



# Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks

Qiming Zhu<sup>1</sup> · Zeliang Liu<sup>2</sup> · Jinhui Yan<sup>1</sup>

Received: 1 August 2020 / Accepted: 16 November 2020 / Published online: 6 January 2021  
© Springer-Verlag GmbH Germany, part of Springer Nature 2021

## Abstract

The recent explosion of machine learning (ML) and artificial intelligence (AI) shows great potential in the breakthrough of metal additive manufacturing (AM) process modeling, which is an indispensable step to derive the process-structure-property relationship. However, the success of conventional machine learning tools in data science is primarily attributed to the unprecedented large amount of labeled data-sets (big data), which can be either obtained by experiments or first-principle simulations. Unfortunately, these labeled data-sets are expensive to obtain in AM due to the high expense of the AM experiments and prohibitive computational cost of high-fidelity simulations, hindering the direct applications of big-data based ML tools to metal AM problems. To fully exploit the power of machine learning for metal AM while alleviating the dependence on “big data”, we put forth a physics-informed neural network (PINN) framework that fuses both data and first physical principles, including conservation laws of momentum, mass, and energy, into the neural network to inform the learning processes. To the best knowledge of the authors, this is the first application of physics-informed deep learning to three dimensional AM processes modeling. Besides, we propose a hard-type approach for Dirichlet boundary conditions (BCs) based on a Heaviside function, which can not only exactly enforce the BCs but also accelerate the learning process. The PINN framework is applied to two representative metal manufacturing problems, including the 2018 NIST AM-Benchmark test series. We carefully assess the performance of the PINN model by comparing the predictions with available experimental data and high-fidelity simulation results, using finite element based variational multi-scale formulation method. The investigations show that the PINN, owed to the additional physical knowledge, can accurately predict the temperature and melt pool dynamics during metal AM processes with only a moderate amount of labeled data-sets. The foray of PINN to metal AM shows the great potential of physics-informed deep learning for broader applications to advanced manufacturing. All the data-sets and the PINN code will be made open-sourced in <https://yan.cee.illinois.edu/> once the paper is published.

**Keywords** CFD · Thermal multiphase flows · Additive manufacturing

## 1 Introduction

It has been widely believed that metal additive manufacturing (AM) can revolutionize mechanical, aerospace, and biomed-

ical industries owing to its superior capability to print metals with complex geometries directly from digital models without the design constraints of traditional manufacturing routes. The market of metal AM has been growing significantly for the past decade. However, compared with the total manufacturing market, the industrial adoption of metal AM has not reached its expected potential due to a lack of reliable process-structure-property relationships. For the past several years, predictive computational models, in conjunction with in-situ and ex-situ measurements and monitoring [1–3], have been playing an indispensable role in enhancing the understanding of the process-structure-property relationship in metal AM. Federal agencies have also conducted several benchmark experiments, such as NIST AM-bench [4] and

✉ Jinhui Yan  
yjh@illinois.edu

Zeliang Liu  
zeliang.liu@ansys.com

<sup>1</sup> Department of Civil and Environmental Engineering,  
University of Illinois at Urbana-Champaign, Champaign, IL,  
USA

<sup>2</sup> Livermore Software Technology, An ANSYS Company,  
Livermore, CA, USA

AFRL AM modeling challenge series [5], to facilitate the development of metal AM modeling tools.

Among various computational models at different scales and fidelity, thermal-fluid process simulation is not only an essential tool to understand the metal AM physics but also acts as a spearhead to derive the process-structure-property relationship. The metal AM process is intrinsically a multi-scale and multi-physics problem, involving rapid, complex, and coupled mass/flow/heat exchanges between gas, liquid, and solid phases, with large density ratios and complicated interfacial phenomenon. Current numerical simulation tools often employ mathematical models that couple Navier-Stokes equations and a heat transfer equation to capture the evolution of temperature and melt pool dynamics during manufacturing processes. For decades, the manufacturing community has been adopting computational methods that directly solve these mathematical models or their weak forms, based on spatial discretization (e.g., finite difference, finite volume, finite element, and mesh-free methods) and time-stepping.

