



Estimation of Surface Stresses on Voxel Meshes using Neuronal Nets on FEA Results in 2D Plane Stress Models

Franz B. Jungreitmayr¹ 

¹Johannes Kepler University Linz, franz.jungreitmayr@jku.at

Abstract. Voxel meshes are a natural choice for several types of analyses, particularly in the context of certain (gradient-based) optimization methods (including machine learning) or biomimicry. Such meshes can not directly be used to calculate meaningful surface stresses, though, e.g. for modeling fatigue. In this work, a process for calculating stresses based on neuronal nets (instead of smoothing and/or remeshing) is proposed, thereby potentially enabling the integration of stress calculation into optimization methods that rely on backpropagation (including generative design approaches). The method is demonstrated for 2D plane stress problems. Verification is attempted and the concrete behavior as well as limitations of the current implementation are discussed.

Keywords: Voxel, FEA, Stress Estimation, Machine Learning, Neuronal Net

DOI: <https://doi.org/10.14733/cadaps.2021.990-999>

1 INTRODUCTION

This work presents a neuronal-net-based method for estimating surface stresses based on a FEA results obtained on a voxelized mesh.

Voxel-style meshes are a natural choice for several types of analyses, particularly in the context of optimization methods: Many machine-learning methods operating on geometric data (specifically those incorporating convolutional networks) operate on such representations [1]. In addition, CT-type scans of natural structures like trabecular bone traditionally yield voxel-style representations of reality [5]. Voxelization shall, in this context, be defined as representing a (typically mostly smooth) geometry by all cubes of a fixed grid that are judged as “inside”. Continuous representations (i.e. each voxel is assigned a density value) are commonly used. In this work, though, density will be binary and active voxels are those, the centroids of which lie inside the original model (see figure 1).

The use of voxel meshes may also be promoted by recent trends in production: Additive manufacturing technologies may not even require an actually smooth geometry. So providing a suitable voxelized design may even be increasingly sufficient for production. Note, that for such an approach, effective compression strategies are known to facilitate an efficient representation of complex geometries [4].

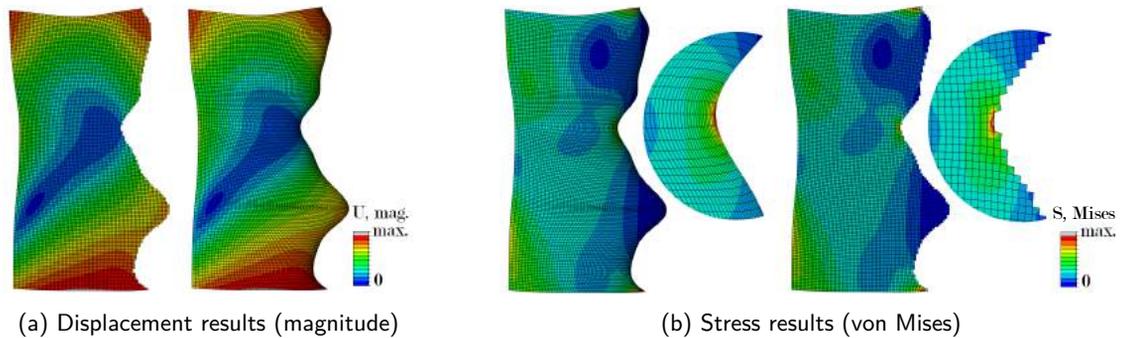


Figure 1: Results from one sample problem, on both smooth and voxelized models

For many application in mechanical engineering, fatigue calculation is a necessity. Although fatigue may occasionally be estimated based on voxel-wise strains [3], in particular for biological materials, the accurate calculation of local surface stresses is a necessity with many engineering materials due to typical modes of failure. Particularly in a high cycle fatigue context, stress results on a component's model's surface may be a necessary and even sufficient basis for an estimation of durability. The latter is particularly true for commonly used models describing fatigue as a stress-life relationship [2].

