

## SOLVING TRANSIENT INVERSE HEAT TRANSFER PROBLEMS IN COMPLEX GEOMETRIES USING PHYSICS-GUIDED NEURAL NETWORKS (PGNN)

D. Emonts<sup>1\*</sup>, J. Yang<sup>1</sup>, R. H. Schmitt<sup>1</sup>

<sup>1</sup> RWTH Aachen University, Laboratory for Machine Tools and Production Engineering WZL, Aachen, Germany

\*Corresponding author; e-mail: D.Emonts@wzl.rwth-aachen.de

### Abstract

Temporally and spatially unstable thermal conditions lead to transient or inhomogeneous thermo-elastic behavior of workpieces during manufacturing or geometric inspection. Temperature monitoring by means of sensors consign transient temperature fields, but do not yield information about the heat flow acting as thermal boundary condition, which is a relevant input parameter for nearly any thermal simulation. Addressing the need for efficient methods, the authors propose an approach to solve inverse heat transfer problems in complex geometries. In the presented study, locally acting heat loads are experimentally investigated based on virtual demonstrators running in FEM. The conducted method shows high potential for transient heat flow modelling in terms of accuracy and computational efficiency.

### Keywords:

Physics-guided neural networks; transient heat flow; inverse modelling; temperature measurement

## 1 INTRODUCTION

Temperature is a major impact on production engineering processes, highly influencing process design with respect to lead times, tool wear and final workpiece quality. Thus, the investigation of thermal behavior of machine tools, workpieces, manufacturing environment and their mutual interdependencies have been subject to research for years. According to past research, the thermal state of the machine tool and workpiece together account for up to 75% percent of part deviation from ideal geometry [Ess 2012; Li 2017]. In this context, most studies and modelling approaches solely focus on errors of the machine tool itself, whereby effects originating from the workpiece are often neglected. Further strengthened by increasing precision requirements, the explicit error modelling of the workpiece and its overall error contribution with special respect to geometric inspection is shifting more into focus of relevant research. [Yang 2017; Ross-Pinnock 2018; Ohlenforst 2019]

While most industrial established compensation approaches make use of tempering techniques, meaning not to counteract against, but to mitigate the effects of temperature at the expense of costly acclimatization or scheduled downtimes, the economic inefficiency of these methods is prevailing. Seeking for alternative solutions to account for thermo-elastic workpiece effects not just for high precision measurements, but also manufacturing or assembly the utilization of model-based compensation techniques is auspicious. A review of industrial practice reveals though, that thermal influences of the workpiece are approximated very conservatively in many instances – applying, if any, Newton's Law of cooling (1) or linear thermal expansion models (2).

$$\frac{dQ}{dt} = -h \cdot A \cdot (T_{obj}(t) - T_{env}) \quad (1)$$

$$\frac{\Delta L}{L} = \alpha \cdot \Delta T \quad (2)$$

The underlying assumptions of consistent single materials, constant heat transfer coefficients over the whole geometry and known homogenous temperature distributions do not reflect real-world conditions though. FEM, as commonly used numerical approach to perform multi-physics simulations, is a suitable tool to predict thermally induced workpiece deformations, offering the opportunity to consider multiple varying temperature loads, such as heat flux from the manufacturing process, heat conduction between clamping surfaces or variable ambient temperatures. Although it is possible to simulate and numerically solve virtually any process, the efficient application often suffers from defining correct initial (IC) and boundary conditions (BC), since they have to be known and set up in the simulation model in advance. Not to mention the high computational effort, making the method infeasible for many cases, especially large workpieces. Workpieces in production environment are considered as open, transient thermodynamic systems in continuous interaction with its variable environment. Hence, it is necessary to keep track of the influencing variables, such as transient ambient temperatures, surface temperature distributions or heat fluxes and feed them back into simulation models. Therefore, the importance of cyber physical production systems (CPPS), which is used as the representative term for the fusion of virtual systems (e.g. digital twin in FEM) with physical systems (sensor data) was already pointed out as key technology for thermal error correction [Ohlenforst 2019].

While monitoring ambient and surface temperatures using temperature sensors is a straightforward task, capturing heat fluxes is more challenging, which requires more advanced modelling techniques. At this point, the need for methods to solve inverse heat transfer problems (IHTP) arises. In this context, inverse posed problems address heat fluxes, which are the cause of a temperature distribution and can be quantified by temperature measurements. Most commonly information about heat fluxes are heuristically estimated, which represent an uncertainty for common forward transient simulation methods, such as FEM – thus, the determination of heat fluxes is a valuable asset, acting as additional boundary condition. In fact, inverse calculations belong to optimization problems. Providing additional observation data besides a partial differential equation system (PDE), it aims to derive the unknown parameter  $\theta^*$ , representing heat flux in the course of this study:

$$\theta^* = \operatorname{argmin}(\hat{T} - T_{ob}) \quad (3)$$

$\hat{T}$  denotes the temperature derived from calculation and  $T_{ob}$  is the observed temperature. In order to solve inverse problems, the Conjugate Gradient Method (CGM) is a commonly used approach to retrieve unknown parameters through minimizing the discrepancy between observations and the solution of PDEs. CGM is a computationally intensive procedure, seeking for an iterative solution for the direct problem using the current assumption of unknown parameters. Additionally, it involves the effort to calculate the gradient in conjugated direction and to formulate and solve the adjoint problem. Therefore, CGM has limited scope to carry out real-time prediction and diagnosis, especially as it comes to complex or even large workpieces. [Huang 2009, Burghold 2017]

Besides above discussed methods, the application of neural networks in performing multi-physics simulations is an emerging branch of research. The traditional data-driven way faces intractable problems though. To obtain high accuracy and generalizability, the training requires large amounts of labeled data. The process of obtaining data, by either virtual simulations or field experiments, is time and cost consuming. Furthermore, recent works realizing such methods are likely to break physical laws, even if they have achieved low overall errors [Karpatne 2018]. Neural networks usually conduct regression on a given data set, but learn nothing about the implicit physical rules. This affects the liability for implementation.

