Coupled Systems Mechanics, *Vol. 11, No. 1 (2022) 43-54*
https://doi.org/10.12989/csm.2022.11.1.043

Towards a reduced order model of battery systems: Approximation of the cooling plate

Anna Szardenings*[1], Nathalie Hoefer[1a] and Heike Fassbender[2b]

[1]*Battery System Development, Volkswagen Group Components, Gifhorner Strasse 180, 38112 Braunschweig, Germany*
[2]*Institute for Numerical Mathematics, TU Braunschweig, Universitaetsplatz 2, 38106 Braunschweig, Germany*

(Received June 6, 2021, Revised July 6, 2021, Accepted November 11, 2021)

Abstract. In order to analyse the thermal performance of battery systems in electric vehicles complex simulation models with high computational cost are necessary. Using reduced order methods, real-time applicable model can be developed and used for on-board monitoring. In this work a data driven model of the cooling plate as part of the battery system is built and derived from a computational fluid dynamics (CFD) model. The aim of this paper is to create a meta model of the cooling plate that estimates the temperature at the boundary for different heat flow rates, mass flows and inlet temperatures of the cooling fluid. In order to do so, the cooling plate is simulated in a CFD software (ANSYS Fluent ®). A data driven model is built using the design of experiment (DOE) and various approximation methods in Optimus ®. The model can later be combined with a reduced model of the thermal battery system. The assumption and simplification introduced in this paper enable an accurate representation of the cooling plate with a real-time applicable model.

Keywords: battery cooling; CFD simulation; data driven model; data sampling; multiple input multiple output system

1. Introduction

In the development of battery systems for electric vehicles, simulation techniques have gained importance in order to evaluate the design and performance of the system for different scenarios. Especially the thermal behavior of the battery needs to be examined, to ensure safety and reliability of the system (Julien *et al.* 2016, Rao and Wang 2011). An important part of the thermal management is the cooling system, which is used to regulate the temperatures of the battery cells. Computational fluid dynamics (CFD) is commonly used to analyze such high dimensional systems. To evaluate the temperature inside the battery transient simulations over a certain driving cycle are necessary. However, such CFD-simulations are not always feasible due to their high computational cost.

In order to be able to evaluate the temperatures inside the battery in real time, it is therefore necessary to create a lower dimensional model of the battery system. This model can then be used

*Corresponding author, Ph.D. Student, E-mail: anna.szardenings@volkswagen.de
[a]B.Sc. Student, E-mail: nathalie.hoefer@volkswagen.de
[b]Professor, E-mail: h.fassbender@tu-bs.de

http://www.techno-press.org/?journal=csm&subpage=8 ISSN: 2234-2184 (Print), 2234-2192 (Online)

on the battery management system for control and regulation (Szardenings 2022, Szardenings *et al.* 2020, Sancarlos *et al.* 2020). One part of this reduced model is the cooling plate, which is the focus of this work. The cooling plate model is built to calculate the heat transfer from the thermal battery system into the cooling plate.

Previous works focus on estimating the fluid parameters such as pressure drop and velocity using data driven models (Schenck and Fox 2018). For the estimation of heat transfer geometrical input parameters have been used for the meta models, using up to 15 input parameters (Xie *et al.* 2007, Islamoglu 2003, Park *et al.* 2006).

However, here a fixed cooling plate geometry is considered and thus, input and output parameters differ from previous works. The aim is to create a real-time applicable model of the cooling plate. The model is created in order to calculate an accurate boundary condition from the cooling plate. Later this model can be coupled with a thermal model of the battery system. Much more input parameters are needed for this application compared to other heat transfer models which will be described in Section 3. Further, this paper proposes several assumptions and simplifications to the cooling plate.

2. Data driven models

In this section the design of experiments (DOE) methods and meta modeling methods for the development of a data driven model are described. The methods will be later used for the approximation of the cooling plate.

2.1 Design of experiments

The data set for the meta model is created using a DOE. In a DOE a set of input parameters is varied within its range in order to capture the design space of dimension k of the model. Simulations or experiments are carried out for each input parameter variation to generate a snapshot of output parameters.