On a voxelized representation of geometry, calculating stresses is not immediately possible in a meaningful way: If stresses are directly derived from FEA results on a voxel model, these may heavily be influenced by discretization (edge length, element formulation and even coordinate origin) rather than the reflecting the implied shape. Surface stresses will not converge towards a smooth progression due to the jagged nature of the model, i.e. angles between neighboring elements' faces at the surface will be multiples of $\pi/2$. Still, displacement results converge with small element size and are consistent with results from smooth meshes (see figure 1a compared to figure 1b), as the voxel model actually approximates the stiffness of the original geometry.

One obvious strategy for calculating stresses more accurately is to derive an explicit estimation of the underlying geometry, meshing this geometry suitably and performing FEA on this new mesh, before calculating stresses from the resulting displacements. With this strategy, known implementations of the smoothing and re-meshing step (potentially incorporating complex feature recognition) will break the backtraceability of the overall process by switching from a voxel to a non-voxel mesh. Many optimization algorithms, though, benefit from the availability of exact gradients. For linear finite element calculations as well as neuronal nets, the calculation of the respective outputs' gradients with respect to the elements' stiffness (i.e., a voxels' modulus of elasticity) is known. Consequently, a stress calculation based on the voxel mesh might benefit such algorithms.

Combining such an FEA-based analysis with machine-learning-based design automation processes combining previously learned features in complex ways [10] may yield a potentially powerful optimization strategy: Figure 2 depicts the forward pass through a hypothetical system for numerically optimizing geometrical shapes. The process is based on structural simulation as well as features of heuristic nature, optimizations on such systems are frequently referred to as generative design [8]. Using a density field ρ as representation of geometry, convolutional neuronal nets (CNN) can be used to identify (desired or undesired) features. Such features may reflect targets like manufacturability, best design practices, cost or even aesthetics. Outputs of this analysis could be formulated as quality measures $Q_i(\rho)$. In parallel, finite element analysis can be performed on the very same representation of geometry, resulting in a displacement field. From this, stresses can be calculated (the focus of this work) which then serve as a basis for modeling fatigue, ultimately giving a safety factor $SF(\rho)$, which will act as another quality measure. Additional measures, either directly derived from the density field (e.g. total weight) or extracted from intermediate results (e.g. selected displacement

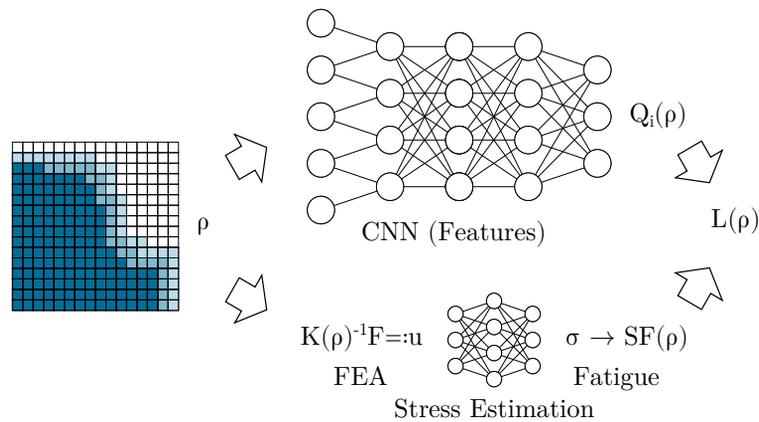


Figure 2: Forward Pass of used in a hypothetical generative Design process

values) might be included. Ultimately, from all quality measures a overall loss function $L(\rho)$, or – equivalently – a target function can be designed. Based on this evaluation process (the forward pass), which should allow backpropagation, an optimization can be set up to generate suitable shapes. Potentially suitable optimization methods are particularly under investigation in a computer vision context [7].

In specific applications, FEA has been replaced entirely by heuristic models altogether [6]. Such approaches typically have to be trained on specific class of problems. This work strives to avoid this limitations and be potentially applicable to any mechanical design problem.

In this work, a potentially novel method for estimating surface stresses is proposed. Within this first investigation, it is applied onto the two-dimensional plane stress problem.