To combine the benefits of analytical methods with the advantages of machine learning techniques, Physics-Guided Neural Networks (PGNN) were introduced recently [Alexandre 2018], [Holl 2018], [Zhu 2019]. The basic idea is to apply physical laws into a neural network, whereby the most common approach is to add certain physical laws into the loss function in order to force the neural network to comply with them [Karpatne 2018], [Kim 2019]. A given thermal transfer problem has the following form:

$$\mathcal{D}(\mathbf{x}, t, T) = 0 \quad \mathbf{x} \in \Omega \quad (4)$$

$$\mathcal{B}(\mathbf{x}, t, T) = 0 \quad \mathbf{x} \in \partial\Omega \quad (5)$$

Where  $\mathcal{D}$  denotes all differential operator in the thermal Poisson equation (6) and  $\mathcal{B}$  describes the BCs.

$$\rho c \frac{\partial T}{\partial t} = \lambda \left( \frac{\partial^2 T}{\partial \mathbf{x}^2} \right) \quad \mathbf{x} \in \Omega \quad (6)$$

In (6)  $\lambda$  denotes thermal conductivity coefficient and  $\rho, c$  are the density and specific heat capacity of the workpiece

respectively. The quantity  $\mathbf{x}$  represents 3D spatial coordinates,  $t$  is time and  $T$  is the solution of the problem, pursuant to the temperature depending on  $\mathbf{x}$  and  $t$ . The domain is specified by the geometry  $\Omega$  and  $\partial\Omega$  boundary, namely surface of the workpiece.

$$\mathcal{D}(\mathbf{x}, t, \mathcal{N}(\theta)) = 0 \quad \mathbf{x} \in \Omega \quad (7)$$

$$\mathcal{B}(\mathbf{x}, t, \mathcal{N}(\theta)) = 0 \quad \mathbf{x} \in \partial\Omega \quad (8)$$

To solve above equations by means of PGNNs, a neural network is designed with the network output  $\mathcal{N}(\theta)$  being the surrogated solution, which replaces the real solution in the equation system.  $\theta$  are weights and biases of the neural network. If equation (4) and (5) hold,  $\mathcal{N}(\theta)$  approximates the original solution  $T$  in the given spatiotemporal domain. Therefore, the loss function  $\mathcal{L}$  of the neural network can be formulated as the following:

$$\mathcal{L} = \mathcal{D} + \mathcal{B} \quad (9)$$

The learning process aims to minimize  $\mathcal{L}$  to 0. For a PGNN, solving IHTP is no more difficult than its direct counterpart. By specifying the unknown BC as parameter  $\theta^*$  and enclosing additional data in terms of observed temperature distributions  $T_{ob}$ , the direct problem can be transferred into an inverse one:

$$\mathcal{L}_{inv} = \mathcal{D}(\mathcal{N}(\theta)) + \mathcal{B}(\theta^*, \mathcal{N}(\theta)) + ||\mathcal{N}(\theta) - T_{ob}|| \quad (10)$$

When loss is minimized, it means the surrogated solution has satisfied the PDE, the BCs and the observation. Afterwards, the unknown parameter  $\theta^*$  can be obtained using machine learning. Most common neural network frameworks, such as *TensorFlow*, can optimize these parameters the same way as all other parameters ( $\theta$ ) within the network [Martín 2016]. Compared to the large amount of  $\theta$ , the computational resource for several  $\theta^*$  is almost ignorable. Therefore, PGNNs can be theoretically considered to be more efficient than traditional numerical methods to solve IHTP.

## 2 EXPERIMENTAL SETUP AND METHOD

At first, this chapter precludes an overview to the workflow of conducted analysis with a subsequent presentation of the objects under investigation (2.1). In section (2.2) the design of experiments with respective parameter definition is described. Following, the software infrastructure is introduced in 2.3. The main challenge during training of the neural network, the hyperparameter tuning is discussed in 2.4.

### 2.1 Workflow and Objects of Investigation

Presenting the basic procedure, Fig.1 introduces an overview of the general workflow of experiments, breaking down demonstrators, individual processing steps and necessary data flows.

