Noname manuscript No.
(will be inserted by the editor)

Casting Hybrid Twin: Physics-based reduced order models enriched with data-driven models enabling the highest accuracy in real-time

Amine Ammar · Mariem Ben Saada · Elias Cueto · Francisco Chinesta

Received: date / Accepted: date

Abstract Knowing the thermo-mechanical history of a part during its processing is essential to master the final properties of the product. During forming processes, several parameters can affect it. The development of a surrogate model makes it possible to access history in real time without having to resort to a numerical simulation. We restrict ourselves in this study to the cooling phase of the casting process. The thermal problem has been formulated taking into account the metal as well as the mould. Physical constants such as latent heat, conductivities and heat transfer coefficients has been kept constant. The problem has been parametrized by the coolant temperatures in five different cooling channels. To establish the offline model, multiple simulations are performed based on well-chosen combinations of parameters. The space-time solution of the thermal problem has been solved parametrically. In this work we propose a strategy based on the solution decomposition in space, time, and parameter modes. By applying a machine learning strategy, one should be able to produce modes of the parametric space for new sets of parameters. The machine learning strategy uses either random forest or polynomial fitting regressors. The reconstruction of the thermal solution can then be done us-

A. Ammar
LAMPA Laboratory & ESI Group Chair, Arts et Metiers Institute of Technology, 2 boulevard du Ronceray, BP 93525, 49035 Angers cedex 01, France
E-mail: Amine.Ammar@ensam.eu
M. Ben Saada
LAMPA Laboratory & ESI Group Chair, Arts et Metiers Institute of Technology, 2 boulevard du Ronceray, BP 93525, 49035 Angers cedex 01, France
E-mail: Mariem.BENSAADA@ensam.eu
E. Cueto
ESI Group Chair, Aragon Institute of Engineering Research, Universidad de Zaragoza. Maria de Luna s/n, 50018 Zaragoza, Spain
E-mail: ecueto@unizar.es
F. Chinesta
PIMM Laboratory & ESI Group Chair, Arts et Metiers Institute of Technology, CNRS, Cnam, HESAM Université, 151 boulevard de l'Hôpital, 75013 Paris, France
E-mail: francisco.chinesta@ensam.eu

ing those modes obtained from the parametric space, with the same spatial and temporal basis previously established. This rationale is further extended to establish a model for the ignored part of the physics, in order to describe experimental measures. We present a strategy that makes it possible to calculate this ignorance using the same spatio-temporal basis obtained during the implementation of the numerical model, enabling the efficient construction of processing hybrid twins.

Keywords Smart manufacturing · Physics-based modeling · Model Order Reduction · PGD · Data-driven modeling · Artificial intelligence · Hybrid Twin · Casting

1 Introduction

Metal Casting is one of the oldest materials forming technique which is widely employed in industrial environment. It enables manufacturing complex shaped parts with high productivity and less raw consumption [?]. Gravity casting is the simplest form of casting that consists of pouring molten alloy into a mould cavity with no force other than gravity, where it cools and solidifies. The mould can be made of sand, metal or some other materials [?]. The permanent mould casting is a process that uses a metal mould namely tool steel [?], iron and bronze [?]. The most stringent requirement on permanent moulds is their cooling ability. They are characterized by high thermal conductivity which allows to increase the rate of heat transfer and to reduce the solidification time. As a consequence, the produced cast parts present better dimensional tolerances, superior surfaces finishing, and higher mechanical properties [?,?]. In industry, the most up-to-date application of permanent mould casting is the aluminium alloys due to their excellent properties such as excellent cast ability [?], high electrical and thermal conductivity [?,?,?], low density, low weight and high strength to weight ratio [?,?].

To ensure high quality casting products, the casting stages need to be well controlled starting by mould preparation, alloy melting, pouring, and finally solidification process. Inaccurate supervision at these stages leads to casting defects [?]. The cooling stage has a significant effect on the microstructure of the cast parts, which means on their mechanical properties. It is necessary to understand the heat transfer process inside the mould to ensure the required mechanical properties in the casting. The heat released during the solidification is transferred within the mould by conduction. Once the heat reaches the mould walls, it is transferred to the air essentially by natural convection [?]. It is well known that for aluminium alloys, the cooling rate directly affects the microstructure morphology and the size of the grains: Raising up the cooling rate during the casting can significantly refine the microstructure and thereby improve the mechanical properties of produced parts [?,?,?]. The material and geometry of permanent metal mould contribute on the heat transfer process, that is, on the casting cooling [?]. In permanent mould casting, it is highly

recommended to have homogeneous distribution of temperature in the mould. Non-uniform cooling causes defects in the cast parts such as low residual stress, hot spots and distortions in the form [?]. These casting defects could be reduced by using "cooling channels" moulds. They date back to 1990 and were initially suggested for injection moulding [?,?] and then were extended to others fields such as extrusion [?], hot sheet metal forming [?], forging [?], and die casting [?,?,?]. Karakoc et al. showed, in references [?] and [?], that the porosity in the cast parts was reduced by 43% and the average particle size of the cast parts was 13.5% smaller than those parts obtained with standard moulds. Both of these studies were carried out through experimental methods and numerical simulations. Norwood et al. were also used the simulation tools to optimize the design of cooling channels to ensure a high product quality and minimize production costs [?].

Today, numerical simulations are widely used in casting optimization process. However, for an optimized casting configuration, the simulations analyses were generally based on a particular set of parameters. In addition, the requirement of very accurate and reliability data increases significantly the calculation time of numerical simulations that means the computing coasts. Thus, the numerical simulations in casting process is still interesting through the use of artificial intelligence. It is hence possible nowdays in metal casting processes to applied powerful tools and models developed with reasonable number of simulations that allows predicting parts defects and controlling complex processes [?,?,?]. For example, Jiang et al. used back propagation neural network models to establish a relationship between the continuous casting parameters and the cooling rate which was based on secondary dendrite arm spacing compute [?]. They showed that this model has a higher accuracy in the optimization of the continuous casting technology. Others researchers used also artificial neural networks and they were more focused on the cooling-solidification process and the heat transfer coefficient as well [?,?]. Susac et al. applied artificial neural network to predict the thermal field of permanent mould based on the thermal history of the aluminium cast parts [?]. Vasileiou et al. proposed a genetic optimization algorithm aided with numerical simulations to determine the heat transfer values in casting [?]. However, for every new casting change in material and/or in shape, this approach should repeat again. Researchers tried to developpe interesting approaches for thermal field evolution in the cast and in the mould as well. Despite these efforts, most of the proposed approach are limited to the cast part design, casting process parameters, and also to the number of input parameters. The present work proposes a new approach combining physics-based reduced order models, enabling parametric studies, and data-driven model enrichment in the so-called hybrid modelling framework, enabling the highest accuracy with respect to the experimental measurements, while proceeding under the stringent real-time constraint.

1.1 Empowering engineering from the use of surrogates

Efficient design and system control are needed for quick evaluations of the system response for any choice of the parameters involved in the associated model. Usual numerical simulation techniques remain unable to provide results under the stringent real-time constraints imposed by control.

Parametric models, also called surrogates, metamodels or response surfaces, make it possible to attain such feedback rates. Then, on top of these surrogates, simulation, optimization, uncertainty propagation or simulation-based control become attainable even under the stringent real-time constraint. Thus, the challenge of developing efficient simulations is translated into the one of an efficient construction of such surrogates, that is far from being a trivial task.

In fact, if one assumes a multivalued input $\mathbf{X}$ and an associated multivalued output $\mathbf{Y}$, the surrogate is no more than the mapping $\mathbf{Y} = \mathbf{F}(\mathbf{X})$, where $\mathbf{F}(\mathbf{X})$ constitutes the searched model, that in general consists of a linear or a nonlinear regression.

Constructing a regression is not difficult, conceptually speaking. However, the amount of data needed for this purpose strongly depends on the model complexity.

Since complexity will depend on the dimension of the data (number of features involved in $\mathbf{X}$) and variables to model (size of $\mathbf{Y}$), one is tempted to proceed reducing the data dimensionality prior to create the regression.

Data dimensionality reduction can be performed by using a linear reduction—for instance by employing principal component analysis, PCA—or a nonlinear one, making use of manifold learning techniques, for instance, or in a more transparent manner, by training autoencoders able to map the data into a reduced latent space.

Usually in the case of engineering, and more particularly in casting process simulation, where the temperature field is expected to depend on few process parameters (like in this paper the temperature of the fluid flowing into the so-called cooling channels disposed in the mould) we look to infer 3D fields from few features. Thus we firstly need to reduce computer memory storage space and enable real-time predictions for temperature field. Then we can move to creating a regression (linear or non-linear) between the features and the reduced description of the temperature field.

In turn, this regression can be linear (even when non-linear approximation functions are involved) or non-linear. Polynomial linear regressions are very usual, and they were adapted to address multidimensional problems by making use of separated representations [**?**,**?**].

Regularization allows us to address rich approximations while keeping the amount of data to a minimum [**?**]. These situations result, in general, in underdetermined linear systems, that need for appropriate regularizations to avoid overfitting. Elastic Net regularizations combining the Ridge L2-regularization, that prevent overfitting, and the Lasso L1-regularization, that promotes sparsity, are widely adopted [**?**].