As by the use of neuronal nets, a heuristic strategy is pursued, the method incorporates generating reference problems, extracting data for training, and finally training and applying the model. Those steps are presented in the following.

2 GENERATION OF REFERENCE PROBLEMS

To provide a sufficient data basis for the training of a neuronal net, a meta-model of problems was defined. This was chosen as follows: each training problem is constructed based on a mesh of square quadrilateral elements. The total width of the base mesh is 0.5 and its height is 1.0. While the other three edges of this mesh remain fixed, the right edge of each sample is distorted according to the following class of function. An example mesh can be seen in figure 1. Let y be the vertical coordinate in this representation and the center point of the original right edge (that is to be distorted) the origin of the coordinate system. Then the function (representing a Fourier series)

$$s(y) = c \sum_{i=1}^n a_i q^i \sin\left(\frac{iy}{2\pi} + \varphi_i + \varphi_0\right) \quad (c, n, q) = (0.2, 10, 0.5)$$

gives the boundary of the deformed patch for $y \in [-0.5, 0.5]$. The parameters $\varphi_i \in [0, 2\pi[$ and $a_i \in [0, 1[$ were chosen uniformly randomly for each sample. Subsequently, φ_0 was adjusted so that $s(-0.5) = s(0.5) = 0$. Accordingly, the continuous mesh was constructed by distorting a regular mesh of bi-linear plane stress quadrilateral elements.

Analogously to the randomly generated geometry, random boundary constraints were defined in polar coordinates with respect to the origin. Static terms enabling omnidirectional strain were introduced for all points on the undeformed three edges:

$$u_\varphi = x_r \sum_{i=1}^n b_i q^i \sin\left(\frac{i x_\varphi}{2\pi} + \chi_i\right) \quad u_r = c_0 + x_r \sum_{i=1}^n c_i q^i \sin\left(\frac{i x_\varphi}{2\pi} + \psi_i\right)$$

Figure 1a shows the displacements calculated for such a randomly generated sample problem. As stiffness of the models coincides (assuming sufficiently small edge lengths in the voxel model), the interior displacement values show very good consistence. As described above, surface stress results in the voxelized model do not locally give reasonable approximation of the stresses in the smooth model, as shown in figure 1b.

3 EXTRACTION OF TRAINING DATA

Subsequently, for every sample problem (defined by the above described random geometry and random boundary constraints), for every sufficiently interior voxel on the deformed edge, an input dataset as well as target stress data were extracted. Node displacements were used as primary input. Nodes that were not inside the geometry, i.e. that had no elements assigned to them, were assigned displacement $(0, 0)$, as an arbitrary nonzero value as indicator for outside nodes may have impeded the learning process. This obviously adds ambiguity to the meaning of nodes with displacements (close to) 0. So, in addition, a binary field (i.e. containing only values 0 or 1) was added, carrying the information about the nodes being present in the model. Accepting some limitations in the analogy, this additional field could be interpreted as a density field.

To obtain the associated targeted stress values from the continuous model, bilinear extrapolation of integration-point data was performed. The target location was chosen to match the respective voxel's centroidal position projected onto the edge of the geometry. From a total of 2000 pairs of meshes, 23 were excluded due to excessive distortion. From each of the remaining problems, 76 voxels on the non-straight edge were used as locations for stress evaluation. Each of those locations ultimately gave 4 data sets by rotation, resulting in a total of 601 008 data sets for training the model.

Let n_u and n_d denote the halved edge length of the patches (of displacements and densities respectively) used for input, hence being of Dimension $4n_u^2$. Using the index $\cdot^{(0)}$ to indicate original data from the reference problems, the extracted data is comprised of:

- Input data:
 - Node “densities” indicating a node's presence in the model, $\mathbf{d}^{(0)} \in \{0, 1\}^{4n_d^2}$
 - Displacements in the x-Direction, $\mathbf{u}_x^{(0)} \in \mathbb{R}^{4n_u^2}$
 - Displacements in the y-Direction, $\mathbf{u}_y^{(0)} \in \mathbb{R}^{4n_u^2}$
- Target vector $\sigma^{(0)} = (\sigma_{xx}^{(0)}, \sigma_{yy}^{(0)}, \sigma_{xy}^{(0)})$