Different methods exist to define input parameter combinations for validation and test data. Here, a large number of input parameters is necessary. One examined method is Latin hypercube sampling (LHS). LHS is a space filling sample design. From all input parameter permutations, the LHS chooses randomly a number of permutations, such that each sub interval of each parameter is only included once in the sampling size. This reduces the number of input parameter variations. Here the number of experiments n_e is not dependent on the dimension of the design space k (Santner *et al.* 2003). Further, central composite designs (CCD) are evaluated. CCD is based on a full fractional design, where two or more variations per model parameter x_i with equidistant distance between those are examined, resulting in 2^m experiments. This allows to examine main effects (x_1, x_2) and interactions (x_1x_2) between the input parameters. More practical are variations with a fractional factorial design, where m_l parameter combinations are left out. In CCD the full fractional design is combined with star points in between to determine the curvature of the design space. A fractional design is built with two design points for each parameter (2^k) and equidistant distance between those. Star points are added in a distance α from the center of the fractional design. For $\alpha = 1$ this results in a face centered composite design (CCF). The advantage is that the full parameter space is examined and quadratic effects can be evaluated. The method results in a total of $2^m + 2^{m-m_l}$ experiments (Santner *et al.*2003, Simpson *et al.* 2001, Noesis Solution 2020).

2.2 Meta modelling methods

Data driven models approximate a correlation of inputs and outputs, for example based on simulation results, by using numerical methods. They can be built without knowing the physical equation of the underlying problem. Once trained they can be used to reproduce the outputs for various input combination that can be different from the training data set. Also, the models tend to have much lower computational cost than a CFD simulation model itself (Kleijnen 2009). Various approaches have already been studied for the approximation of fluid and heat transfer models. For data driven models especially deep neural networks (DNN) for the approximation of the navier-stokes equationand fluid dynamics have been emphasized (Schenck and Fox 2018, Kutz 2017). Also, other non- linear systems have been identified through neural networks (Prasad and Bequette 2003). For the approximation of heat transfer problems neural networks have been successfully applied using mul-tiple input and output parameters from test data or simulation (Peng *et al.* 2020, Islamoglu 2003, Xie*et al.* 2007, Thibault and Grandjean 1991, Jambunathan *et al.* 1996). A further meta model in fluid dynamics and simulation is the kriging method, which is mostly used for optimization problem andis especially suitable for deterministic simulation models (Park *et al.* 2006, Kleijnen 2009).

In general, a meta model calculates an output $\hat{\boldsymbol{y}} = \boldsymbol{g}(\boldsymbol{x})$ that has a minimal error ϵ compared to the exact solution of the simulation $\boldsymbol{y} = f(\boldsymbol{x})$:

$$\boldsymbol{y} = \hat{\boldsymbol{y}} + \epsilon \ . \tag{1}$$

Here, $\boldsymbol{x}$ is a vector of *n* independent variables with upper and lower bounds that spans the design space of the model. The function $\boldsymbol{g}(\boldsymbol{x})$ can be a combination of linear and non-linear functions and algorithms depending on the chosen method. The parameters of the meta model are trained using the input-output data set from a DOE. During the training process the meta model is designed not only to map the training data set but also all possible solutions in the design space of the simulation model (Santner *et al.* 2003, Giunta and Watson 1998).

Interpolative models use polynomial functions to fit data points from original model in a certain interval. Kriging models are one example. Here, an output average μ for the parameter domain is calculated from the DOE and is extended through a noise function $\boldsymbol{Z}(\boldsymbol{x})$. This correlation function describes the correlation of the output to the mean value and its parameters can be computed through the evaluation of the correlation matrix of the data set. For an universal Kriging method the output average can also be replaced through a regression model $\boldsymbol{g}(\boldsymbol{x})$, such as a polynomial function (Jeong *et al.* 2005, Ryu *et al.* 2002, Kleijnen 2009):

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{Z}(\boldsymbol{x}). \tag{2}$$

Another interpolative method is radial basis function (RBF). The meta model is built from the distance between two design points in the design space, in the assumption that neighboring values affect each other. This is done by calculating the euclidean norm between two points and applying a basis function to the norm:

$$\boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda_i \phi(||\boldsymbol{x} - \boldsymbol{x}_i||) \, . \tag{3}$$

Where $\boldsymbol{x}$ are design points from the training data set, λ_i are the weights to be solved, $r = ||\boldsymbol{x} - \boldsymbol{x}_i||$ is the distance from the origin $\boldsymbol{x}_i$ and ϕ is the basis function. The meta model builds a linear system of equation for a n-dimensional problem of the form $\boldsymbol{A}\lambda = \boldsymbol{b}$, where the elements of **A** are defined as $a_{ij} = \phi(||\boldsymbol{x}_i - \boldsymbol{x}_j||)$. Different polynomial approaches have been

studied for the kernel function ϕ. Most commonly multi quadratic functions are used with $\phi(r) = \sqrt{r^2 + \gamma}$ with γ being a positive shape parameter (Skala 2017, Giunta and Watson 1998, Volpi *et al.* 2015, Buhmann 2000).