When the amount of data is large enough and it is expected to be distributed on a nonlinear manifold, artificial neural networks, ANN, [**?**] become an appealing choice.

1.2 Filling the gap between knowledge and observations: the hybrid twin

A particular situation occurs when physics is solved very efficiently by employing surrogates, whose construction has just been addressed, but a significant gap between the predictions and the observations is noticed. Such a gap reflects the limitations of the considered model, that can be inaccurate or incomplete with respect to the addressed physics. In this situation, two direct alternatives exist: (i) refine the physics-based model to improve the prediction performance; or (ii) correct (or enrich) the physics-based model by adding a data-driven model of the observed gap—something that we refer to as modelling the ignorance (i.e., the gap between measures and simulations). This second route is at the origin of the so-called hybrid-twin concept, addressed in our recent works [**?**,**?**,**?**,**?**,**?**].

The main advantage of this augmented framework is twofold. First it offers the possibility of explaining the (usually) most important part of the resulting hybrid (or augmented) model: the one concerning the physics-based contribution. Second, with a deviation less nonlinear that in the case of the observed reality (the physics-based model contains an important part of such nonlinearity), less data becomes sufficient for constructing the data-driven model.

2 Methods

This section revisits usual surrogate constructors that make use of separated representations, and proposes an appealing alternative that overcomes these. Our objective in this study is to elaborate a parametric solution with a representation compatible with the use of machine learning techniques, so as to enable the prediction of new scenarios associated with arbitrary parameter choices.

2.1 A space-time and parameters separated representation

We consider a field T defined in the physical domain, $\mathbf{x} \in \Omega_x \subset \mathbb{R}^D,\ D = 2, 3$. This field evolves in time $t \in [0, +\infty)$. Our problem depends on a set of parameters $\mathbf{p} = p_1, p_2, \ldots, p_n, \mathbf{p} \in \Omega_p \subset \mathbb{R}^n$.

It is assumed that a design of experiment makes it possible to obtain the evolution of the field T, in space and time, for several combinations of parameters $\mathbf{p}$. Our solution is then written in a general form $T(\mathbf{x}, t, \mathbf{p})$.

The representation of this solution, specifically according to the parametric dimension, is discrete. Artificial intelligence plays the role of interpolating (or

extrapolating) from the set of parameters already considered in the training stage.

In those approaches we have developed so far, we used a non-intrusive dimensionality reduction that expresses the solution from a finite sum of products of functions. For this purpose, we rely on the singular value decomposition. In order to apply this singular value decomposition approach, we need to operate on a discrete representation of the field T. In its classical form, the singular value decomposition decomposes the field into sums of tensor products of two discretized functions. A simple choice is to consider space and time on the one hand, and parameter space on the other. The reader can refer to [**?**] to see an example of the application of this approach.

The continuous form reads:

$$T(\mathbf{x}, t, \mathbf{p}) = \sum_{k=1}^{\infty} F^k(\mathbf{x}, t) H^k(\mathbf{p}). \tag{1}$$

This form corresponds to a discrete form which could be written with the index notation as

$$\mathbb{T}_{ij} = \sum_{k=1}^{\infty} \mathbf{F}_i^k \mathbf{H}_j^k, \tag{2}$$

where the subscripts i and j, refer here to the degrees of freedom in space, time and parameter dimensions, respectively.

The previous form can be rewritten in the tensor form

$$\mathbb{T} = \sum_{k=1}^{\infty} \mathbf{F}^k \otimes \mathbf{H}^k. \tag{3}$$

The determination of this form can be made in a direct way, by using a classical calculation based on the eigenvalue decomposition. However, in what follows, we use an iteration procedure, easily generalizable later to more dimensions, the so-called high-order singular value decomposition.

To find the series $(\mathbf{F}^1, \mathbf{H}^1), (\mathbf{F}^2, \mathbf{H}^2), \ldots$ we assume that the solution at iteration $k-1$ is known and given by

$$\tilde{\mathbb{T}} = \sum_{m=1}^{k-1} \mathbf{F}^m \otimes \mathbf{H}^m, \tag{4}$$

where $\tilde{\mathbb{T}}$ represents the field discrete approximation.

The difference between the initial discrete field and the approximated one is noted by $\mathbb{T}' = \mathbb{T} - \tilde{\mathbb{T}}$, which represents the approximation residual. The associated iteration algorithm solves:

$$\mathbf{F}_i^k = \frac{\sum_j \mathbb{T}'_{ij} \mathbf{H}_j^k}{\sum_j (\mathbf{H}_j^k)^2}, \tag{5}$$

$$\mathbf{H}_j^k = \frac{\sum_i \mathbb{T}'_{ij} \mathbf{F}_i^k}{\sum_i (\mathbf{F}_i^k)^2}, \tag{6}$$

until the convergence (fixed point) is reached.

The enrichment stops when the norm of the residual $\mathbb{T}'$ becomes lower than a tolerance criterion, fixed by the user. Here we assume that the enrichment process stops after K couples have been computed.

It follows that vectors $\mathbf{H}^k$ contains the parameter weights at each considered choice of parameter $\mathbf{p}$, $\mathbf{p}^j$. Thus, the component $\mathbf{H}_j^k$, $k = 1, \ldots, K$, is related to $\mathbf{p}^j = (p_1^j, \ldots, p_n^j)$.

Thus, one is tempted to train, from the available data couples $(\mathbf{p}^j, \mathbf{H}_j^k)$, an AI-based regression to evaluate the scalar H^k, $\forall k$, for any other value of $\mathbf{p}_{\text{new}}$, noted by $H^k(\mathbf{p}_{\text{new}})$, from which the reconstructed espace-time solution $\mathbf{T}_{xt}(\mathbf{p}_{\text{new}})$ reads

$$\mathbf{T}_{xt}(\mathbf{p}_{\text{new}}) = \sum_{k=1}^{K} \mathbf{F}^k \; H^k(\mathbf{p}_{\text{new}}). \tag{7}$$

2.2 Separating space and time

The approach that we have just presented fails to address problems with too many degrees of freedom in space and time. It is therefore useful to separate the temporal dimension from the spatial one (see [?] for an example).

The simplest option consists of performing a singular value decomposition in space and time for each solution associated to the parameters choice $\mathbf{p}^h = (p_1^h, p_2^h, \ldots)$, $h = 1, \ldots, H$. This SVD allows us to write

$$T(\mathbf{x}, t; \mathbf{p}^h) = \sum_k \; {}^hF^k(\mathbf{x}) \; {}^hG^k(t), \tag{8}$$

whose discrete form reads

$${}^h\mathbb{T}_{ij} = \sum_k \; {}^h\mathbf{F}_i^k \; {}^h\mathbf{G}_j^k, \tag{9}$$

and, in tensor form

$${}^h\mathbb{T} = \sum_k \; {}^h\mathbf{F}^k \otimes {}^h\mathbf{G}^k. \tag{10}$$

This expression does not allow us to build a response surface on the parametric space. To this end, we must express our different functions in a common approximation basis. To avoid redundancies between the different functions ${}^h\mathbf{F}^k$ and ${}^h\mathbf{G}^k$, obtained during the performed simulations for different parameter choices, and in order to guarantee the orthogonality of the basis, a proper orthogonal decomposition is performed.

Let $\mathbb{Q}$ be the matrix composed by the functions ${}^h\mathbf{F}^k$ for different parameters choices $\mathbf{p}^h$

$$\mathbb{Q} = [{}^1\mathbf{F}^1, {}^1\mathbf{F}^2, \ldots, {}^1\mathbf{F}^K, {}^2\mathbf{F}^1, {}^2\mathbf{F}^2, \ldots, {}^2\mathbf{F}^K, \ldots, {}^H\mathbf{F}^K]. \tag{11}$$

The resulting orthonormal eigenvectors are noted as $\mathbf{B}_1, \mathbf{B}_2, \ldots$.

By selecting the r eigenvectors associated with the r highest decomposition eigenvalues, the space approximation basis reads

$$\mathbb{B} = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_r]. \tag{12}$$

To express the basis obtained for a set of parameters h into the global basis (12), we define the coordinates matrix ${}^h\beta$ enabling the expression of ${}^h\mathbb{F} = [{}^h\mathbf{F}^1, \ldots, {}^h\mathbf{F}^K]$ into the common basis (12), according to

$$\mathbb{B}\ {}^h\beta = {}^h\mathbb{F}, \tag{13}$$

whose solution results from

$${}^h\beta = (\mathbb{B}^T\mathbb{B})^{-1}(\mathbb{B}^T\ {}^h\mathbb{F}). \tag{14}$$

The same rationale applies on the time vectors, leading to

$$\mathbb{C}\ {}^h\gamma = {}^h\mathbb{G}. \tag{15}$$

Thus, finally, the approximation reads

$${}^h\mathbb{T} = {}^h\mathbb{F}\ ({}^h\mathbb{G})^T = \mathbb{B}\ {}^h\beta({}^h\gamma)^T\ \mathbb{C}^T, \tag{16}$$

or, by defining the new matrix ${}^h\alpha = {}^h\beta({}^h\gamma)^T$, it results

$${}^h\mathbb{T} = \mathbb{B}\ {}^h\alpha\ \mathbb{C}^T. \tag{17}$$

Artificial intelligence intervenes here to obtain each component of the matrix α, $\alpha_{ij}(p_1, p_2, \ldots, p_n)$ from the existing knowledge: ${}^h\alpha_{ij}(p_1^h, p_2^h, \ldots, p_n^h)$, $h = 1, \ldots, H$.