As a neuronal net will be used to model the non-linear effects of the problem (and standard training methods are not very well suitable for arbitrarily scaled input data), the following three-step normalization process is used:

- Removing rigid body rotations: As overall rigid body rotations were deliberately excluded from the reference problem definition, these will be small in the training set. Hence, the learned heuristic will not learn that the problem is invariant to rotations. In real-world problems, in contrast, they may frequently occur and – in combination with the last step (displacement scaling) – give inconsistent results.
- Centering: The displacement values will be offset so that the arithmetic mean value will be zero

- Displacement scaling: The displacement values will be scaled so that the quadratic mean value will be zero. This – in contrast to the two previous steps – requires the consistent scaling of the stress values.

Let $k_u < 4n_u^2$ and $k_d < 4n_d^2$ denote the number of nodes present in the model in the respective (displacement or “density”) field and $\mathbf{u}_{:,j}^{(0)}$ denote the displacement and $\mathbf{x}_{:,j}^{(0)}$ the location vector of the j -th node of the patch. Then, assuming suitability of linearization for the rotation step, the normalization algorithm described above can be written as:

$$\begin{aligned}\bar{\alpha} &= \frac{1}{k_u} \sum_j \frac{\|\mathbf{x}_{:,j}^{(0)} \times \mathbf{u}_{:,j}^{(0)}\|}{\|\mathbf{x}_{:,j}^{(0)}\|} \\ \mathbf{u}_x^{(1)} &= \mathbf{u}_x^{(0)} + \bar{\alpha} \mathbf{x}_y^{(0)} \\ \mathbf{u}_y^{(1)} &= \mathbf{u}_y^{(0)} - \bar{\alpha} \mathbf{x}_x^{(0)} \\ \mathbf{u}_i^{(2)} &= \mathbf{u}_i^{(1)} - \frac{\mathbf{u}_i^{(1)} \cdot \mathbf{1}}{n}, i \in \{x, y\} \\ \mathbf{u}_i^{\text{input}} &= \mathbf{u}_i^{(2)} \frac{\sqrt{k_u}}{\sqrt{\mathbf{u}_x^{(2)} \cdot \mathbf{u}_x^{(2)} + \mathbf{u}_y^{(2)} \cdot \mathbf{u}_y^{(2)}}}\end{aligned}$$

Stress scaling was adjusted so that the resulting model would be independent of the elastic modulus and the voxel size. The use of \hat{y} , which denotes half the edge length of the displacement input range (instead of the voxel edge length) and constant factor of $\sqrt{3}$ results from the intention to have $\sigma_{xx}^{\text{target}} = 1$ for constant uniaxial stress. Effects of Poisson’s ratio were neglected at this point, it was set at 0.3 in all calculations.

$$\sigma^{\text{target}} = \frac{\sqrt{k_u} \hat{y}}{E \sqrt{3} \underbrace{\sqrt{\mathbf{u}_x^{(2)} \cdot \mathbf{u}_x^{(2)} + \mathbf{u}_y^{(2)} \cdot \mathbf{u}_y^{(2)}}}_{=: 1/\sigma_{\text{ref}}}} \sigma^{(0)}$$

This way, the model can be trained to return a dimensionless stress factor independent on voxel resolution and material.

4 TRAINING OF THE MODEL

Training of neuronal nets was performed using a stochastic gradient descent algorithm incorporating a momentum term. The loss function was defined as the mean square error with respect to the plane stresses as a vector. The net used as an initial basis for the investigations is described in the following table:

Layer	Type	Dimension	Activation
1	fully connected	$(4n_d^2 + 8n_u^2) \times 256$	tanh
2	fully connected	256×128	tanh
3	fully connected	128×64	tanh
4	fully connected	64×3	1

For this net, several cases of input vectors were specifically investigated: $(n_d, n_u) \in \{(8, 8), (8, 4), (4, 4)\}$. The results in terms of von-Mises equivalent stress on sample problems not used for training are visualized in figure 3.