The predictive capacities of these approaches have been significantly enhanced, thanks to the researchers' persistent efforts in numerical method development from the manufacturing and computational mechanics/mathematics communities. For example, Lawrence Livermore National Lab developed a thermal-fluid solver using the Arbitrary-Lagrangian Eulerian technique, which can simulate laser powder bed fusion (LPBF) processes at powder-scale [6–9]; Knapp et al. [10] and Mukherjee et al. [11,12] developed a coupled thermal-fluid model to simulate directed energy deposition (DED) and laser powder bed fusion (LPBF) processes. Lin et al. [13,14] developed a control-volume finite element approach to simulate directed energy deposition process. Lattice Boltzmann method has been used to model the metal powder melting and re-solidification in [15–17]; Zohdi group employed a discrete particle method to describe the selective laser sintering process [18–20]; Yan et al. developed a volume-of-fluid (VoF) based thermal-fluid solver to model multi-layer and multi-track LPBF process [21–25]; Panwisawas et al. also employed a VoF method by using OpenFOAM to analyze the inter-layer and inter-track void formation [26]; Li et al. developed a thermal-fluid model by combining level set method and Lagrangian particle tracking to investigate powder-gas interaction in LPBF processes [27]. CFD-ACE+, a code developed by ESI group, has been used to analyze the defects such as porosity, balling, and denudation in metal AM [28,29]; The last author of this paper [30] developed a gas–liquid–solid thermal flow model based on the level set method and residual-based variational multi-scale method to simulate laser spot melt pool flows. Li et al. used a mesh-free model based on material point method for selective laser beam melting processes [31]. Gan et al. developed a finite element method (FEM) based thermal-

fluid model and applied it to the NIST AM-Bench problems [32].

The core of these conventional approaches can be viewed as the process of using numerical approximations to solve PDEs without using labeled experimental/computational data. Despite the continued success and evolution, these methods require sophisticated mathematical treatments for spatiotemporal discretizations, coupling strategies, boundary conditions, and linear solvers to ensure stability, robustness, and efficiency. The application of these approaches to real additive manufacturing problems is prohibitively expensive and intricate. The high-fidelity simulations are typically executed in a parallel environment and consume massive high-performance computing (HPC) hours. Also, the performance of these approaches is often problem-dependent, necessitating numerical practitioners to have a deep understanding of not only the manufacturing problems but also the underlying mathematical techniques.

Machine learning (ML) and artificial intelligence (AI) have the potential to accelerate breakthroughs in thermal-fluid modeling for metal AM processes by harnessing data from sensors, experiments, and high-fidelity simulations. In general, ML focuses on algorithmic modeling of data and making predictions of labels based on observations, with emphasis on making accurate predictions for classification and regression tasks. Modern deep learning approaches have demonstrated tremendous successes in domains ranging from sentiment analysis to chemical predictions to material design [33–38]. The first reason for the major success of modern ML techniques, especially deep learning, is the availability of vast amounts of data (big-data). The second reason is that many technical burdens have been mitigated by advances in both hardware and software, including high-performance computers, graphics processing units (GPUs), fast large-scale optimization schemes, new optimality guarantees, and many user-friendly open-sourced packages, such as Tensorflow [39], PyTorch [40], Theano [41], and Caffe [42].

However, using deep learning for AM process modeling is still challenging. The primary challenge arises from the lack of large labeled data-sets since either experimental measurements or high-fidelity simulated data of AM processes are expensive to attain, rendering the big data-based ML/AI algorithms infeasible. The good news in scientific problems, however, is that there is highly condensed knowledge and expertise available in fundamental conservation, evolution, or constitutive principles, which are often expressed as a set of partial differential equations (PDEs). One can incorporate this type of knowledge into ML/AI models to enhance their predictive capability in sparse data regions. Nowadays, these approaches are coined as scientific machine learning (SciML) in the computational mathematics/mechanics communities. In particular, a widely used approach in SciML is to train a conventional deep learning (DL) model such as

Gaussian process regression (GPR) [43,44] or deep neural network (DNN) [45–49] with physical principle constraints. Existing research has demonstrated SciML’s capability in sparse-data scenarios for various application areas, such as environmental study [50,51], material science [52,53], and cardiovascular modeling [46,54,55].

Although most of the SciML applications are restricted to single physics systems, we envision the general concept can be extended to tackle the multi-physics problems in metal AM. Thus, this paper put forth a SciML framework for metal additive manufacturing processes to predict the temperature field and melt pool fluid dynamics via a physics-informed neural network (PINN). We aim to fully take advantage of the prediction capabilities of deep neural networks while significantly reducing the amount of costly training labeled data. To this end, the physical conservation laws of momentum, mass, and energy are fused into a fully connected neural network by penalizing the loss function with the residuals of the Navier-Stokes equations and enthalpy conservation equation on a set of collocation points. Owing to this additional knowledge, the learning process only requires a small amount of labeled data-set. Besides, to impose the necessary Dirichlet boundary condition (BC), we borrow the idea from the interface-capturing approach widely used in multiphase fluid mechanics, in which a small portion of the neural network is solely used to enforce the Dirichlet BC by a Heaviside function. This “hard” approach can not only precisely satisfy the Dirichlet BC but also speed up the learning process, compared with the conventional “soft” approach that uses additional constraint in the loss function to enforce the BC. Once the model is trained, the quantities of interest, such as temperature, velocity, pressure, and melt pool dimensions, can be predicted accurately.