The major drawback of such an approach is that the α coordinate matrix is not diagonal. This leads to a very large number of α_{ij} values involved in the training process. In addition, the numerous projections into the common truncated POD approximation basis introduce an additional error.

2.3 The proposed approach: A space, time and parameter separated representation

In this section we propose an approach that combines the advantages of the two procedures just described.

This approach relies on a high-order singular value decomposition involving three functions:

$$T(\mathbf{x}, t, \mathbf{p}) = \sum_{k=1}^{\infty} F^k(\mathbf{x}) G^k(t) H^k(\mathbf{p}), \tag{18}$$

whose discrete form reads

$$\mathbb{T}_{ijh} = \sum_{k=1}^{\infty} \mathbf{F}_i^k \mathbf{G}_j^k \mathbf{H}_h^k. \tag{19}$$

Following the rationale previously introduced, the approximation is obtained by successive enrichments (until obtaining the desired accuracy at $k = K$) and at each enrichment step k iterating until convergence, that is, until attaining the fixed point, according to

$$\mathbf{F}_i^k = \frac{\sum_{j,h} \mathbb{T}'_{ijh} \mathbf{G}_j^k \mathbf{H}_h^k}{\sum_j (\mathbf{G}_j^k)^2 \sum_h (\mathbf{H}_h^k)^2}, \tag{20}$$

$$\mathbf{G}_j^k = \frac{\sum_{i,h} \mathbb{T}'_{ijh} \mathbf{F}_i^k \mathbf{H}_h^k}{\sum_i (\mathbf{F}_i^k)^2 \sum_h (\mathbf{H}_h^k)^2}, \tag{21}$$

$$\mathbf{H}_h^k = \frac{\sum_{i,j} \mathbb{T}'_{ijh} \mathbf{F}_i^k \mathbf{G}_j^k}{\sum_i (\mathbf{F}_i^k)^2 \sum_j (\mathbf{G}_j^k)^2}. \tag{22}$$

Using the same rationale previously described, from the couples $(\mathbf{p}^h, \mathbf{H}_h^k)$, $k = 1, \ldots, K$, a regression is constructed to infer the scalars $H^k(\mathbf{p}_{\text{new}})$, $\forall k$, related to the parameters choice $\mathbf{p}_{\text{new}}$.

The reconstructed solution reads

$$\mathbb{T}(\mathbf{p}_{\text{new}}) = \sum_{k=1}^{K} (\mathbf{F}^k \otimes \mathbf{G}^k) H^k(\mathbf{p}_{\text{new}}). \tag{23}$$

The main steps of this methodology are summarized in figure 1

Fig. 1 Summary of the methodology
`shema_bloc-eps-converted-to.pdf`

3 Case study

The problem here consists in the metal cooling that fills a mould during the casting process. The mould cavity is created using tool steel and endowed with cooling channels. The metal used to fill the cavity is an aluminium-silicon alloy.

We denote by Ω_1 the domain filled by the metal and by Ω_2 the mould, being Γ the interface between the metal and the mould. Γ_2 represents the interface between the mould and the surrounding environment occupied by the air.

In the mould there are five cooling channels where the cooling liquid circulates. The interfaces between the mould and the cooling channels are denotes $C_i, i = 1, \ldots, 5$.

The thermal properties including the metal (with subscript 1) and the mould (with subscript 2) are given below (all quantities are expressed in the international units system):

- The convection coefficient on Γ is denoted $h_{12} = 500$ for the external boundary and 300 for the internal one.
- The convection coefficient on Γ_2 is denoted $h_{\text{air}} = 20$.
- The convection coefficient between the mould and the cooling liquid on C_i, $\forall i$, is denoted $h_c = 10^4$.
- The product of the density by the heat capacity for the part is $\rho_1 C_{p_1} = 5.4\ 10^6$.
- The product of the density by the heat capacity for the mould is $\rho_2 C_{p_2} = 1.5\ 10^6$.
- The conductivity of the metal is $\lambda_1 = 70$.
- The conductivity of the mould is $\lambda_2 = 40$.
- The air temperature outside the mould is $T_{\text{air}} = 20$.

The system of equations to solve during the time interval $[0, t_{max} = 300]$ is given by

$$\rho_1 C_{p_1} \frac{\partial T_1}{\partial t} = -\nabla.\mathbf{q}_1, \qquad \mathbf{q}_1 = -\lambda_1 \nabla T_1, \tag{24}$$

for $(\mathbf{x}, t) \in \Omega_1 \backslash \Gamma \times (0, t_{\max}]$,

$$\rho_2 C_{p_2} \frac{\partial T_2}{\partial t} = -\nabla.\mathbf{q}_2, \qquad \mathbf{q}_2 = -\lambda_2 \nabla T_2, \tag{25}$$

for $(\mathbf{x}, t) \in \Omega_2 \backslash (\Gamma_2 \cup C_1 \ldots \cup C_5) \times (0, t_{\max}]$.

These equations are subjected to the boundary conditions

$$\mathbf{q}_1.\mathbf{n} = h_{12}(T_{1\Gamma^-} - T_{2\Gamma^+}) \qquad \text{on } \Gamma^-, \tag{26}$$

$$\mathbf{q}_2.\mathbf{n} = h_{12}(T_{2\Gamma^+} - T_{1\Gamma^-}) \qquad \text{on } \Gamma^+, \tag{27}$$

$$\mathbf{q}_2.\mathbf{n} = h_{air}(T_{2\Gamma_2^-} - T_{\text{air}}) \qquad \text{on } \Gamma_2, \tag{28}$$

$$\mathbf{q}_2.\mathbf{n} = h_c(T_{2C_i^-} - p_i) \qquad \text{on } C_i, \tag{29}$$

where p_i refers to the temperature of the fluid circulating inside the channels and the superscripts Γ^+ and Γ^- the two sides of the interface.
The variational formulation for the problem on Ω_1 with a test field Ψ^* writes

$$\int_{\Omega_1} \Psi^* \rho_1 C_{p_1} \frac{\partial T_1}{\partial t} d\Omega_1 = + \int_{\Omega_1} \nabla\Psi^* \mathbf{q}_1 d\Omega_1 - \int_{\Gamma} \Psi^* \mathbf{q}_1.\mathbf{n}\, d\Gamma \tag{30}$$

$$= - \int_{\Omega_1} \lambda_1 \nabla\Psi^* \nabla T d\Omega_1 - \int_{\Gamma} \Psi^* h_{12}(T_{1\Gamma^-} - T_{2\Gamma^+})\, d\Gamma \tag{31}$$

This can be rewritten as

$$\int_{\Omega_1} \Psi^* \rho_1 C_{p_1} \frac{\partial T_1}{\partial t} d\Omega_1 + \int_{\Omega_1} \lambda_1 \nabla\Psi^* \nabla T_1 d\Omega_1 + \int_{\Gamma^-} h_{12} \Psi^* T_{1\Gamma^-}\, d\Gamma^- - \int_{\Gamma^+} h_{12} \Psi^* T_{2\Gamma^+}\, d\Gamma^+ = 0 \tag{32}$$

By skipping the details of the integration using the Galerkin approach with peace-wise linear functions the discrete system writes after simplification of the test field

$$\mathbb{M}_1 \dot{\mathbf{T}}_1 + \mathbb{K}_1 \mathbf{T}_1 + \mathbb{D}_1 \mathbf{T}_1 - \mathbb{P}_1 \mathbf{T}_2 = 0 \tag{33}$$

In equations (32) and (33) we have kept the same order of the different contributions so that the reader can make the correspondence between the different terms.
A similar approach for the domain Ω_2 gives the following system

$$\mathbb{M}_2 \dot{\mathbf{T}}_2 + \mathbb{K}_2 \mathbf{T}_2 + (\mathbb{D}_2 + \mathbb{E}_2) \mathbf{T}_2 - \mathbb{P}_2 \mathbf{T}_1 = \mathbf{J}_2, \tag{34}$$

where the new terms $\mathbb{E}_2$ and $\mathbf{J}_2$ account for the contributions of the convective heat transfer with air and with coolant.
The coupled system to be solved writes finally

$$\begin{pmatrix} \mathbb{M}_1 & 0 \\ 0 & \mathbb{M}_2 \end{pmatrix} \begin{pmatrix} \dot{\mathbf{T}}_1 \\ \dot{\mathbf{T}}_2 \end{pmatrix} + \begin{pmatrix} \mathbb{K}_1 + \mathbb{D}_1 & -\mathbb{P}_1 \\ -\mathbb{P}_2 & \mathbb{K}_2 + \mathbb{D}_2 + \mathbb{E}_2 \end{pmatrix} \begin{pmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{J}_2 \end{pmatrix} \tag{35}$$