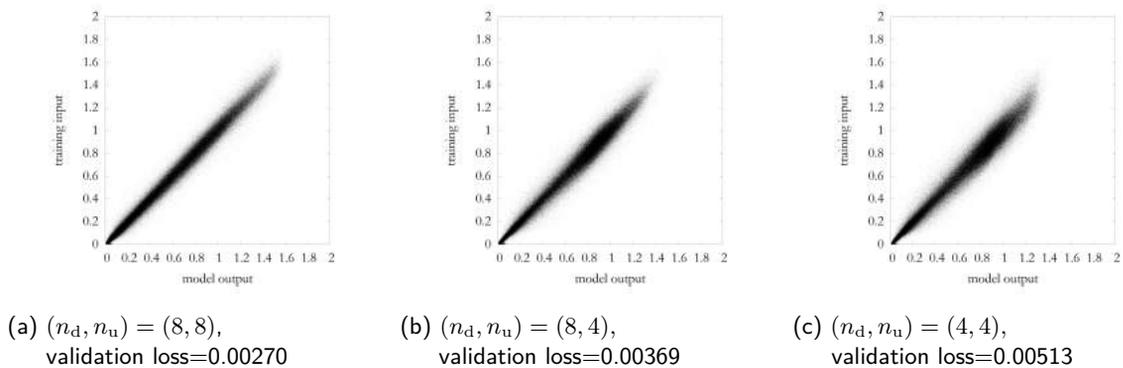


Figure 3: Fit of model output to Reference Solution

Number show that a reasonably accurate prediction can already be made at $n_d = n_u = 4$ within the meta-model of training problems. Nevertheless, increasing the scope of “perception” of the model somewhat improves the prediction quality. Using four times the input dimensions results in roughly half the loss. In figure 3b, one can – in addition to a greater deviation – observe a slight convex bend with high relative stresses. That is, high stress concentrations tend to get underestimated by the model.

In addition, variations of the neuronal nets were generated by removing the first hidden layer of dimension 256 or the two first layers leaving only one hidden layer to successively decrease complexity. Verification loss values (with respect to sample problems not used for training) are given below, together with the total number of degrees of freedom in the model.

Hidden Layer Sizes	DOFs at $(n_d, n_u) =$			Losses at $(n_d, n_u) =$		
	(8,8)	(8,4)	(4,4)	(8,8)	(8,4)	(4,4)
256,128,64	238 211	139 907	90 755	0.00270	0.00369	0.00513
128,64	106 883	57 731	33 155	0.00283	0.00364	0.00516
64	49 411	24 835	12 547	0.00292	0.00376	0.00523
lin. reg.	2 307	1 155	579	0.0129	0.0129	0.0129

These results show very little dependency on model complexity, that is, even the simplest models gave reasonably accurate results. This indicates that the problem of estimating stresses on sufficiently smooth underlying geometries is much simpler than initially expected. For reference, the results of a linear regression model on the same data-set is given in the last line.

A basic investigation of the weights in the first layer of the neuronal net is shown in 4. The norm of all gains with respect to each input channel is visualized. It is not surprising that particularly in the bigger model (4a) the stresses depend on relatively wide geometry input, while most relevant displacement (i.e. strain) information is very local. In contrast, the density in the very center is completely redundant (stresses are only queried for existing elements).

5 VERIFICATION

Outside-model validation was attempted to investigate the suitability of the proposed process for real-world engineering problems. A beam with a circular notch under torsional loading was chosen, as it exhibits particularly significant differences to the problems within the training data set: Instead of the harmonic nature of the