Neural networks (NN) describe another method for creating meta models. In neural networks input and outputs correlation are described through a concatenation of functions, whose parameters are trained by a set of training data. They consist of an input layer, containing an array of input parameters $\boldsymbol{a}^{[0]}$, L hidden layers, defining the deepness of the network, and an output layer $f(\boldsymbol{x}) = \boldsymbol{a}^{[L]}$. Each of the hidden layer represents a linear or non-linear function *h* that is applied to the input vector $\boldsymbol{z}$ of the layer

$$\begin{aligned} \boldsymbol{z}^{[l]} &= \boldsymbol{W}^{[l]}\boldsymbol{a}^{[l-1]} + \boldsymbol{b}^{[l]} \\ \boldsymbol{a}^{[l]} &= \boldsymbol{h}^{[l]}(\boldsymbol{z}^{[l]}) \,. \end{aligned} \tag{4}$$

Where $\boldsymbol{a}^{[l-1]}$ is the output from the previous layer, $\boldsymbol{W}^{[l]}$ are the weights and $\boldsymbol{b}^{[l]}$ are the biases. Other important parameters for the training of NNs are the learning coefficient ($a \leq 1$) and the numberof epochs *E*. *E* defines the number of times the NN is trained with a set of data from the defined pool of training data. a defines how the parameters (weights and biases) of the NN are updated in an epoch. Some advantage of NN's are that they are able to represent high non-linear behavior of a data sample and are able to handle many input and output parameters (Xie *et al.* 2019, White *et al.* 2019, Goodfellow *et al.* 2016). Neural networks are often classified into shallow neural network (SNN) with only one hidden layer and deep neural networks (DNN) with an adaptive number of hidden layers greater than one.

The data driven models can be evaluated by comparing the training error ϵ_{train}, defining how good the model can replicate the training data set and the generalization error ϵ_{test}, describing the ability to approximate data of the design space different from the training set (Goodfellow *et al.* 2016). The errors are calculated in comparison with the results from the CFD simulation.

3. Cooling plate model

The considered CFD simulation model in our work is the cooling plate of a battery system which is placed beneath the battery cells. This bottom cooled system uses a water based coolant. The cooling plate consists of several channels that are optimized in their design to ensure a uniform temperature distribution in the battery system (Pesaran 2001, Rao and Wang 2011).

3.1 CFD simulation model

The cooling plate is simulated in a CFD software (ANSYS Fluent ®), with the goal to calculate the heat transfer from the battery system into the fluid. Hence, several assumptions and simplifications can be made to the model. For the fluid a fully developed flow is assumed at every time step at a certain flow rate and temperature. A fluid volume needs approximately 3 sec. at maximum mass flow rate of 0.1776 kg/s to fully develop a steady flow in the channels. Vice versa when changing the mass flow to 0 kg/s the fluid is motionless in about 3 sec.. With this assumption a change to higher mass flow rate would result in slightly higher overall heat transfer, vice versa a change to a lower mass flow rate results in lower heat transfer. Thus, the errors would cancel out in a steady state simulation. Further, the fluid is assumed to be laminar, with a maximal Reynolds Number

of 790 in the channels.A steady and stable flow pattern and no vortexes are expected (Baehr and Stephan 2006). Through this assumption it is possible to calculate the flow stationary, with velocity and pressure values inde-pendent from time.

In the coupled battery system model the cooling plate and thermal battery system are coupled through the heat flow rate of neighbouring faces, satisfying the following equation (Annaratone 2010, Weigand *et al.* 2013):

$$\dot{q} = \alpha \cdot \left(T_{plate} - T_{bat}\right). \tag{5}$$

Where $\dot{q}$ is the specific heat flow rate through a face of area *A*, α the heat transfer coefficient, T_{plate} isthe temperature at the cooling plate top and T_{bat} is the temperature at the bottom of the battery system. When separating the cooling system from the thermal model, each of those values can be used as aconservative boundary condition.

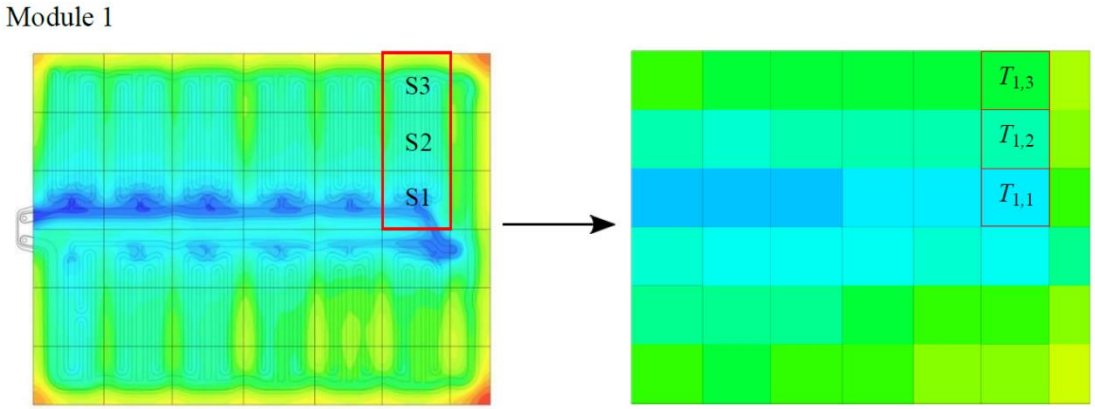

Fig. 1 Approximation of the cooling plate with averaged values (Here temperature T). Each surface beneath a battery module is split in three surfaces (S1, S2 and S3) on the cooling plate