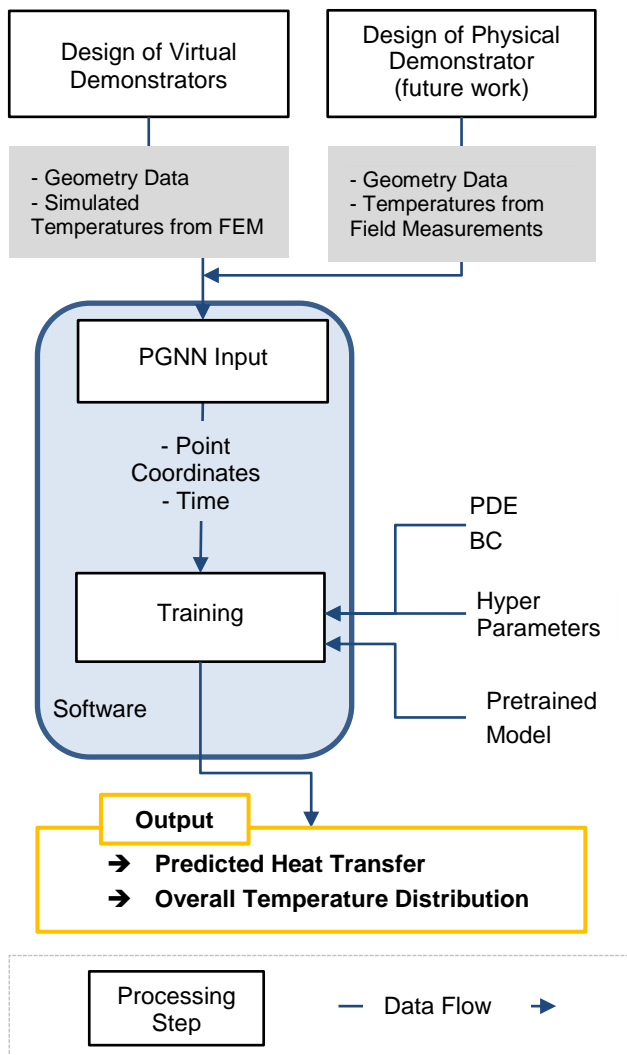


Fig. 1: Workflow of experiments with corresponding inputs and outputs

Firstly, to implement the approach of using PGNNs to build an inverse heat transfer model, five different basic shapes were investigated. Externally designed in CAD, they cover plate, cylinder, solid half cylinder (SHC), hollow half cylinder (HHC) and sphere (Fig. 2). The created data set serves as virtual sensor data, which was utilized to tune the model and validate the performance for defined geometries. To feed the neural network with temperature gradients, the virtual sensors were chosen within pre-defined areas, as depicted in red. Then, transient thermal FEM models were created, using these geometries (virtual demonstrators) for predefined physical conditions (Table. 1).

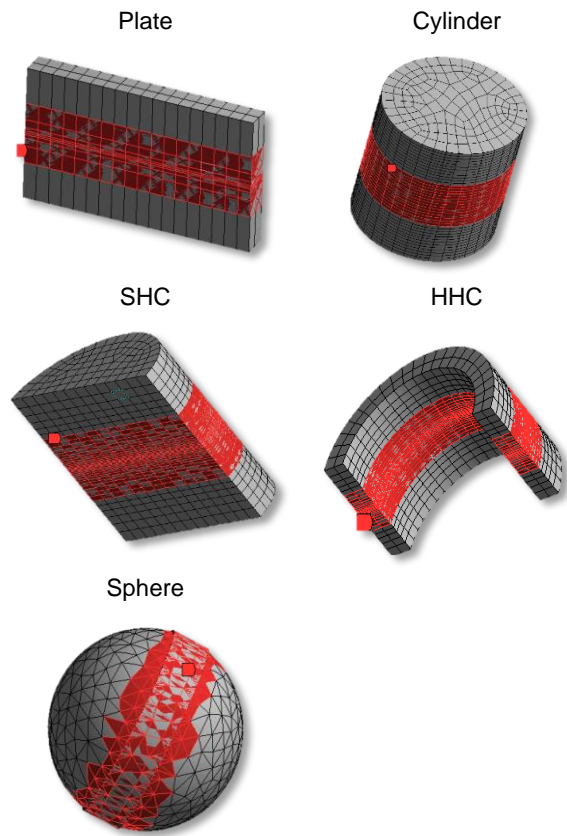


Fig. 2: Virtual demonstrators and sensor applying area (red).

## 2.2 Design of Experiments (DOE)

The presented series of experiments aims to verify the capacity of PGNN at an early stage and serve as preparation for more complex tasks. For all experiments identical BCs are applied, namely:

Dirichlet BC (Initial temperature)

$$T = F(\mathbf{x}, t) \quad \mathbf{x} \in \Omega \quad (11)$$

Neumann BC (heat flux):

$$\frac{\partial T}{\partial \mathbf{n}} = -\frac{1}{\lambda} \mathbf{q}(\mathbf{x}, t) \quad \mathbf{x} \in \partial\Omega \quad (12)$$

and Robin BC (natural convection):

$$\frac{\partial T}{\partial \mathbf{n}} = \frac{1}{\lambda} h(T_{am} - T) \quad \mathbf{x} \in \partial\Omega \quad (13)$$

Table. 1: DOE overview

Property	Magnitude	Location
Heat Flux	5000 W/m <sup>2</sup>	400 mm < z < 600 mm
Convection HTC	5W/(m <sup>2</sup> C)	All Surfaces (Heated area excluded)
Initial Temperature	25°C	Whole Body
Ambient Temperature	25°C	
Material	Cast Iron	--
Timeframe	600 s	--

$F(*)$  is the function describing the temperature distribution at spatiotemporal points  $(x, t)$ ,  $\lambda$  represents the thermal conductivity coefficient,  $\mathbf{n}$  denotes the boundary normals,  $q$  is the function of heat flux,  $h$  the heat transfer coefficient and  $T_{am}$  the ambient temperature around the workpiece. The testing conditions and corresponding magnitudes for the virtual demonstrators are chosen with respect to the estimated conditions based on technical specifications of a physical demonstrator, which will be investigated in future. In the inverse problem, the unknown heat flux, which is caused e.g. by induced thermal energy from a machining process, is aimed to be estimated by the PGNN. To constraint the degree of freedom for the system, surface temperature data is provided by the demonstrators by virtual sensors coming from FEM.