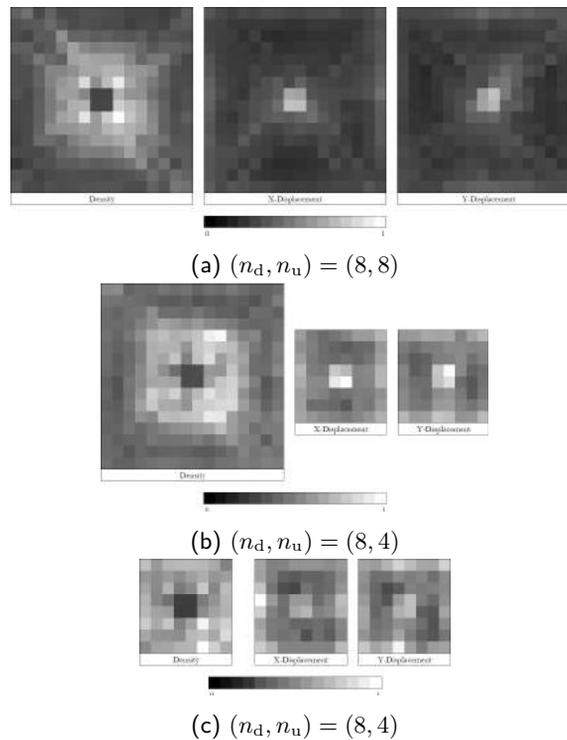


Figure 4: Weights (2-Norm) in the first Layers

training data, it consists of segments of constant curvature, connected by sharp corners. Still, in principle, it should be possible to estimate the curvature – and thereby the stress values – within those segments assuming the choice of a suitable resolution.

A sketch of the geometry and a plot of concentration factors from [9] is given in figure 5. The choice of parameters for the validation problem $(r/b, B/b) = (0.1, 1.15)$ is highlighted in the figure. The voxelized model investigated in the following was constructed so that at the critical location (i.e. the center of the notch), there was always a gap exactly one voxel wide. This choice was made in order to provoke possible artifacts of the models, thereby serving as a plausible worst-case problem.

6 compares the stress estimates of the models. The lower section of figure 6 shows the dependency of the models' (von-Mises-equivalent stress) estimates on the meshing size. The meshing size is given by the multiple of voxel edges amounting to the notches radius.

For context, the upper section of figure 6 displays the scope of input for selected models (resolution indicated by the tick marks connecting the grids to the plot). Black cells are parts of the area of $n_i = 4$ gray cells of the area of $n_i = 8$. White cells are not part of the model. Note that this visualization is element-based, while the actual input to the model is node-based.

The curves for the $n_u = 4$ seem to show a plateau for 1.5 to 2.5 elements per radius in good accordance with the reference solution. This should not be overstated, as in this range the notch is represented by a constant voxel mesh and differences in model output only result from slight differences in the displacement field (and, thereby, reference stress).

The results show a strong variation of the model output with resolution. The tendency to underestimate at low resolutions is to be expected: the smoothness of training problems will lead to an underestimation of

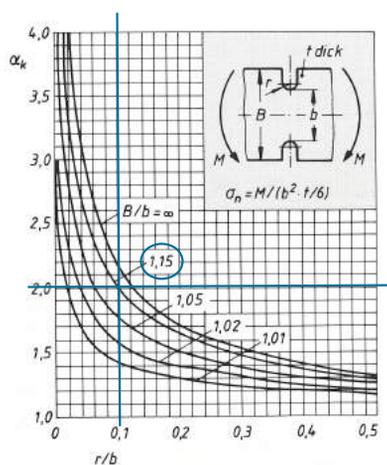


Figure 5: Stress concentration factors from [9]