Table 1 Temperature deviation in thermal battery model for different BCs

	BC HTC		BC Heat flow rate		BC Temperature	
	ΔT Abs.	ΔT Rel.	ΔT Abs.	ΔT Rel.	ΔT Abs.	ΔT Rel.
Average	0.77 K	0.9%	2.01 K	3.4%	-0.23 K	0.8%
Maximum	2.79 K	4.8%	4.41 K	7.5%	1.63 K	2.8%

In the cooling plate model the values from Eq. (5) are calculated at the interface of the cooling plate to the battery modules. The values are averaged over three areas per module interface to represent the different flow sections of the cooling plate. The simplification is shown in Fig. 1. The three surfaces per module are chosen to depict the different areas under each module. This is sufficient forthe correct calculation of the temperature in the battery system, as will be shown later. The approximation results in a total of 42 surfaces

For validation, the thermal battery system was simulated with averaged boundary conditions (BC) from the cooling plate and the resulting temperatures are compared with a coupled model of the system (cooling plate and thermal system). The averaged values at the cooling plate faces are

obtained from separate simulations of the CFD model of the cooling plate. Three different types of boundary conditions are tested: fixed temperature, heat transfer coefficient (HTC), heat flow rate boundary. The results are shown in Table 1. The temperature BC shows the best results in the thermal model.

Based on the previous assumptions an averaged error of -0.23 K is estimated in the batteries temperatures when comparing the full battery system model and the decoupled model. The maximal loca; temperature deviation in the thermal battery system is 1.63 K compared to the coupled system. The results also show, that an approximation of the cooling plate surface with three surfaces per battery module is sufficient. Hence, the cooling plate simulation model is set up to have the temperature at the plate as an output parameter.

3.2 Approximation model

To create a real-time applicable model of the cooling plate, data driven modelling methods are evaluated. In order to do so, snapshots are calculated from the fluid model for each input parameter set using a DOE. The collected training data is used to build the meta model. Further, the model is validated using a test data set different from the training data. The DOE and meta model is built in Optimus ®. The software is chosen as tool, since it can directly be coupled with Ansys Fluent ® simulations to generate the snapshots and is able to provide a solver independent functional mock up unit (FMU) of the model, that can be imported into any software and model of choice (Noesis Solutions 2019, Noesis Solution 2020). The work-flow is shown in Fig. 2. With an input array defined by the design point in the DOE the simulation set up is parameterized and applied to a CFD model loaded through case and data file. The simulation is performed on a high performance cluster and the output is written back into an array. As discussed in 3 the cooling plate surface is split into 42 smaller surfaces. Here, input parameters are the fluid temperature at the inlet T_{fluid}, the mass flow rate at the inlet $\dot{m}_{fluid}$ and the 42 heat flow rates $\dot{q}_{cooling,i}$ at the cooling plate surface boundary condition. The input parameter boundaries are shown in the Table 2 and define the design space for the DOE. As output parameters, the 42 averaged surface temperatures ($T_{plate,i}$) at the cooling plate are evaluated through the CFD simulation as described in Fig. 1.

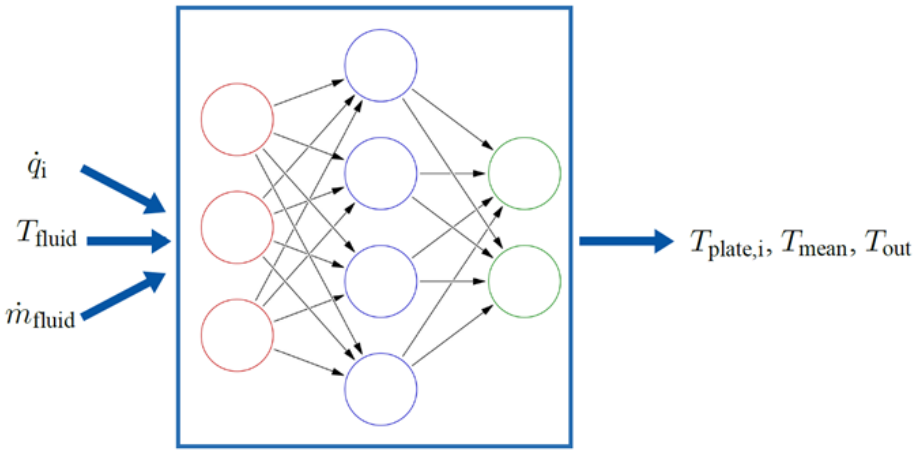

Fig. 2 Input and output parameters of the data driven model with i={1, 2, ..., 42}