### 2.3 Software infrastructure

The developed software tool (Fig. 3) uses an open source python package named DeepXDE [Lu 2020].

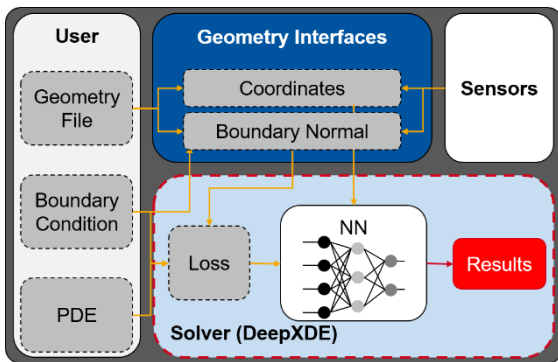


Fig. 3: Software infrastructure.

DeepXDE provides a deep learning library for solving differential equations, hence providing the core framework to conduct the workflow of using PGNNs. While DeepXDE entails the modules for building basic geometries, assigning physical conditions (PDE, BC, IC) and training of the neural network, additional interfaces must be developed to overcome limitations regarding the input of complex geometries and sensor integration, which was necessary for the conducted study.

By means of the package integrated tool named *Constructive Solid Geometry* (CSG), with which the geometry is limited to basic components, like cuboid, cylinder, etc. and their combinations, the capabilities of defining complex geometries from engineering data are very limited though. Since the long term goal of this early stage study is to implement the procedure in field tests for more complex geometries, requiring the software to receive and interpret standard file formats, like VTK or STL, the authors developed an interface. The interface enables the extraction of point coordinates as well as boundary normals from external geometry files and aggregate them with temperature data in a PGNN interpretable pattern.

There is a major difference in geometry handling between FEM and PGNN. While FEM calculates through all nodes in the mesh at all timesteps and, thus, involves tedious work to balance the calculation efficiency and accuracy through careful design of mesh, PGNN only needs a part of the pointset at every timestep. That is because DeepXDE considers timestamps as nothing else than an additional dimension accompanying the 3D coordinates of spatial points and samples training data in the whole 4D domain instead of running through all points step by step. Such a scheme enables to use geometries with fine granularity accompanied by only little increase in time consumption.

Besides and without the constraints of common meshing, a flexible sensor integration is achieved. Compared to FEM, where adding extra points in meshing requires rebuilding the mesh and consequentially costs extra time, it turned out that the developed framework allows to directly integrate any number of sensors into the “meshless” solver, since there is no geometrical interaction among spatial points. For further understanding about the solver, the authors refer to [Lu 2020]. Additionally, adding extra sensor data even after the training has started makes the approach flexible. In order to integrate sensors for PGNN calculation, the coordinates and boundary normals of the surface are mandatory. Corresponding boundary normals are directly passed by the STL files. The described infrastructure results in almost no extra calculation for model preparation and thus entails significant potential for increase of efficiency in comparison to alternative simulation tools.

### 2.4 Hyperparameter Tuning

After setting up the initial model, specifying the geometry, ICs and BCs, it is necessary to specify the hyperparameters for training. Hyperparameters are parameters that can usually not be trained through the neural network [Goodfellow 2016] and thus must be assigned through empirical or heuristic knowledge. The selection of these hyperparameters influences the output of neural networks in a fundamental way. In this study, all investigated use-cases must comply with the same PDE. Therefore, most hyperparameters can remain, once they were determined for one case. However, the following four factors: *Neural Network Architecture, Loss Weights, Point Sampling Scheme and Initial value of unknown BC*, which are significantly influenced by the geometrical characteristics, need to be specified case by case. They are described in the following.

#### Neural Network Architecture

As a starting point, the presented study adopted a basic feed forward neural network (FNN) as neural network architecture. The layers in FNN are fully connected, which means any neuron from one layer is connected with all neurons in the next layer and vice versa. Besides FNN, a Resblock, which adds extra connections for every two layers, was adopted (Fig. 4). The scale of network in terms of layers and neurons depends on the complexity of the problem. Besides input and output layers, a Resblock (consisting of 2 layers) with 50 neurons in every layer is adopted.

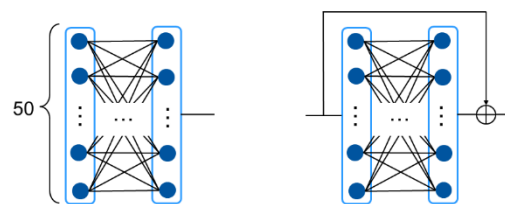


Fig. 4 : FNN (left) and Residual Neural Network (right).