In order to take into account the phase change latent heat for the metal, we use effective value of $(\rho_1 C_{p_1})_{\text{eff}}$:

$$(\rho_1 C_{p_1})_{\text{eff}} = \rho_1 C_{p_1} + A \frac{\exp(-\frac{(T-T_\varphi)^2}{\delta^2})}{\delta\sqrt{\pi}}. \tag{36}$$

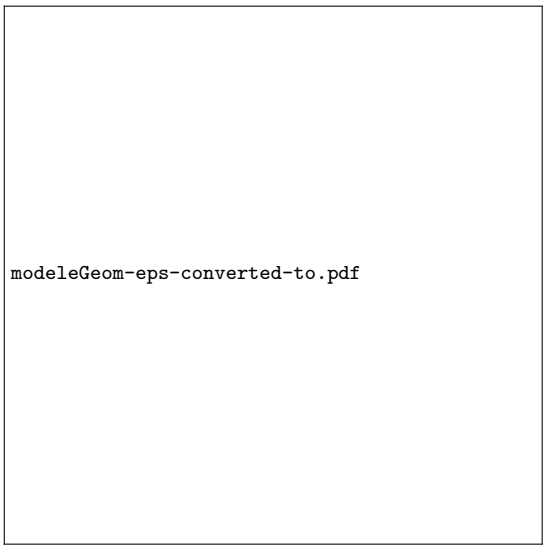

Fig. 2 Model geometry (meter unit for dimensions)

The introduction of this relation to model latent heat effects comes from [**?**] and [**?**]. The idea consists to replace the constant value of $\rho_1 C_{p_1}$ by an effective value that is augmented by a new curve in the form of a smoothed Dirac function. The area under this curve represents the latent heat and controlled by the parameter A. δ is the phase change temperature range. It characterizes the global width of the curve. It is homogeneous to a temperature. T_φ is the temperature around which the phase change occur.
The numerical values considered in our study are $A = 3.3.10^8$, $\delta = 1.1$, $T_\varphi = 380$.
The simulation is done with an implicit approach in time and with a time step equal to 1 second in a time interval of 300 seconds.
The five variable parameters in this study are the temperatures of the fluid circulating in the five cooling channels. They will be denoted by $\mathbf{p} = p_1, \ldots, p_5$.
The domain of this study is presented in figure 2. The casted part has a width equal to 0.1 and a height equal to 0.06. The external dimensions of the mould are 0.16×0.12. The computational mesh is represented in figure 3.
Figure 4 shows the thermal field on the mould and metal assembly, when our five parameters are uniformly set to 20. However, to better identify the distribution of temperature in each region, an exploded representation is given in figure 5. The initial conditions are such that temperature is equal to 500 degrees Celsius for the metal and 100 degrees Celsius for the mould. The illustrations of figures are after $300s$ cooling time.
Another illustration is shown in figures 6 and 7 where we deliberately unbalanced different temperatures in the cooling channels to see the consequence on the thermal distribution, in both, the part and in the mould.

Fig. 3 Meshed computational domain

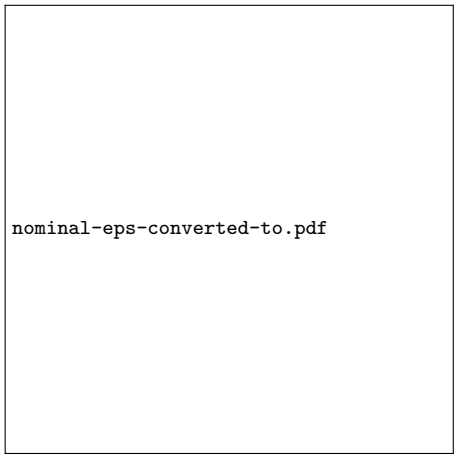

Fig. 4 Temperature distribution in degrees Celsius with homogeneous parameters $p_1, \ldots, p_5$

4 Parametric surrogate

A design of experiments was generated with 200 simulations. Each of the simulations start from the initial temperature field described above, and the temperature evolution is calculated during 300 seconds.

From these 200 simulations, 150 were used in the model training, 30 were used for testing, and the remaining 20 will be used for validation purposes as discussed later.

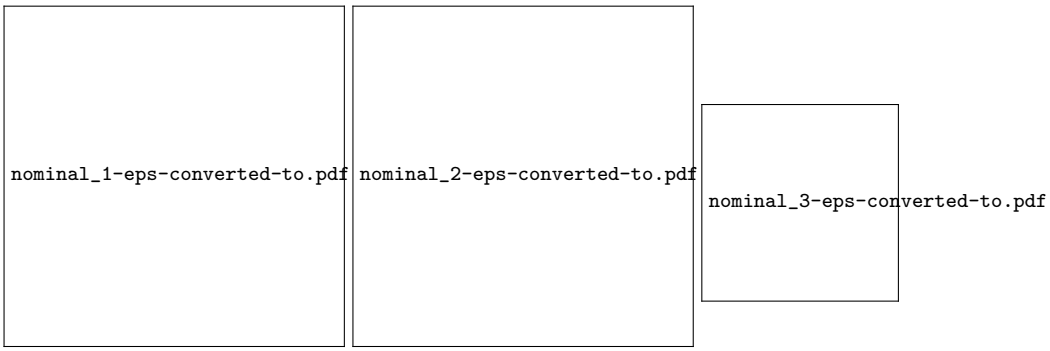

Fig. 5 Temperature distribution in degrees Celsius with homogeneous parameters $p_1, \ldots, p_5$ in the different components

Fig. 6 Temperature distribution in degrees Celsius with heterogeneous parameters $p_1, \ldots, p_5$

These 200 configurations, consisting of different parameters choices $\mathbf{p}^h$, were generated using the Latin hypercube sampling. The interval in which the different parameters take their values is $[0, 100]$.

Even if, during the simulations, we are interested in the thermal field in the global domain, part and mould, during the machine learning phase, we will focus only on the domain defined by the cavity because indeed our interest focus in controlling the evolution of the temperature in the part, which can affect its properties in service, from the level of residual stresses.

The temperature field in the domain defined by the cavity was then stored for the 300 iterations and for the different parameters, in a three-dimensional matrix description $\mathbb{T}$. The application of the singular value decomposition on this matrix leads to different modes in space, in time, and in the parameters space.

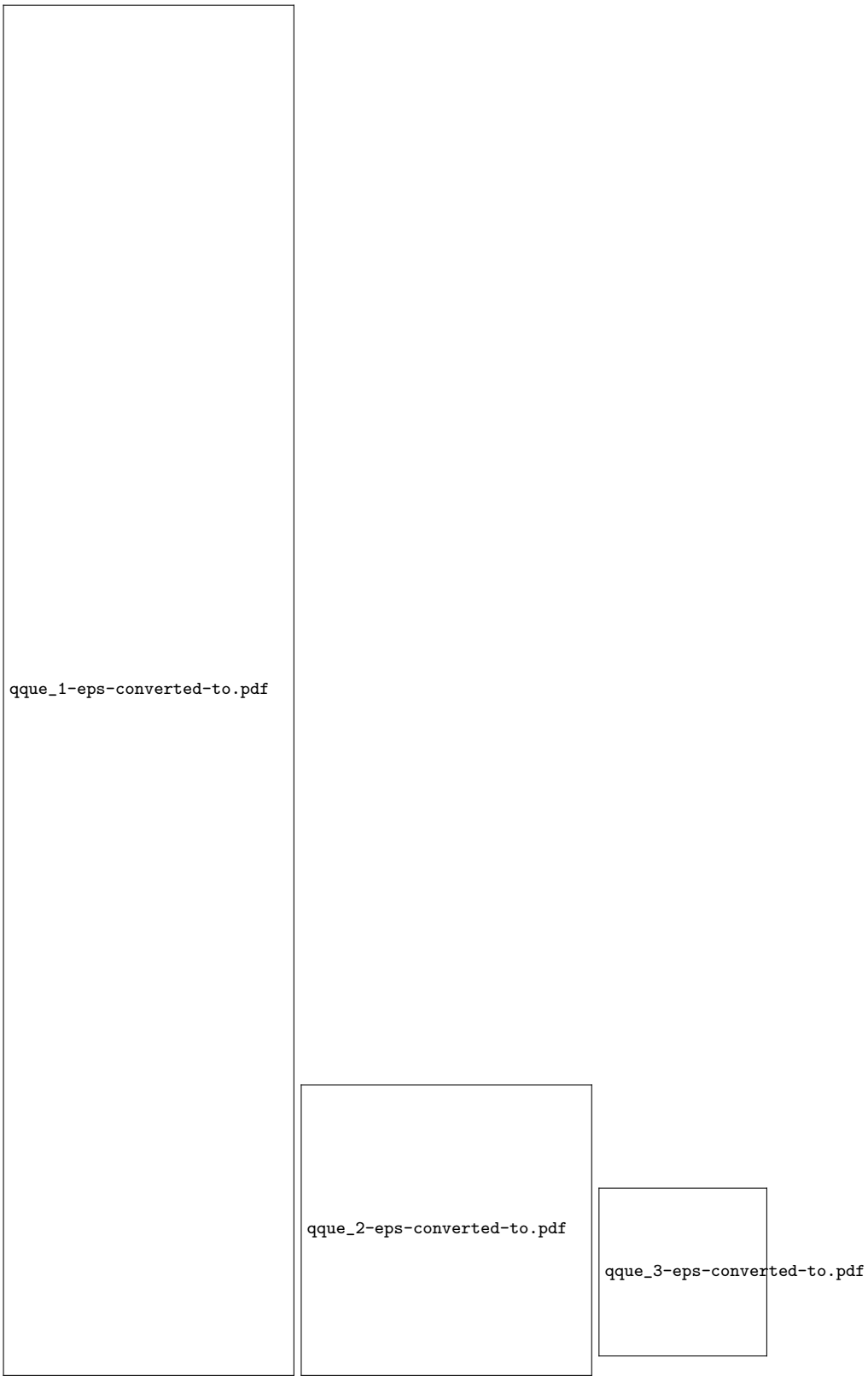

Fig. 7 Temperature distribution in degrees Celsius with heterogeneous parameters $p_1, \ldots, p_5$ in the different components