The paper is structured as follows. Section 2 represents the physics-informed neural network framework, in which the PDEs of physical principles, design of loss function, enforcement of Dirichlet BC, and training procedures are introduced in an articulated way. Section 3 presents the training data generation by using a FEM based residual-based variational multi-scale thermal fluid flow formulation. Section 4 demonstrates the applications of the PINN framework to two representative manufacturing problems. The first application is using the PINN framework to solve a classic solidification problem from the textbook by Dantzig and Rappaz [56]. For this problem, the PINN is informed by the energy conservation law and trained without labeled data-set. We compare the performance of “hard” BC and “soft” BC on this problem in terms of both accuracy and learning efficiency. The predictive capability of the PINN is assessed by comparing it with the standard finite element method (FEM) with resolution refinement studies. The second application is utilizing the PINN framework to predict the temperature field and melt pool fluid dynamics for the 2018 NIST AM-Bench

test series. We utilize a validated finite element based variational multi-scale formulation (VMS) [30] to generate the synthetic training data-sets. The investigations show that the PINN, informed by conservation laws of momentum, mass, and energy, can accurately and efficiently predict the melt pool dimension, fluid field, and cooling rate for the three selective laser beam melting tests done by NIST with a small amount of training data. We summarize the contributions and limitations of the paper and outline future work in Sect. 5.

## 2 Machine learning model

### 2.1 Governing partial differential equations

This section presents the governing partial differential equations (PDEs) of the thermal-fluid flows in metal AM processes. The theory of the equations builds upon the tacit assumptions that the solid phase is a highly viscous fluid with the same constant density as the liquid phase, and the loss of metal material due to vaporization [57,58] and the effects on heat loss, composition change and fluid motion are negligible. A flat top surface is adopted based on the fact that the melt pool deformation is small compared with the melt pool dimensions in the problems considered in the paper. With the above assumptions, the thermal-fluid model based on conservation laws of momentum, mass, and energy is defined as the following coupled PDEs

$$\rho(\mathbf{u}_{,t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{g}) + \nabla p - 2\mu \Delta \mathbf{u} = \mathbf{0} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

$$(\rho c_p T)_{,t} + \mathbf{u} \cdot \nabla (\rho c_p T) + (\rho L f_L)_{,t} + \mathbf{u} \cdot \nabla (\rho L f_L) - \kappa \nabla^2 T - Q_T = 0 \tag{3}$$

Here Eqs. 1 and 2 are the Navier-Stokes equations of incompressible flows, where  $\mathbf{u}$  is the velocity field,  $p$  is the pressure field,  $\mathbf{g}$  is the gravitational acceleration vector,  $\nabla$  is the gradient operator,  $\Delta$  is the Laplace operator,  $\rho$  and  $\mu$  are the density and dynamic viscosity, respectively.

Equation 3 is the conservation equation of energy, where  $T$  is the temperature,  $c_p$  is the specific heat capacity,  $L$  is the latent heat of fusion,  $\kappa$  is the thermal conductivity,  $Q_T$  is an energy source.

To have well-posed systems, Eqs. 1–3 are subjected to the following Dirichlet and Neumann boundary conditions

$$\mathbf{u} = \mathbf{u}_{bc} \tag{4}$$

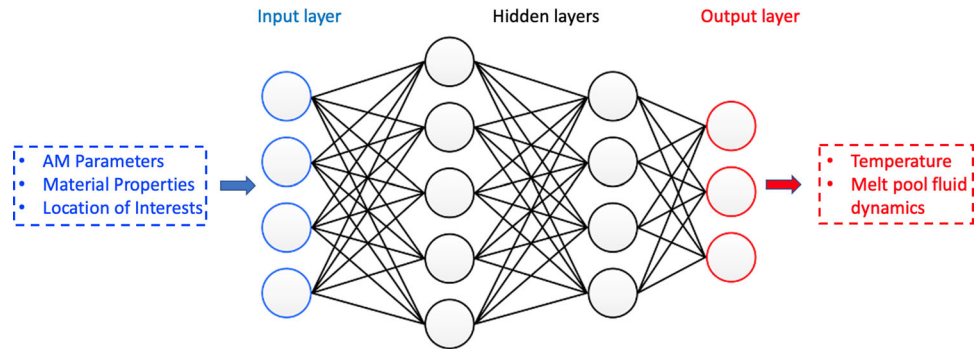
$$p = p_{bc} \tag{5}$$

$$T = T_{bc} \tag{6}$$

$$-p\mathbf{n} + 2\mu \nabla^S \mathbf{u} \cdot \mathbf{n} = \boldsymbol{\tau} \tag{7}$$

$$\kappa \nabla T \cdot \mathbf{n} = q \tag{8}$$

**Fig. 1** A fully connected deep neural network for metal AM



where  $u_{bc}$ ,  $p_{bc}$  and  $T_{bc}$  are the prescribed velocity, pressure, and temperature on Dirichlet boundaries, respectively.  $\tau$  and  $q$  are the prescribed traction and heat flux on the Neumann boundaries, respectively.  $\nabla^S$  is a symmetric gradient operator and  $\mathbf{n}$  is the unit normal vector on the boundary.