#### Loss Weights

For implementation, the magnitude of BCs may vary a lot. In the course of this study, the heat flux from the outer source is much larger than that from convection. If they are summed up naively and implemented into the loss function, the optimizer will be likely to ignore the gradient from convection and therefore may only find local optima. Hence, Loss weights are elementary to balance all terms in the loss function, maintaining their weighted magnitude within a similar range.



$$\mathcal{L}_{weight} = W \cdot \mathcal{L} \quad (14)$$

In this study, the loss weight is set manually to make sure all BCs in loss lie within a span of  $10^0$ - $10^2$  and the weight of PDE term is always 1. Besides, Loss weights also serve to balance the training difficulty in different BCs [Guo 2018]. In the presented work, experiments indicated that Neumann (12) and Robin (13) BCs are harder to learn than Dirichlet BCs (11). Fig. 5 shows a training process only balancing the magnitude but ignoring the training difficulty of different BCs. The y-axis represents the loss/initial loss ratio at epoch 0 in logarithm scale. As illustrated, the Dirichlet condition, namely the initial temperature and observation decreases to a lower level than other terms in the whole training process.

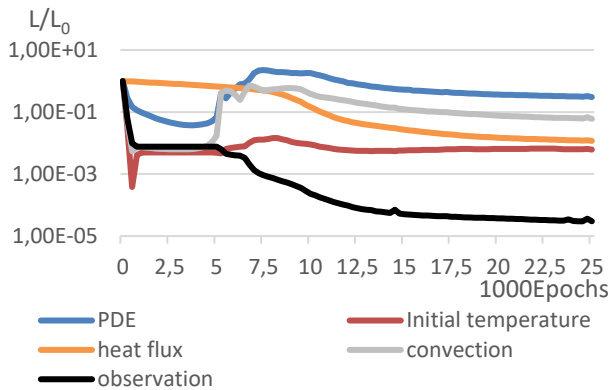


Fig. 5 : Loss of PDE and different BCs.

In order to overcome the training difficulty in specific BCs, larger loss weights can be assigned to force the optimizer to conduct gradient descent faster on difficult BCs and accelerate overall training efficiency.

#### Point sampling scheme

Meshing is a crucial factor using the example of FEM, especially for complex geometries with subtle structures. In contrast, the conducted experiments with PGNNs revealed, that the neural network is less sensitive to the shape and the geometrical interaction among points of the model. Randomly distributed points within the complete spatiotemporal domain are a good foundation of sampling, whereby the gradient of output will influence the neural network a lot. Within the study, it could be observed that it is relatively hard for the neural network to learn high gradients. This is indicated, as it tends to give out blurred results, when temperature changes fast. Residual adaptive refinement (RAR), proposed by the original developers of the utilized software library *DeepXDE*, can improve results in this case. It iteratively samples new points, which have the largest residual against the PDE to increase accuracy. Since this method involves additional iteration, it will increase training time though. Therefore, a more straightforward way of sampling, which is manually adding more points at areas of interest (points with expected high gradients) for training, was applied within the presented work. Thus and unlike RAR, the performance of the neural network could be increased, by maintaining same training time.

#### Initial value of inverse problem

Before starting to calculate the inverse problem, it is necessary to specify the initial value of unknown BCs. One needs to specify the initial value based on a priori knowledge. Fig. 6 shows the approximation of unknown heat flux on a HHC with different initial values. As illustrated, it can be observed that the accuracy of initial

assumption does not necessarily influence the convergency of heat flux, which means the neural network is less sensitive to erroneous assumptions of the initial value. This is evident especially for overestimations, because the optimizer tends to decrease the unknown parameter on an almost constant speed at the early stage of the training. If the initial estimation is lower than the ground truth, the training time seems to increase with a tendency of underestimation of final prediction though. Therefore, it is recommended to set initial values over the expected heat flux based on a priori knowledge. Such phenomenon also applies to other geometries. Because other geometries are even simpler, the PGNN can recover much faster even if the initial value is low.

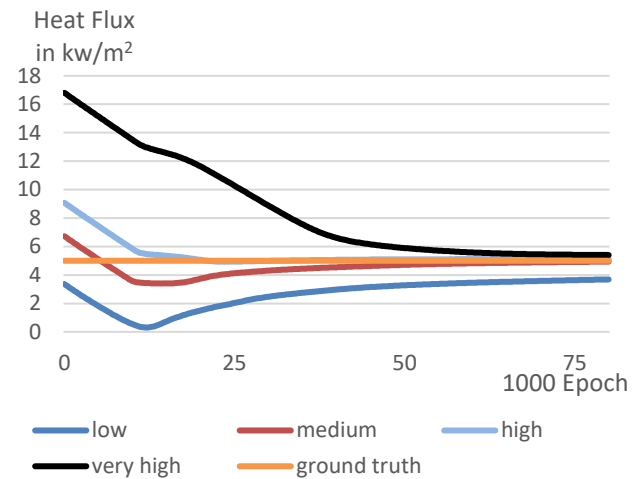


Fig. 6 : Inverse results with different initial assumption.

### 3 RESULTS

At first, this section presents achieved inverse results for presented virtual use-cases (3.1). Subsequently, the temperature distribution reconstructed with the BC from inverse calculation is presented (3.2).

#### 3.1 Inverse prediction of unknown BC

Fig. 7 demonstrates the evolution of estimation of unknown heat flux with increasing training epochs (the number of iterations of the neural network running through the whole dataset) on different geometries.