Table 2 Upper and lower boundaries of the input parameters, as well as nominal value

Parameter	Lower boundary	Upper boundary	Nominal
Heat flow rate [W/m²]	0	4000	2000
Mass flow rate [kg/s]	0	0.1776	0.1776
Inlet temperature [K]	293.15	308.15	298.15

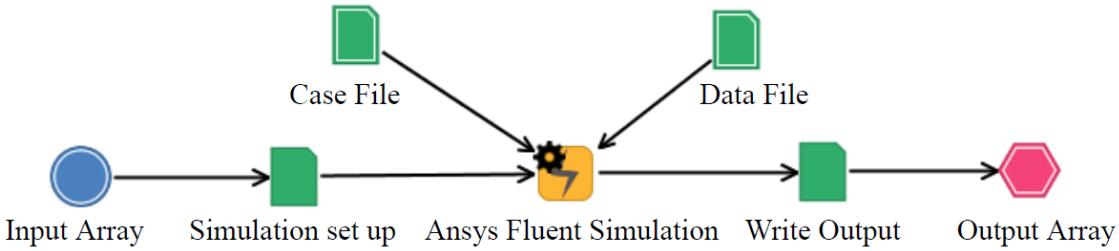

Fig. 3 Optimus work-flow for the calculation of snapshots from the Ansys Fluent model

Also, for validation reasons the average and outlet temperature of the cooling fluid are calculated. In total this results in 44 input and output parameters.In our case the simulation model used is deterministic, meaning, the output parameters for a fixed input parameter set is always the same.

The schematic view of the data driven model with input and output parameters is shown in Fig. 3.

For comparison the following regression parameters and error measures are used in order to evaluate the quality of the meta model compared to the CFD simulation regarding the training and test data.

The sum of squared errors defines the deviation between model and simulation $SSE = \sum_{i=1}^{n}(y - \hat{y}_i)^2$ and the total sum of squares the deviation from the mean data output $\bar{y}$ with $SSTOT = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2$. The degree of determination (R^2) and the mean squared error (MSE)with the number of experiments n and number of model parameters m are then defined as (Noesis Solutions 2019):

$$R^2 = \frac{SSE}{SSTOT} \quad \text{and} \quad MSE = \frac{SSE}{n-m} \,. \tag{6}$$

Further the mean deviation is calculated as

$$\Delta\bar{y} = \frac{\sum_{i=1}^{n} y_i - \bar{y}_i}{p} \,. \tag{7}$$

For a model validation, beside the training data, nine input parameter sets are defined as test data. Three parameter variations in the mass flow rate, the inlet temperature and the heat flux are evaluated. The meta model prediction is then again compared to the CFD results for this parameters. Here, the *MSE* and the temperature deviation are evaluated.

4. Results and discussion

In this section the results of the four most promising meta models are discussed. For the DOE a

CCF method was chosen and 87 design points are considered. Further, a LHC method with a total of 300 experiments has been conducted, from which 252 points are considered. Both DOE methods are tested in combination with different meta modeling methods.

The Table 3 shows the resulting mean square errors and degree of determination of the meta models evaluated with the training data set from a DOE sampling method. The best model quality with training data is achieved for the radial basis function (RBF) in combination with the face centered composite design (CCF). The Kriging method in combination with the CCF gives the least promising results.

Table 3 Model accuracy for different DOE methods and meta models based on the training data

	CCF RBF	CCF Kriging	LHC DNN	LHC Kriging
MSE	1.309e − 8	0.177	0.017	0.096
R^2	≈ 1	0.909	≈ 1	0.998

Table 4 Model accuracy based on a test data set different from the training data defined through *MSE* and average temperature deviation between meta model and CFD simulation

	CCF RBF	CCF Kriging	LHC DNN	LHC Kriging
MSE	0.82	0.83	12.18	2.53
ΔT_{mean} [K]	0.44 ± 0.07	0.45 ± 0.08	2.64 ± 2.37	0.76 ± 1.52
ΔT_{max} [K]	3.54	3.54	11.05	15.38