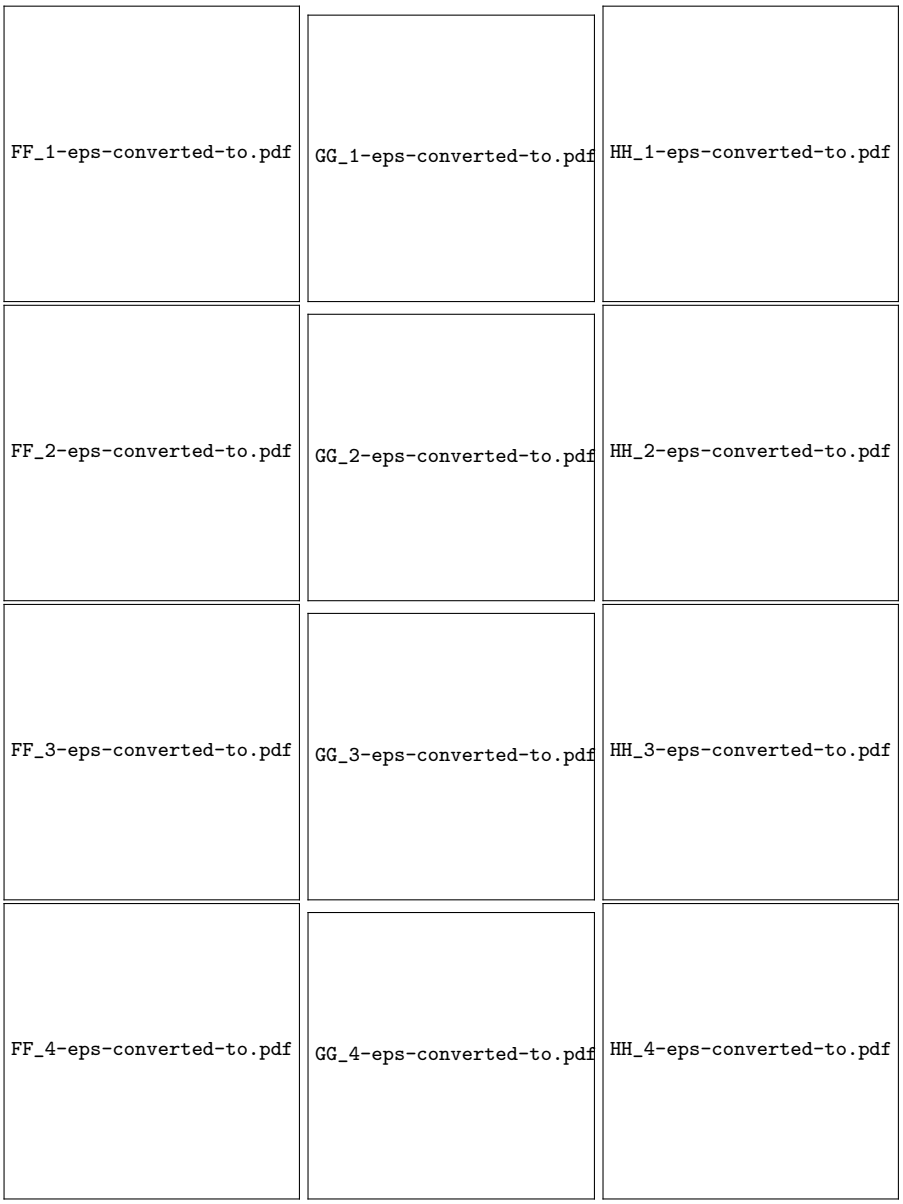

Fig. 8 Modal decomposition (space-time-parameters) of the discrete temperature field $\mathbb{T}$: **F** functions (left), **G** functions (center) with time expressed in seconds and **H** functions (right).