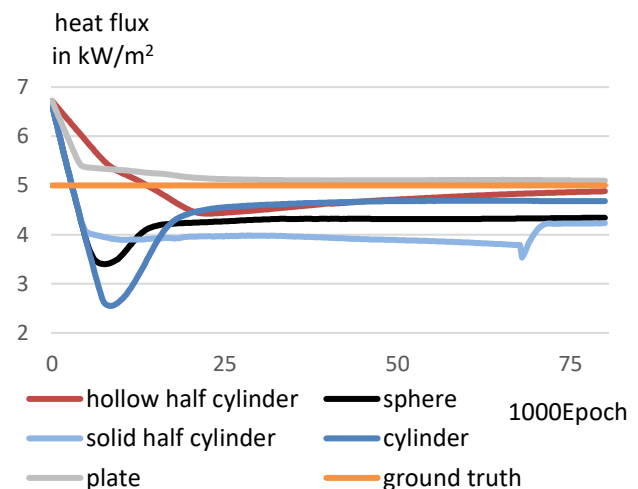


Fig. 7: Inverse results for different geometries.

The chosen stop criterion is the sum of absolute change in approximated heat flux  $< 25 \text{ W/m}^2 / 2000$  epochs. The reason for considering the sum of a period instead of

change between individual epochs is to avoid training stop at turning points, where the estimation shifts from decreasing to increasing. It is noticed that the estimation of HHC is much more elaborate than other geometries, with the neural network converging after much longer epochs. For comparison, the training epoch of all other geometries has been prolonged to the same as of HHC.

For the inverse problem, the amount and positioning of sensors is of vital importance regarding the accuracy of results. For some geometries like full cylinder, sphere and SHC an underestimation for determination of heat flux could be observed. This is influenced by the fact, that the simulated timeframe is relatively short and the thermal information inside the geometries has not been captured. During testing, it could be observed, that if additional data near the heated area and inside the geometry is given, this underestimation can be alleviated. For further research, this problem might affect especially workpieces of thick solid bodies. In contrast, the results for plate and HHC are more accurate, since they are thin in the direction of heat flux.

### 3.2 Rebuilt temperature distribution

To determine the unknown BCs, the overall temperature distribution is directly calculated, which is depicted in Fig. 8.

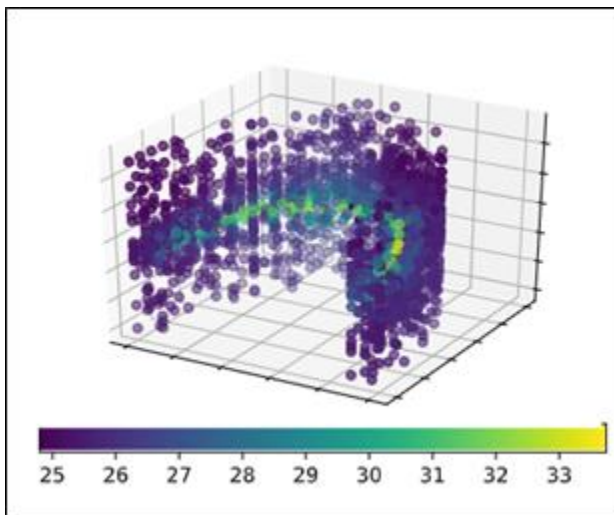
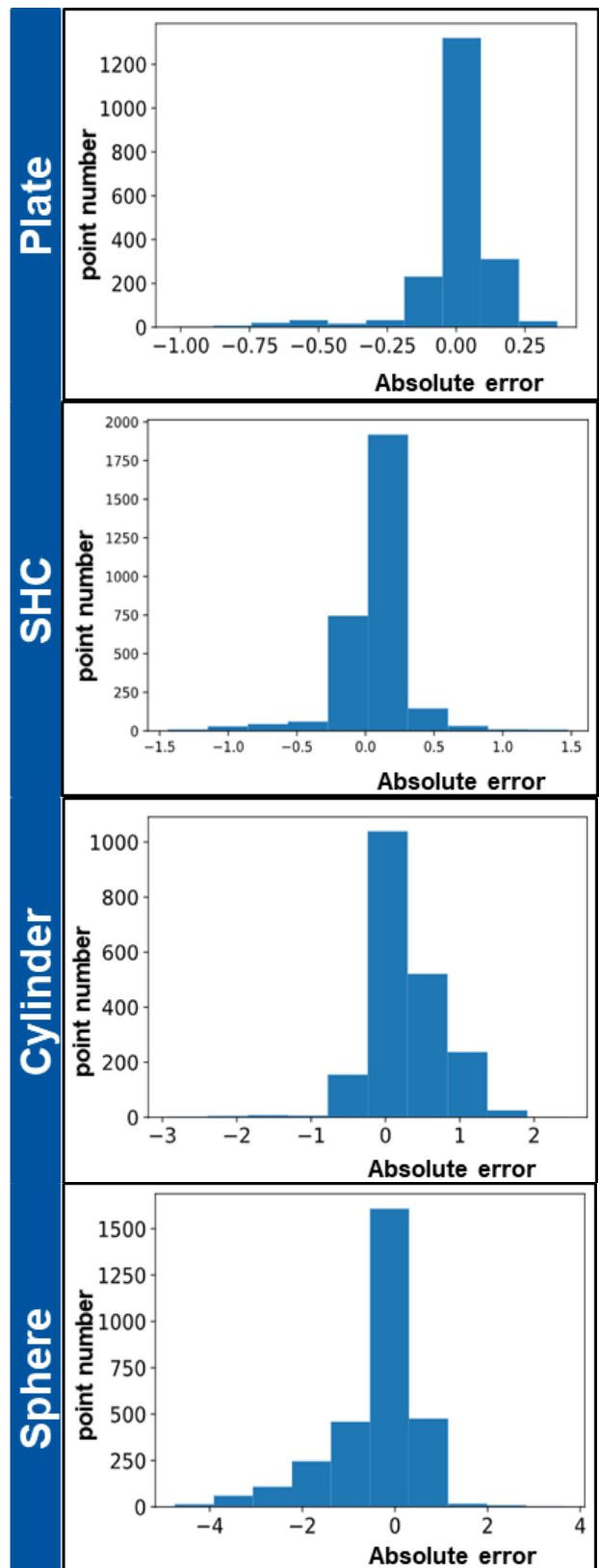