In Table 4 the meta models are evaluated for a parameter set different from the training data. MSE and temperature deviation between the CFD model and the meta model are displayed. Again the RBF and CCF combination gives the best results with an average temperature deviation of 0.44 K. Further, the Kriging model with the CCF gives better results than the other two combinations, in contrast to Table 3. The DNN and Kriging with LHC show on average an acceptable model quality, with 2.64 K and 0.76 K temperature deviation. However, the models fail to predict individual data points, with maximal deviations up to 11.05 K and 15.38 K. For the deep neural network it can be assumed that the good approximation of the training data and worse performance on the test data results from an over-fitting of the model. The DNN is set up with three hidden layer and a total of 1024 neurons, both parameters optimized by the software. A SNN in combination with the LHC show even lower model quality on average (ΔT_{mean}= 4.41 K). An increase in the number of design points in the LHC sample to 412 does result in on average lower temperature deviation, but worsens significantly the temperature prediction for some input combinations. The optimal number of neurons and layers for the NN and the optimal training data set are to be evaluated in future investigations. So far the best results were obtained when the number of neurons was optimized through Optimus ®by setting the accuracy-speed trade-off to accurate. Some tests were performed with a NN with less neurons and more layers, the models however showed a higher approximation error in the training aswell as in the test data.

RBF and Kriging are both constructed through the assumption of parameter correlations (Giunta andWatson 1998). Here, the temperatures and heat flux on neighbouring surface on the cooling plate areoften similar and depending on each other. This can be one reason why the interpolating methods perform better than the NN on the cooling plate model. Further when comparing the number of

neurons in the NN (1024) to the number of weights λ_i in the radial basis function and number of parameters in the correlation matrix in the Kriging method, it can be assumed that the high number of neurons is related to the over-fitting of NN (Noesis Solution 2020).

In general, the data generated from the CCF method gives a better model quality. One assumption is, that especially the maximal and minimal boundaries of the parameter space are important for the estimation of the temperatures on the cooling plate. With LHC random values are set for the inputs and thus, not all extrema of the design space are included in the training data and could result in the lower model quality.

The evaluation of the DOE and the approximation models also showed that especially for lower mass flow rates the models are not able to approximate the temperatures correctly. Therefore, in the model generation and in the results above all data points with temperature above 333 K are excluded. This can be explained by set up of the CFD model, which has the cooling fluid as only heat sink. Hence,with very low mass flow rates and constant heat source the stationary solution of the simulation model results in very high temperatures. It could be noticed that temperatures are not physically reasonable below a mass flow rate ≈ 0.02 kg/s. Thus, the resulting approximation model is only valid for higher mass flow rates and the lower boundary needs to be adapted.

To overcome this problem another approach is proposed for very lower mass flow rates as describedin Fig. 4. Depending on the mass flow rate of the cooling fluid either the meta model as FMU is solved to calculate the cooling plate temperatures or a heat capacity model of the cooling plate and fluid is used. In the heat capacity model the following equation is solved:

$$\Delta T = \sum \dot{q} \cdot A \cdot \frac{\Delta t}{c_p} \tag{8}$$

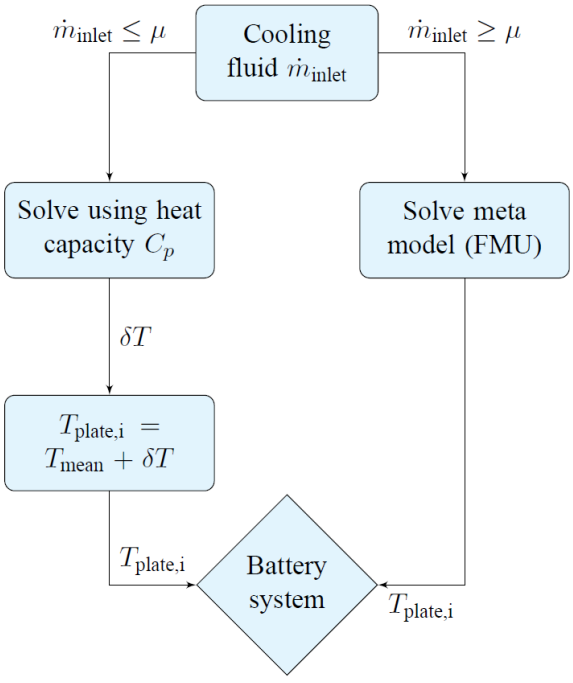

Fig. 4 Decision tree for coupling the cooling plate with thermal model of the battery system based on a minimal mass flow rate μ

Where C_p is the total heat capacity of the cooling plate, A the total surface area, $\sum \dot{q}$ the sum of all heat fluxes and δT the resulting temperature change. The temperature change is then added to the mean cooling plate temperature that was calculated from the meta model or thermal capacity model in a previous step. The resulting temperature at the cooling plate ($T_{\text{plate,i}}$) is coupled in both cases with the battery system.