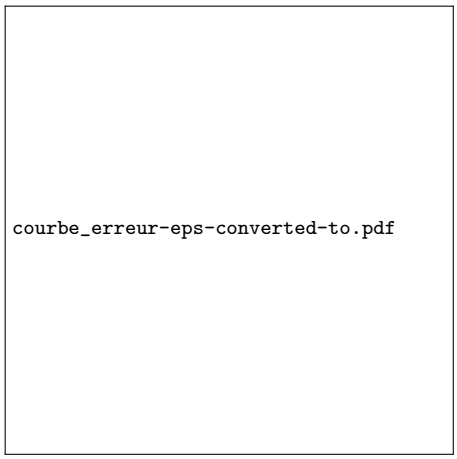

Fig. 9 Error versus number of modes considered in the modal decomposition

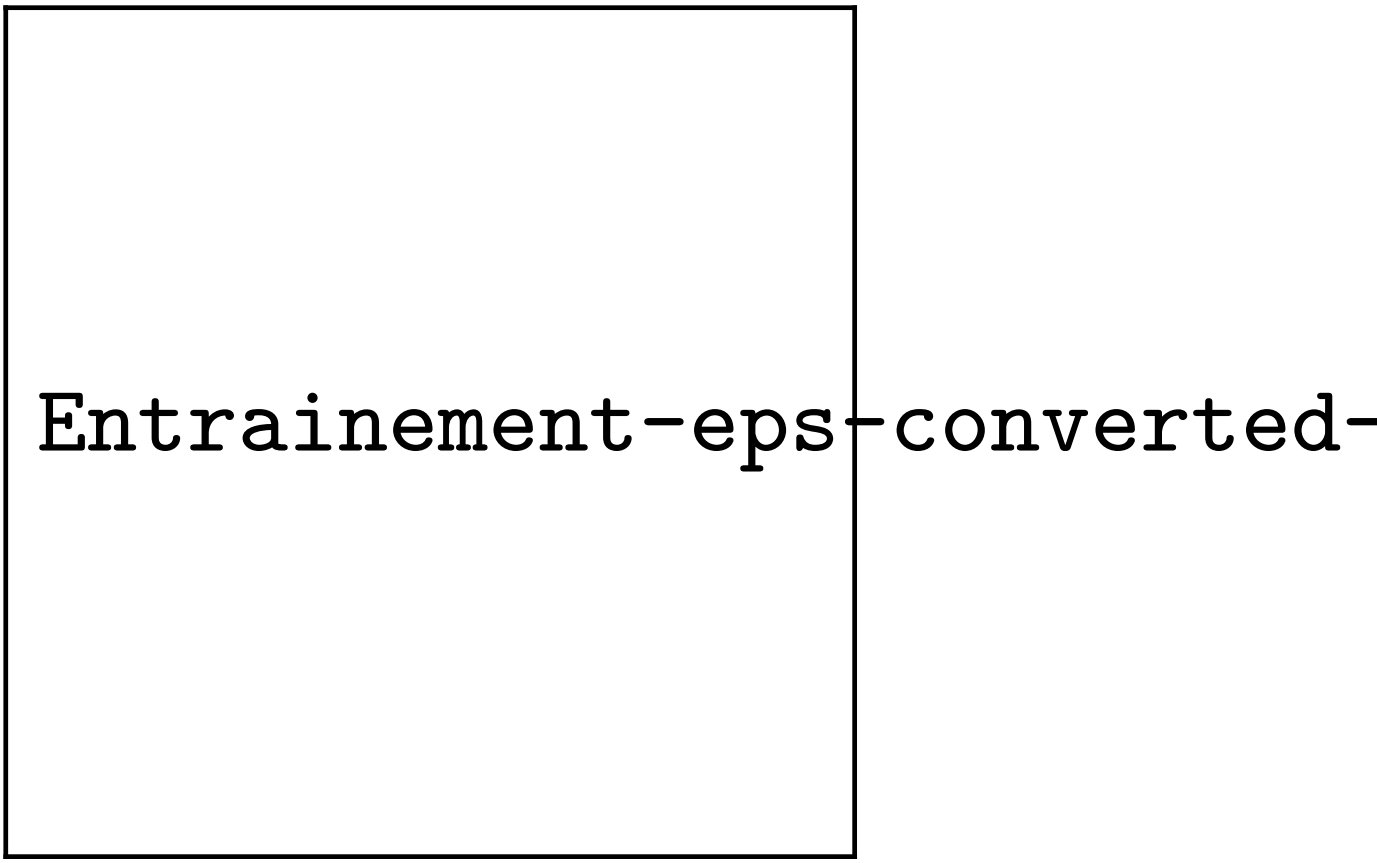

Fig. 10 Polynomial training

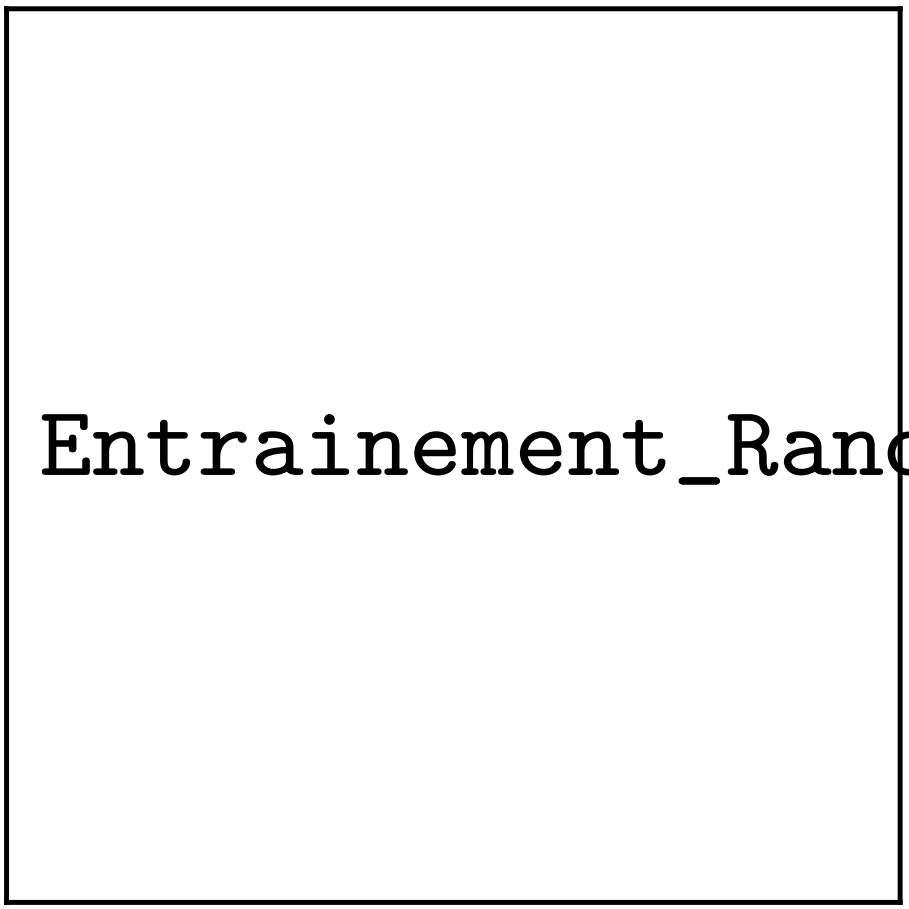

Fig. 11 Random Forest training

Figure 8 shows the first four modes of the decomposition. The left column depicts the modes in space. The central column presents the time modes. Finally, the right column presents the parametric modes. In the figures of the right column, the order of the points is completely arbitrary. In order to simplify the visual representation, we represented only 10% of the points in the design of experiments, that is 18 over the 180 (training and test sets). On the x-axis each point represents a parameter data-point $\mathbf{p}^h$ (the five temperatures of the cooling fluid circulating in the five channels) and on the y-axis the associated value of function $\mathbf{H}_h^k$.

The relative norm of the residue represented in Fig. 9 proves that the first mode is the most relevant, and that 40 allows reducing the error by three orders of magnitude.

Fig. 12 First validation case. Temperature in degrees Celsius at 100 and 300 seconds and associated errors in degrees Celsius

Two training set one based on random forest approach and another based on polynomial interpolation were performed on all the H-functions, the training set consisting of the couples $(\mathbf{p}^h, \mathbf{H}_h^k)$, $\forall h,\ \forall k$.

The graphs in Figs. 10 and 11 represent the performance of the predictions for the first 12 functions $H^k, k = 1, \ldots, 12$. For each of the functions we represent the inferred value versus the value given by the simulation (considered as a reference value) and we indicate at the top of each image, the performance of the training.

These performances are quantified from the root mean square error (RMSE) and the R2 coefficient. The first line deals with he set used for training and the second line is for the set used for the test. By comparing both approaches, it turns out that in this case the polynomial approach performs better. An approach based on neural networks (not presented here) provides results that are very close to those obtained from the polynomial regression.

Fig. 13 Second validation case. Temperature in degrees Celsius at 100 and 300 seconds and associated errors in degrees Celsius

In order to quantify the performances of our method on the sets of parameters used for the validation, we will directly compare the thermal fields with the reference simulations. Indeed, these simulations used for the validation did not intervene in the singular value decomposition. We will use the modal basis extracted from the SVD built on the training simulations combined with the estimation of the H-functions based on AI-based regressions.

The comparison made directly on the thermal field on all the 20 simulations used for the validation, gives deviations which do not exceed 0.4 degree in the temperature values. Figs. 12 and 13 concern two arbitrary combinations of parameters, and depict the temperature field at 100 and 300 seconds. The thermal fields presented here are the ones obtained by reconstruction from the use of the surrogate. The bottom figures represent the error with respect to the reference solution. These errors remain relatively small and are acceptable

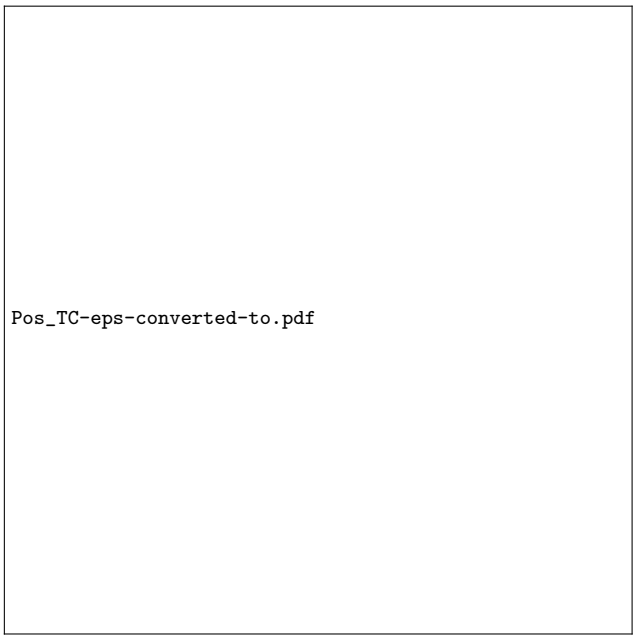

Fig. 14 Thermocouple location

for a prediction of the thermo-mechanical properties induced by the thermal field.

5 Construction of the Hybrid Twin

Our objective in this part is to set up a hybrid twin of the casting process. This twin shall be able to learn the difference between numerical simulations and experimental observations. As we have not yet developed experiments for the case presented above, we will generate the experimental data synthetically.

We use the numerical model previously developed as the basis, while increasing the conductivities by 10% and reducing the convection coefficients by 10%. From now on, we note by experimental results the numerical data generated under these conditions.

The experimental observation is normally limited to a set of thermocouples. In our case this set is presented in Fig. 14. The indexes of the eight nodes where thermocouples are placed are noted by $i_1, i_2, \ldots, i_8$.

For a set of parameter $\mathbf{p}^h$ we denote by ${}^h\mathbb{T}^{\mathrm{Exp}}_{i'j}, i' = i_1, \ldots, i_8, j = 1, \ldots, 300$ the matrix containing the experimental temperature evolution at the eight thermocouples for the 300 simulation time steps. We also denote by ${}^h\mathbb{T}^{\mathrm{Num}}_{i'j}$ the matrix containing the simulated temperature evolution at the same nodes where the thermocouples are located.

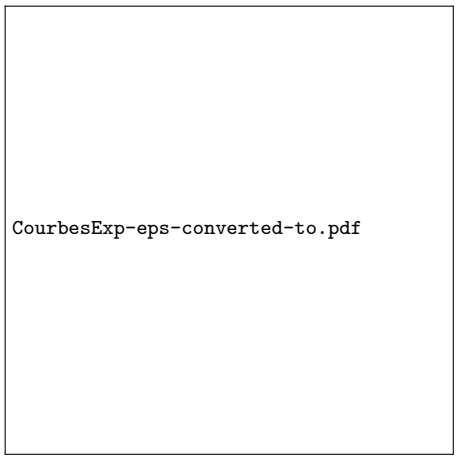

Fig. 15 Experimental temperature (in degrees Celsius) during time (in seconds) at the thermocouple location

Our aim is to establish a correction model based on the tensor decomposition of the numerical simulation ${}^{h}\mathbb{T}_{ij}^{\text{Num}} = \sum_{k} \mathbf{F}_i^k H_h^k \mathbf{G}_j^k$.