Fig. 8: Temperature distribution of HHC for inverse solution.

It represents the output of neural network in a qualitative way, where temperature gradients are successfully constructed for the HHC, in which the medium front area is heated up while non-heated areas remain at ambient temperature, due to convection. This could be observed for all geometries, but is only demonstrated for the HHC due to lack of space.

For analysis, the temperature derived from PGNN for inverse calculation is compared to the results coming from FEM, where the heat flux was set up in advance using Ansys® FEM software. However, it is noticeable that points with deviations (ranging between +5°C and -2°C) could be observed for all use-cases. Those extreme points represent a clear minority though, which allows the conclusion that the general approach is feasible for the intended purpose (Fig. 9). Currently smoothing these points with their neighbors seems to be an approach, whereat further investigations are necessary.



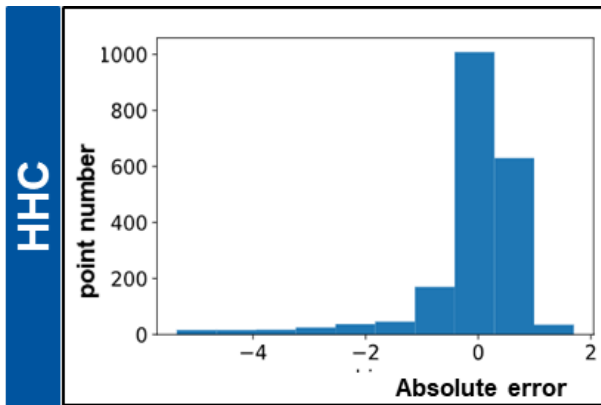


Fig. 9 : Distribution of absolute thermal errors against FEM.

The relative error (L2) and mean absolute percentage error (MAPE) between derived temperature from PGNN and FEM are shown in Fig. 10. Although only a subset of the geometry experiences a heat flux, the authors still apply statistic metrics to the whole body. That is due to the following two reasons: 1) During heating convection acts on the whole surface as continuous BC; 2) Any division to extract sections, where temperature change is significant involves subjective judgement because the temperature distribution is continuous on the geometry. For the inverse problem magnitudes of lower than 2.5% could be achieved in all cases. Results show relatively lower errors for the plate and SHC. A presumption is that the boundary normal on the flat surface of these geometries is homogenous, and thus makes it easier for the neural network to approximate.

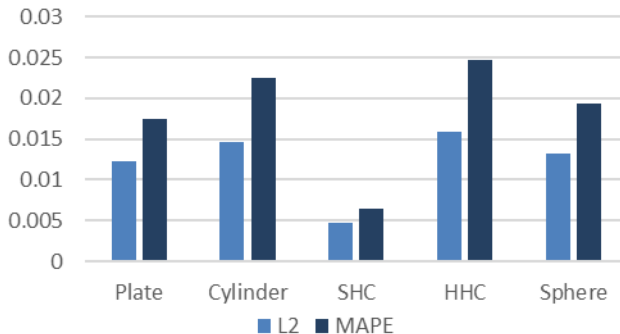


Fig. 10 : Metrics for PGNN output.

#### 4 CONCLUSION AND OUTLOOK

This paper presents a prototype for efficient direct and inverse thermal simulation based on physics guided neural networks (PGNN), providing convenient sensor integration and remarkable accuracy. Despite significant progress in transient thermal modeling in the context of production engineering, reviewing the state of the art reveals weaknesses of entrenched methods, especially related to the efficient inverse heat transfer modelling of complex geometries and flexible sensor integration, which was the motivation for the study. The challenge of the presented work was to develop a tool, aiming to bypass the drawbacks of existing methods by fusing machine learning, analytical knowledge and virtual measurements into one prototype.

The first task was to take up a deep learning library for solving differential equations and modify setting levers to solve inverse heat transfer problems according to

presented use-cases. In the course of that, a geometry interface was developed, enabling to connect complex geometries from external file formats with the software. Besides, a flexible sensor integration was implemented within the geometry interface. After defining the physical problem in terms of PDE and BCs, the neural network architecture was designed. Subsequently, the performance of PGNN was tested virtually for five different geometries in transient-thermal FEM analyses. The achieved results show that the approach of using PGNNs can achieve low overall error with both MAPE and L2 error below 2.5% regarding basic geometries. However, locally thermal errors of higher than 5°C could be observed, which must be further investigated in the future.