5. Conclusions

The main advantage of the proposed method is that the output from cooling plate model, mainly the boundary condition for the thermal model, can be reduced to few output values instead of a field value, resulting in an error of 0.23 K in the thermal model. The results show that it is possible to reconstruct the temperature behavior of the cooling plate through a data driven model with an accuacy of 0.44 K in the test data set. This is achieved by using a CCF design for the generation of the training data (88 experiments) and a RBF method as a meta modelling method. The Kriging method showed similarly good results (0.45 K) in combination with the CCF method. The neural network and Kriging model in combination with the LHC showed good results on the training data, but failed to predict the output parameters for the test data. The investigation also demonstrated that some approximation models are very sensitive to the training data and others are not. The results further showed that the meta model can only be used for higher mass flow rates above ≈ 0.02 kg/s. For lower mass flow rates a heat capacity model is proposed to approximate the temperature change due to a heat flux.

6. Outlook

The cooling plate model built in Optimus ® can be exported as standalone FMU and then implemented into another simulation model, for example using Matlab and Simulink ®. The goal is to couple this model with thermal model of the battery system as described in Equation 5. The coupling with the thermal model still needs to be investigated. Also, adjustment in the training data set and simulation set up of the cooling plate model may be necessary. Especially the behavior at lower mass flow rates needs further evaluation.

Finally, more validation and training are needed in order to understand the different behaviors of the meta models and further optimize their quality.

The approach proposed in this paper can also be applied to other simulation models and physical problem, when inputs and output have a spatial dependency. The method is especially useful when only specific input-output correlations are of interest and the simulation model itself takes unacceptably long time to compute and contains much more information and data than necessary. The approximation model is able to predict linear as well non-linear correlation and is useful when the physical simulation model cannot be reduced. The models proposed in Hadzalic *et al.* (2020) or Moreno-Navarro *et al.* (2018) deal with structural deformations due to temperature or electric fields. Here, a data driven model approach could substitute the complex simulation model in order to compute for example the maximal displacement for different inputs. However, the effort for the development of such model needs to be compared to the reduction in the computational cost.

References

Annaratone, D. (2010), *Engineering Heat Transfer*, Springer Science & Business Media.

Baehr, H.D. and Stephan, K. (2006), *Waerme-und Stoffuebertragung*, Volume 5, Springer.

Buhmann, M.D. (2000), "Radial basis functions", *Acta Numerica*, **9**, 1-38. https://doi.org/10.1017/S0962492900000015.

Giunta, A. and Watson, L. (1998), "A comparison of approximation modeling techniques-polynomial versus interpolating models", *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 4758.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y. (2016), *Deep Learning*, Volume 1, MIT Press Cambridge.

Hadzalic, E., Ibrahimbegovic, A. and Dolarevic, S. (2020), "3d thermo-hydro-mechanical coupled discrete beam lattice model of saturated poro-plastic medium", *Couple. Syst. Mech.*, **9**(2), 125-145. https://doi.org/10.12989/csm.2020.9.2.125.

Islamoglu, Y. (2003), "A new approach for the prediction of the heat transfer rate of the wire-on-tube type heat exchanger-Use of an artificial neural network model", *Appl. Therm. Eng.*, **23**(2), 243-249. https://doi.org/10.1016/S1359-4311(02)00155-2.

Jambunathan, K., Hartle, S., Ashforth-Frost, S. and Fontama, V. (1996), "Evaluating convective heat transfer coefficients using neural networks", *Int. J. Heat Mass Transf.*, **39**(11), 2329-2332. https://doi.org/10.1016/0017-9310(95)00332-0.

Jeong, S., Murayama, M. and Yamamoto, K. (2005), "Efficient optimization design method using kriging model", *J. Aircraft*, **42**(2), 413-420. https://doi.org/10.2514/1.6386.

Julien, C., Mauger, A., Vijh, A. and Zaghib, K. (2016), "Lithium batteries", *Lithium Batteries: Science and Technology*, Springer International Publishing, Cham.

Kleijnen, J.P. (2009), "Kriging metamodeling in simulation: A review", *Eur. J. Oper. Res.*, **192**(3), 707-716.

Kutz, J.N. (2017), "Deep learning in fluid dynamics", *J. Fluid Mech.*, **814**, 1-4. https://doi.org/10.1016/j.ejor.2007.10.013.

Moreno-Navarro, P., Ibrahimbegovich, A. and Perez-Aparicio, J.L. (2018), "Linear elastic mechanical system interacting with coupled thermo-electro-magnetic fields", *Couple. Syst. Mech.*, **7**(1), 5-25. https://doi.org/10.12989/csm.2018.7.1.005.

Noesis Solutions (2019), OPTIMUS REV 2019.2 - USERS MANUAL, Noesis Solutions, Gaston Geenslaan 11, 3001 Leuven, Belgium, 2019.2nd Edition.

