. https://doi.org/10.1016/j.neucom.2018.05.103.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 2016. "3D U-net: Learning dense volumetric segmentation from sparse annotation." Vol. 9901 LNCS.