Pushing the work closer to real scenarios and improve the performance of PGNN, physical testing will be in the focus of future research. Therefore, a physical test bench is designed. A turbine housing, which is a modified cast blank standing in the shop floor of the research facility of Production Metrology and Quality Management of WZL RWTH Aachen serves as physical demonstrator. To introduce defined heat fluxes on dedicated positions, the housing is equipped with heat pads on the surface, which can be controlled remotely. Furthermore, various temperature sensors comprised of PT1000 elements and a thermography camera ensure a holistic temperature monitoring during the testing.

Additionally, the neural network accuracy, which could only be validated for short simulation timeframes so far (600s), should be tested for larger timeframes. One way is to separate a long time frame into several smaller frames and calculate them in series or even parallel on different GPUs [Meng 2020]. Currently the PGNN is only able to calculate constant parameters in the inverse problem, the interaction among parameters in time-varying BCs will undermine the approximation. To get better output in more realistic scenario, adopting new network architectures, such as a graph neural network GNN [Sanchez-Gonzalez 2020] is a potential strategy. Although this study demonstrates, that already one PGNN design is capable of solving similar thermal simulations for different geometries within same network, the influence of architecture and hyperparameters should be evaluated in terms of further generalization. Meta-learning [Finn 2017] might be a potential way to automate the hyperparameter selection and generalize the representative ability of the neural network in the future.

#### 5 ACKNOWLEDGMENTS

The authors would like to thank Prof. Walter Reichert and Prof. Hans-Jürgen Raatschen from the University of Applied Sciences Aachen for the supervision of student theses, decisively contributing to the presented research.

#### 6 REFERENCE

- [Huang 2009] Huang, Chenghung; Chaing, Mengting. A three-dimensional inverse geometry problem in identifying irregular boundary configurations. *International Journal of Thermal Sciences*, 2009, Vol.48, No.3, pp 502513. ISSN 12900729.
- [Kim 2019] Kim, Byungsoo, et al. Deep Fluids: A Generative Network for Parameterized Fluid Simulations. *Computer Graphics Forum (Proc. Eurographics)*, 2019, Vol.38, No.2, pp 5970.
- [Goodfellow 2016] Ian Goodfellow, et al. *Deep Learning*, MIT Press, 2016, ISBN 978-0262035613.

- [Lu 2020] Lu, Lu, et al. DeepXDE: A deep learning library for solving differential equations. 1907.04502. arXiv, 2020.
- [Ess 2012] Ess, Markus, et al. Dynamic Loads and Thermal Errors on Machine Tools, 2012.
- [Guo 2018] Guo, et al.,. Dynamic Task Prioritization for Multitask Learning. In: Ferrari, Hebert et al., ed. European Conference on Computer Vision. cham: Springer International Publishing, ISBN ISBN 978-3-030-01270-0.
- [Alexandre 2018] Alexandre, M. Tartakovsky, et al. Learning Parameters and Constitutive Relationships with Physics Informed Deep Neural Networks. arXiv, 2018.
- [Holl 2018] Holl, Philipp, et al. Learning to Control PDEs with Differentiable Physics. 2001.07457. arXiv, 2018.
- [Sanchez-Gonzalez 2020] Sanchez-Gonzalez, Alvaro, et al. Learning to Simulate Complex Physics with Graph Networks. 2002.09405. arXiv, 2020.
- [Finn 2017] Finn, Chelsea, et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. 1703.03400. arXiv, 2017.
- [Ohlenforst 2019] Ohlenforst, Markus. Model-based thermoelastic state evaluation of large workpieces for geometric inspection. Aachen, ISBN 3863597524.
- [Zhu 2019] Zhu, Yinhao, et al. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, 2019, Vol.394, pp 5681.
- [Karpatne 2018] Karpatne, Anuj, et al. Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. arXiv, 2018.
- [Meng 2020] Meng, Xuhui, et al. PPINN: Parareal physics-informed neural network for time-dependent PDEs. Computer Methods in Applied Mechanics and Engineering, 2020, Vol.370, pp 113250. ISSN 0045-7825.
- [Martín 2016] Martín, A., et al. TensorFlow: A System for Large-Scale Machine Learning. In: Martín Abadi, ed. Proceedings of the 12th USENIX Symposium on Operating Systems Design. Savannah, GA, USA. 2-4 November, ISBN ISBN 9781931971331.
- [Yang 2017] Yang, Bingru, et al. Thermal compensation for large volume metrology and structures. International Journal of Metrology and Quality Engineering, 2017, Vol.8, pp 21. ISSN 2107-6847.
- [Ross-Pinnock 2018] Ross-Pinnock, David; Mullineux, Glen. Thermal compensation using the hybrid metrology approach compared to traditional scaling. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2018, Vol.232, No.13, pp 23642374. ISSN 0954-4054.
- [Li 2017] Li, Tiemin. Thermal error modeling and compensation of a heavy gantry-type machine tool and its verification in machining. The International Journal of Advanced Manufacturing Technology, 2017pp 120. ISSN 1433-3015.
- [Burghold 2017] Burghold, E. M., et al. Transient contact heat transfer measurements based on high-speed IR-thermography. International Journal of Thermal Sciences, 2017, Vol.115, pp 169175. ISSN 12900729.