Let us denote the difference between experiments and simulation (or model's ignorance), at each thermocouple location, by

$$ {}^{h}\bar{\mathbb{T}}_{i'j} = {}^{h}\,\mathbb{T}_{i'j}^{\text{Num}} - {}^{h}\,\mathbb{T}_{i'j}^{\text{Exp}}. \tag{37}$$

5.1 Ignorance model learnt through a minimization procedure

In order to express this difference in the same space-time basis $(\mathbf{F}_i^k, \mathbf{G}_i^k)$, the following minimization problem should be solved:

$$ \bar{H}_h^k = \underset{\mathcal{H}^k}{\text{argmin}} \left({}^{h}\bar{\mathbb{T}}_{i'j} - \sum_{k} \mathbf{F}_{i'}^k \; \mathcal{H}^k \; \mathbf{G}_j^k \right). \tag{38}$$

It is important to mention at this point that in this minimization we constrain the difference (ignorance) to be written using the functions defined in the space-time description of the numerical simulation. This can sometimes be slightly restrictive. Later we will propose a less restrictive approach later.

In the present case order to alleviate the minimization procedure, we limit the time period in which the minimization applies to the interval $j' = 200, \ldots, 300$.

In Fig. 15 the time evolution of the temperature at the eight thermocouples locations are illustrated for the choice of the parameters indicated in the figure. In Fig. 16 the deviation (ignorance) between the numerical predictions and the experimental observation is shown. In this figure the dashed lines represent

Fig. 16 Deviation and deviation model predictions (values of temperature in degrees Celsius and time in seconds)

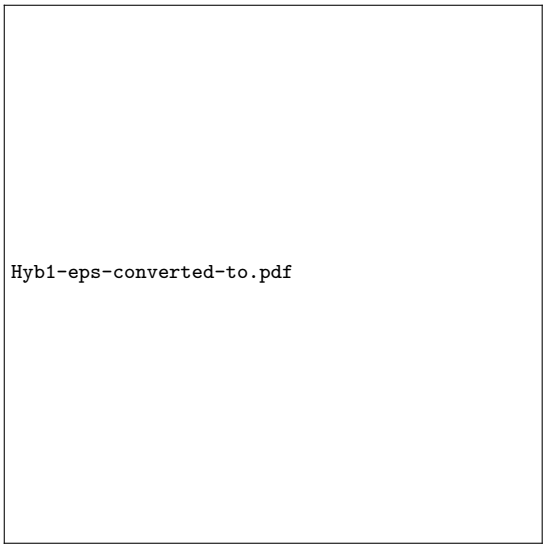

Fig. 17 Numerical result at $t = 300s$ (values in degrees Celsius)

the results of the reconstructed model by using the optimisation procedure described above.

The numerical prediction of our model is provided in figure 17. The reconstructed ignorance is illustrated in figure 18(left). All these figures are produced with the final time ($t = 300$) and with the set of parameters indicated in figure 15.

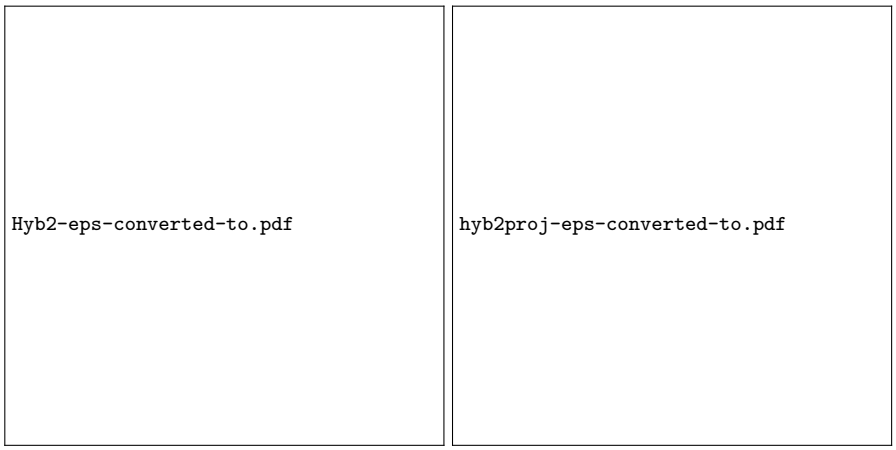

Fig. 18 Ignorance model solution obtained by using the minimization (left) and the projection (right) procedures (values is in degrees Celsius)

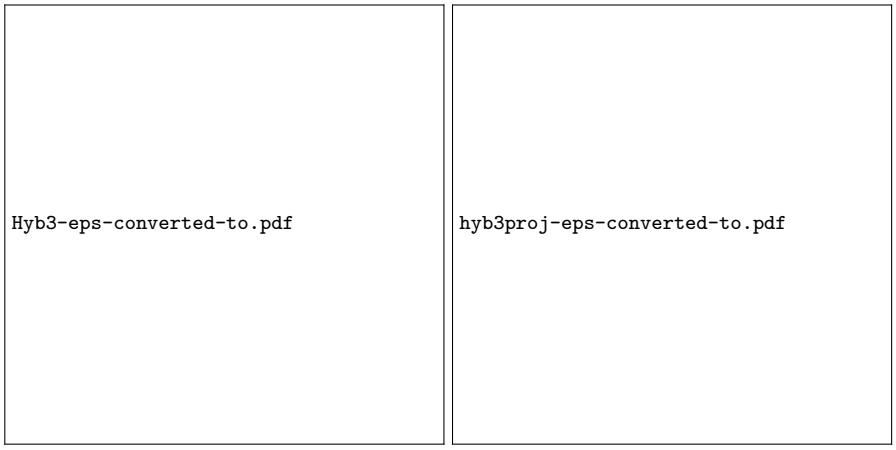

Fig. 19 Superposition of the numerical prediction and the ignorance based on the minimization (left) and the projection (right) procedure (values in degrees Celsius)

The predictions obtained by using the numerical model enriched with the one of the ignorance is represented in Fig. 19(left). Figure 20 shows the experimental temperature at the thermocouples location. Finally, Fig. 21(left) gives the global error of the hybrid twin model, where an impressive error reduction can be identified.

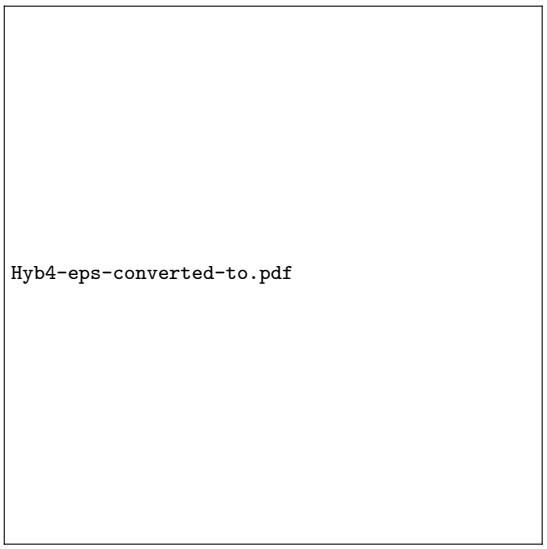

Fig. 20 Experimental measurements (values in degrees Celsius)

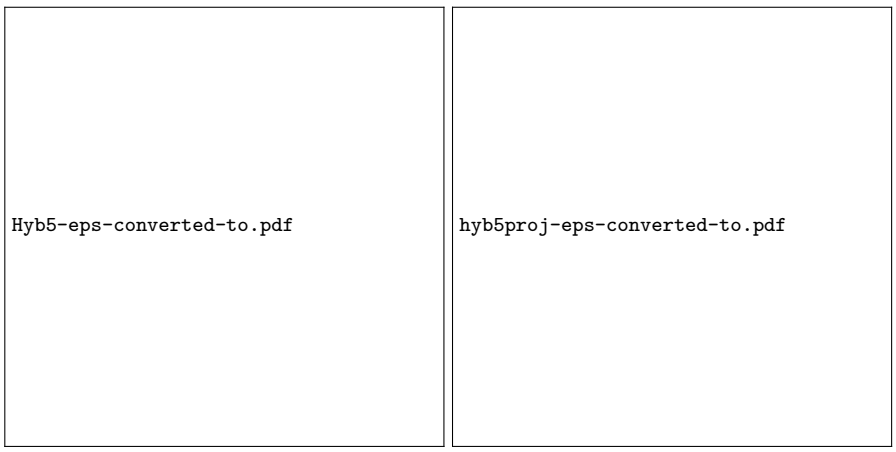

Fig. 21 Prediction error of the hybrid twin model with minimization (left) and with projection (right) (values in degrees Celsius)

5.2 Ignorance model learnt from a projection procedure

We propose here a more general procedure that alleviates some of the constraints of the previous procedure. Here we will use slightly different notations. The equation (19) is rewritten using the Khatri-Rao product ($\odot$) generalized for three matrices.

By defining the following matrices

$$\mathbb{F} = [\mathbf{F}^1, \mathbf{F}^2, \ldots],$$

$$\mathbb{G} = [\mathbf{G}^1, \mathbf{G}^2, \ldots],$$

$$\mathbb{H} = [\mathbf{H}^1, \mathbf{H}^2, \ldots],$$

Eq. (19) can be rewritten as

$$\mathbb{T} = \mathbb{F} \odot \mathbb{G} \odot \mathbb{H}. \tag{39}$$

The simulation matrix $\mathbb{T}$ has dimension $(N \times t \times d)$ and the size of $\mathbb{F}$ is $(N \times K)$ where N is the number of nodes involved in the cavity mesh, t the number of time steps, d the DoE size and K is the number of modes.