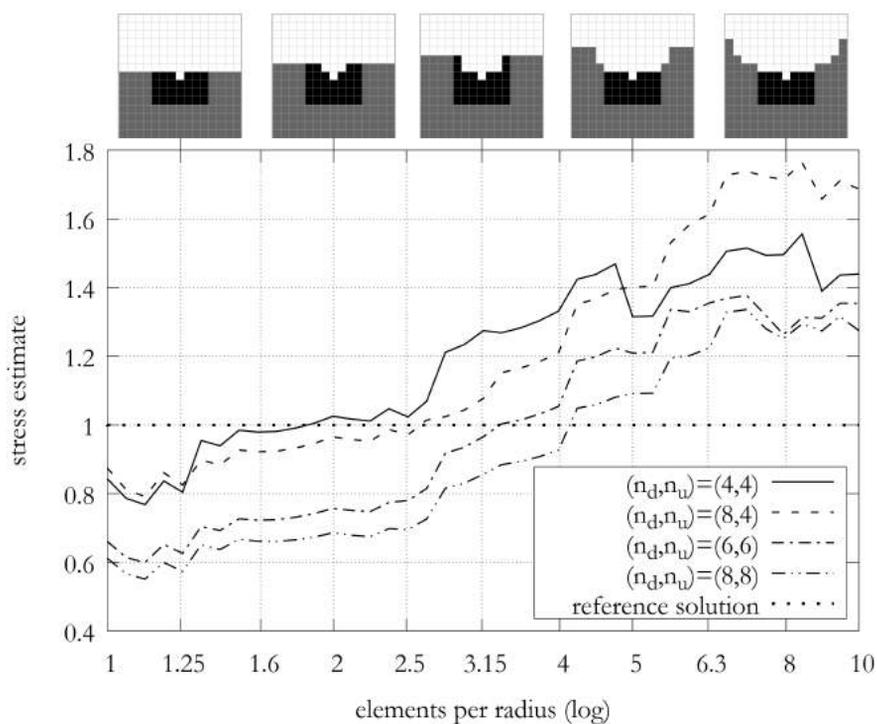


Figure 6: Results for the notched Beam

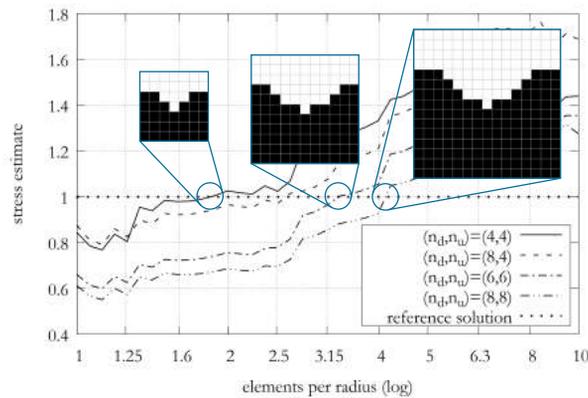


Figure 7: Resolutions resulting in accurate Stress Estimates

curvature as the surrounding segments of curvature 0 will dominate.

The overestimating behavior at high resolutions is less obvious: For high resolutions, the adversely chosen problem gives the same geometric input as very low resolutions (a straight boundary with one single voxel missing). Normalized strain will vary at a different rate (and curvature) within the geometry, determined by overall bending stiffness (i.e. relative width) as well as local stress concentration (notch geometry). The model is seemingly unable to distinguish those effects, though. It hence returns a very similar factor (approximately 1.7 for $(n_d, n_u) = (8, 8)$) for both problems. The reference stress σ_{ref} could be informally described as the “average stress” in the geometry under consideration. For low resolutions, this reference stress gives a value close to the nominal stress, which is to be scaled by a concentration factor in classical mechanical engineering. For high resolutions, in contrast, the reference stress in the voxel model is already very close to the real stress. Nevertheless, it is again scaled by the similar concentration factor.

Figure 7 selectively visualizes the input ranges for “ideal” resolutions in terms of accuracy of the stress estimate. To the author, it is intuitively striking that for all those cases, the input range resembles a somewhat “harmonic” contour. A closer investigation reveals, though, that the theoretical curvatures of a single harmonic fitted to the input contour does not quite give the same maximum curvature as the underlying circular notch.

For context, the upper part of figure 6 shows the scope of input for selected models (resolution indicated by the tick marks connecting the grids to the plot). Black cells are parts of the area of $n_i = 4$ gray cells of the area of $n_i = 8$. White cells are not part of the model.

The curves for the $n_u = 4$ seem to show a plateau for 1.5 to 2.5 elements per radius in good accordance with the reference solution. This should not be overstated, as in this range the notch is represented by a constant voxel mesh and differences in model output only result from slight differences in the displacement field (and, thereby, reference stress).