Noesis Solutions, Gaston Geenslaan 11, 3001 Leuven, Belgium (2020), Optimus 2020.1 Theoretical Background, 1st Edition.

Park, K., Oh, P.K. and Lim, H.J. (2006), "The application of the cfd and kriging method to an optimization of heat sink", *Int. J. Heat Mass Transf.*, **49**(19-20), 3439-3447. https://doi.org/10.1016/j.ijheatmasstransfer.2006.03.009.

Peng, J.Z., Liu, X., Aubry, N., Chen, Z. and Wu, W.T. (2020), "Data-driven modeling of geometry- adaptive steady heat transfer based on convolutional neural networks: Heat conduction", *arXiv preprint arXiv:2010.03854*.

Pesaran, A.A. (2001), "Battery thermal management in ev and hevs: issues and solutions", *Battery Man*, **43**(5), 34-49.

Prasad, V. and Bequette, B.W. (2003), "Nonlinear system identification and model reduction using artificial neural networks", *Comput. Chem. Eng.*, **27**(12), 1741-1754. https://doi.org/10.1016/S0098-1354(03)00137-6.

Rao, Z. and Wang, S. (2011), "A review of power battery thermal energy management", *Renew. Sustain. Energy Rev.*, **15**, 4554-4571. https://doi.org/10.1016/j.rser.2011.07.096.

Ryu, J.S., Kim, M.S., Cha, K.J., Lee, T.H. and Choi, D.H. (2002), "Kriging interpolation methods in geostatistics and dace model", *KSME Int. J.*, **16**(5), 619-632. https://doi.org/10.1007/BF03184811.

Sancarlos, A., Cameron, M., Abel, A., Cueto, E., Duval, J.L. and Chinesta, F. (2020), "From rom of electro-

chemistry to ai-based battery digital and hybrid twin", *Arch. Comput. Meth. Eng.*, **28**(3), 979-1015. https://doi.org/10.1007/s11831-020-09404-6.

Santner, T.J., Williams, B.J., Notz, W.I. and Williams, B.J. (2003), *The Design and Analysis of Computer Experiments*, Volume 1, Springer.

Schenck, C. and Fox, D. (2018), "Spnets: Differentiable fluid dynamics for deep neural networks", *arXiv preprint arXiv:1806.06094*.

Simpson, T.W., Poplinski, J.D., Koch, P.N. and Allen, J.K. (2001), "Metamodels for computer-based engineering design: survey and recommendations", *Eng. Comput.*, **17**(2), 129-150. https://doi.org/10.1007/PL00007198.

Skala, V. (2017), "Rbf interpolation with csrbf of large data sets", *Procedia Comput. Sci.*, **108**, 2433-2437. https://doi.org/10.1016/j.procs.2017.05.081.

Szardenings, A. "Verfahren und vorrichtung zum ueberwachen eines elektrischen energiespeichers, computerprogramm produkt", German Patent DE 10 2020 203 004 A1, to be published in 2022.

Szardenings, A., Petersen, N. and Fassbender, H. (2020), "Concept for thermal analysis of batteries using reduced order modeling", *AIP Conference Proceedings*, **2293**, AIP Publishing LLC.

Thibault, J. and Grandjean, B.P. (1991), "A neural network methodology for heat transfer data analysis", *Int. J. Heat Mass Transf.*, **34**(8), 2063-2070. https://doi.org/10.1016/0017-9310(91)90217-3.

Volpi, S., Diez, M., Gaul, N.J., Song, H., Iemma, U., Choi, K., Campana, E.F. and Stern, F. (2015), "Development and validation of a dynamic metamodel based on stochastic radial basis functions and uncertainty quantification", *Struct. Multidisc. Optim.*, **51**, 347-368. https://doi.org/10.1007/s00158-014-1128-5.

Weigand, B., Koehler, J. and von Wolfersdorf, J. (2013), *Thermodynamik Kompakt*, Springer Berlin Heidel berg, 3rd Edition.

White, C., Ushizima, D. and Farhat, C. (2019), "Neural networks predict fluid dynamics solutions from tiny datasets", *arXiv preprint arXiv:1902.00091*.

Xie, G., Wang, Q., Zeng, M. and Luo, L. (2007), "Heat transfer analysis for shell-and-tube heat exchangers with experimental data by artificial neural networks approach", *Appl. Therm. Eng.*, **27**(5-6), 1096-1104. https://doi.org/10.1016/j.applthermaleng.2006.07.036.

Xie, X., Zhang, G. and Webster, C.G. (2019), "Non-intrusive inference reduced order model for fluids using deep multistep neural network", *Math.*, **7**(8), 757. https://doi.org/10.3390/math7080757.