Concerning the ignorance matrix, with $n = 8$ thermocouples, the matrix size becomes $(n \times t \times d)$. This ignorance matrix reads

$$\bar{\mathbb{T}} = \bar{\mathbb{F}} \odot \bar{\mathbb{G}} \odot \bar{\mathbb{H}}, \tag{40}$$

where

$$\bar{\mathbb{F}} = [\bar{\mathbf{F}}^1, \bar{\mathbf{F}}^2, \ldots]_{(n \times K')},$$

$$\bar{\mathbb{G}} = [\bar{\mathbf{G}}^1, \bar{\mathbf{G}}^2, \ldots]_{(t \times K')},$$

$$\bar{\mathbb{H}} = [\bar{\mathbf{H}}^1, \bar{\mathbf{H}}^2, \ldots]_{(d \times K')}.$$

This matrix has been obtained from a new iterative SVD (involving K' modes) completely independent of the one that served to decompose the simulation solution.

The main idea consists in expressing this new decomposition using the space-time functions of the numerical decomposition. Let us denote by $\mathbb{F}'_{(n \times K)}$ the selection of the n rows of the matrix $\mathbb{F}$. The coordinates of the matrix $\bar{\mathbb{F}}$ into the basis $\mathbb{F}'$ define matrix $\mathbf{a}_{(K \times K')}$

$$\bar{\mathbb{F}} = \mathbb{F}'\mathbf{a}, \tag{41}$$

that defining an undetermined problem, its solution must be regularized. In order to preserve the sparsity this system is solved subjected to the L1-norm minimisation. Thus the obtained solution $\mathbf{a}$ selects naturally the more adequate functions of the numerical basis to express the ignorance.

Concerning the time basis, the coordinates of the matrix $\bar{\mathbb{G}}$ into the basis $\mathbb{G}$ results in matrix $\mathbf{b}_{(K \times K')}$

$$\bar{\mathbb{G}} = \mathbb{G}\mathbf{b}, \tag{42}$$

that being usually overdetermined, a classical minimization procedure performs well (but a L1 norm could be applied if the system becomes undetermined)

$$\mathbf{b} = [\mathbb{G}^T\mathbb{G}]^{-1}[\mathbb{G}^T\bar{\mathbb{G}}]. \tag{43}$$

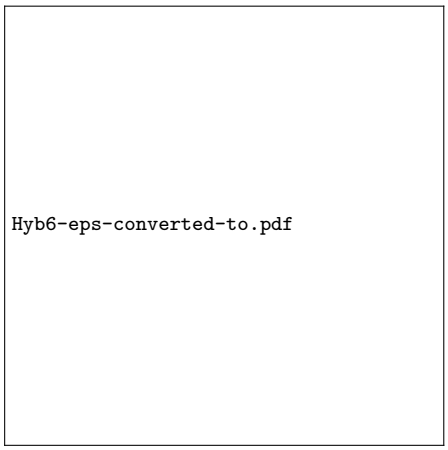

Fig. 22 Experimental temperature field

It is now possible the write the ignorance defined in Eq. (40) by using the space-time basis that comes from numerical simulation

$$\bar{\mathbb{T}}_{(n\times t\times d)} = (\mathbb{F}'\,\mathbf{a})_{(n\times K')} \odot (\mathbb{G}\,\mathbf{b})_{(t\times K')} \odot \bar{\mathbb{H}}_{(d\times K')}, \tag{44}$$

that can be then extended to the whole space domain by simply replacing $\mathbb{F}'$ by $\mathbb{F}$

$$\bar{\mathbb{T}}_{(N\times t\times d)} = (\mathbb{F}\,\mathbf{a})_{(N\times K')} \odot (\mathbb{G}\,\mathbf{b})_{(t\times K')} \odot \bar{\mathbb{H}}_{(d\times K')}. \tag{45}$$

In order to compare the performance of this projection based approach in relation to the minimization based approach described in the previous section, the new proposed procedure is applied to the case-study previously addressed.

The reconstructed ignorance is illustrated in Fig. 18(right). The superposition of the ignorance with the numerical model is depicted in Fig. 19(right). Finally figure 21(right) gives the global error of the hybrid twin model, proving its exceptional performance.

In the particular case of our so-called experimental solution that has been obtained numerically, temperature filed could be known everywhere in the computational domain, as illustrated in Fig. 22. Thus, the global error of the hybrid twin model can be obtained for both, minimization and projection procedures as depicted in Fig. 23.

The performance of the projection method are better in terms of error values but also in terms of error distribution over the domain.

Remark

In order to specify an order of magnitude on the resolution and storage cost we give a small illustration on the studied case. In our study we have a problem which contains N degrees of freedom in space (about 1500), t time steps (about

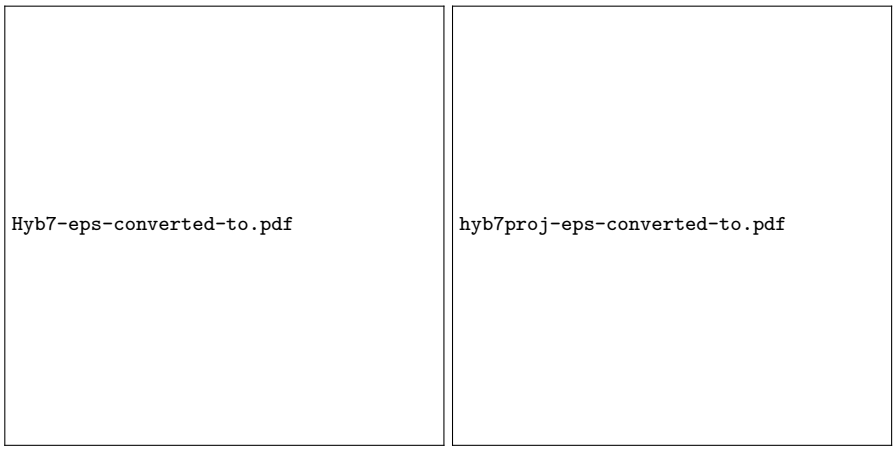

Fig. 23 Hybrid Twin error for both, the minimization (left) and the projection (right) procedures

300) and d combinations of parameters (about 200). The decomposition of the tensor $\mathbb{T}$ as written in equation (19) takes around 5 CPU-seconds for each fixed point iteration involving about 100 alternating resolution of equations (20), (21) and (22). If 50 enrichments are performed the entire decomposition takes about 250 CPU-seconds. In terms of memory storage we are always dealing with a tensor size containing $9 \cdot 10^7$ real values that represents 0.7 Gigabytes assuming a double precision of float number representation. In such situation if we imagine that one would use a more refined mesh which involves twice more degrees of freedom in the physical space representation ($2N$) thus the total used memory is multiplied by 2. It is the same for the CPU cost of the resolution of equations (20),(21),(22). In fact the CPU evolution here is linear and not quadratic because the latter system does not contain inversion but just a set of matrix product operations. However for the hybrid model as we have very little experimental information ($n = 8$ instead of N) the costs of calculation and storage are much reduced.

6 Conclusion

The casting twin addressed in the present paper was developed on a combination of a singular value decomposition strategy with machine learning-based regressions. This approach has been extended to establish a model of ignorance when experimental data is available. To our knowledge, the combination of singular value decomposition with machine learning-based regressions has very rarely been applied to processes in general and we have not found any work in the literature concerning the specific casting process. In most studies using artificial intelligence for processes, inputs and outputs are related to more macroscopic quantities. This new proposed methodology was applied

to a casting part where the different temperatures of the fluid circulating in the cooling channels were considered as variable parameters. The errors of the digital twin, as well as the hybrid twin, were evaluated at different instants of the cooling process and compared to a reference solution.
The error and performance of the parametric surrogates and the hybrid twin were convincing, proving the potential of the proposed approach. Less than one degree Celsius was noted for model accuracy. This remains largely within the tolerance interval in the temperature prediction of such a process.
The machine learning part convincingly showed the ability of the artificial intelligence models used to determine a response surface with completely satisfactory metrics. Both regressions tested (Random Forest and Polynomial) gave rise to RMSE errors less than 0.1 for training and testing sets associated to a determination coefficients generally higher than 0.8. The application framework of the strategy put in place within the framework of this work can be extended to any transient problem without being limited to shaping processes. This can be in particular the case of velocity field evolution in a transient flow, or for example the evolution of chemical concentration in a transient non-homogeneous problem.

Acknowledgements

This material is based upon work supported in part by the Army Research Laboratory and the Army Research Office under contract/grant number W911NF2210271.

This work has also been partially funded by the Spanish Ministry of Science and Innovation, AEI /10.13039/501100011033, through Grant number PID2020-113463RB-C31 and the Regional Government of Aragon and the European Social Fund, group T24-20R.

The support of ESI Group through the Chairs at ENSAM and Universidad de Zaragoza is also gratefully acknowledged.