In summary the model trained in this work exhibits a strong bias towards certain properties of the input geometry (i.e. a narrow-band superposition of harmonic functions), which can only be predetermined by the definition of the training problems. This is consistent with the above minimal decrease in loss due to reduction of model complexity, which may also point towards an inherent bias.

6 CONCLUSION

The model trained in this demonstration works well within the used class of problems (of smooth shape). For problems of lower degree of smoothness, its usability decreased sharply. The limitations may be determined

by the choice of training problems, even though some error is inherent to the problem.

For bio-mimicry tasks, particularly with trabecular structures, this might not be a significant limitation, as some natural structures may tend to be smooth and typical curvatures may be known.

In the context of reverse-engineering of artificial structures or topology optimization/generative design, these limitations will be serious. Discussed issues may be mitigated by the following advancements:

Training problems shall be defined in a more generic, “less smooth” way. Training problems can be constructed by randomly combining predefined geometrical features, i.e. performing Boolean operations on randomly sized and placed primitives. This also transfers well to 3D problems.

The learning process shall be extended to continuous (non-binary) density fields. Such a field trivially carries more information on the underlying shape and should thereby support the estimation process. In addition, optimization processes may require the a usable model to operate on smooth (in terms of density gradient) representations of geometry. In this context, different approaches to calculating stress-like measures might arise, like using stresses and density gradients directly.

Successively, the process depicted in figure 2 shall be demonstrated, providing an expandable generalization of topology optimization, and ultimately transferred to 3D problems.

ORCID

Franz B. Jungreitmayr <http://orcid.org/0000-0003-3083-1908>

REFERENCES

- [1] Brock, A.; Lim, T.; Ritchie, J.M.; Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. CoRR, abs/1608.04236, 2016. <http://arxiv.org/abs/1608.04236>.
- [2] Dowling, N.E.: Mean stress effects in stress-life and strain-life fatigue. In SAE Technical Paper. SAE International, 2004. <http://doi.org/10.4271/2004-01-2227>.
- [3] Harrison, N.M.; McDonnell, P.; Mullins, L.; Wilson, N.; O’Mahoney, D.; McHugh, P.E.: Failure modelling of trabecular bone using a non-linear combined damage and fracture voxel finite element approach. Biomechanics and Modeling in Mechanobiology, 12(2), 225–241, 2013. ISSN 1617-7940. <http://doi.org/10.1007/s10237-012-0394-7>.
- [4] Kämpe, V.; Sintorn, E.; Assarsson, U.: High resolution sparse voxel dags. ACM Trans. Graph., 32(4), 2013. ISSN 0730-0301. <http://doi.org/10.1145/2461912.2462024>.
- [5] Knowles, N.K.; Ip, K.; Ferreira, L.M.: The effect of material heterogeneity, element type, and down-sampling on trabecular stiffness in micro finite element models. Annals of Biomedical Engineering, 47(2), 615–623, 2019. ISSN 1573-9686. <http://doi.org/10.1007/s10439-018-02152-6>.
- [6] Liang, L.; Liu, M.; Martin, C.; Sun, W.: A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. Journal of the Royal Society, Interface, 2018. <https://doi.org/10.1098/rsif.2017.0844>.
- [7] Mahendran, A.; Vedaldi, A.: Understanding deep image representations by inverting them. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5188–5196, 2015.
- [8] Oh, S.; Jung, Y.; Kim, S.; Lee, I.; Kang, N.: Deep generative design: Integration of topology optimization and generative models. Journal of Mechanical Design, 141(11), 2019.
- [9] Wittel, H.; Muhs, D.; Jannasch, D.; Vossiek, J.: Roloff/Matek Maschinenelemente. SPRINGER Fachmedien Wiesbaden, 2015. <http://doi.org/10.1007/978-3-658-09082-1>.
- [10] Wu, J.; Zhang, C.; Xue, T.; Freeman, W.T.; Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. CoRR, abs/1610.07584, 2016. <http://arxiv.org/abs/1610.07584>.