In the model, the solid and liquid phases are distinguished by a liquid fraction  $f_L$ , which takes 1 in the liquid phase, 0 in the solid phase, and a linear profile in the mushy zone [59].  $f_L$  is defined as

$$f_L = \begin{cases} 0 & \text{if } T < T_s \\ \frac{T - T_s}{T_l - T_s} & \text{if } T_s \leq T \leq T_l \\ 1 & \text{if } T > T_l \end{cases} \quad (9)$$

where  $T_s$  and  $T_l$  are the solidus and liquidus temperature, respectively.

With the assistance of  $f_L$ , the material properties in the thermal-fluid model are evaluated by the following interpolation

$$\psi = f_L \psi_L + (1 - f_L) \psi_S \quad (10)$$

where  $\psi$  denotes the specific material property in the model (e.g., density, dynamic viscosity, specific heat capacity, heat conductivity), and  $\psi_L$  and  $\psi_S$  are the corresponding property in the liquid and solid phase, respectively.

### 2.2 Physical informed neural network (PINN) for thermal-fluid flows

Neural network is a computing architecture that is vaguely inspired by the biological neural networks that constitute animal brains [60]. Some typical neural network architectures are fully connected neural network (FC-NN) [61], convolutional neural network (CNN) [62], and recurrent neural network (RNN) [63], which have been successfully used in a variety of machine learning applications, such as system identification and control, signal classification, pattern recognition, 3D reconstruction, sequence recognition, social network filtering, data mining, and medical diagnosis [48,64–67]. In this paper, the PINN of the thermal-fluid model makes

use of a fully connected deep neural network (FCNN) [61], where the neurons of adjacent layers are fully connected. Figure 1 shows the schematic picture of the fully connected neural network used in this paper, which consists of an input layer, hidden layers, and an output layer. A neural network with more than one hidden layer is conventionally called a deep neural network, whose function approximation capability increases with the number of hidden layers and neurons [68]. A deep neural network maps the input  $\mathbf{z}_0$  to the output  $\mathbf{z}_{N_{layer}-1}$  from the input layer to the output layer, where  $N_{layer}$  is the number of layers. In the hidden layers, each layer receives outputs from the previous layer and feeds forward inputs to the next layer. The relation of between the input  $\mathbf{z}_{l-1}$  and output  $\mathbf{z}_l$  of the  $l_{th}$  layer ( $l = 1, \dots, N_{layer} - 1$ ) is defined as

$$\mathbf{z}_l = \sigma_l(\mathbf{w}_l^T \mathbf{z}_{l-1} + \mathbf{b}_l) \quad (11)$$

where  $\mathbf{w}_l$  and  $\mathbf{b}_l$  are the weight matrix and bias vector of this layer. The dimensions of  $\mathbf{w}_l$  and  $\mathbf{b}_l$  are  $N_{l-1} \times N_l$  and  $N_l$ , respectively, where  $N_l$  is the number of neurons in  $l_{th}$  layer. In this paper, we denote the entire hidden parameters of a neural network as  $\mathbf{W} = \bigcup_l \mathbf{w}_l$  and  $\mathbf{b} = \bigcup_l \mathbf{b}_l$ . The dimensions of  $\mathbf{W}$  and  $\mathbf{b}$  are  $\sum_{l=0}^{N_{layer}-2} N_l \times N_{l+1}$  and  $\sum_{l=1}^{N_{layer}-1} N_l$ , respectively.

$\sigma_l$  in Eq. 11 is the activation function that can introduce the non-linearity to the system [69]. Widely used activation functions in deep learning are tanh function, rectified linear unit (Relu) function, and sigmoid function [70]. In this paper, we employ a swish activation function [71], which is a smoothed version of Relu function, defined as

$$\sigma_l(x) = swish(x) = x \text{ sigmoid}(x) = x / (1 + e^{-x}) \quad (12)$$

Neural network is a nonlinear parametric function approximator, so all the information of unknown function (velocity, pressure, and temperature) can be represented by hidden parameters  $\mathbf{W}$  and  $\mathbf{b}$ . The goal of the neural network is to learn the following mapping for a given set of manufacturing parameters (e.g., alloy properties, laser power, and scanning

speed):

$$[t, \mathbf{x}] \xrightarrow{\mathbf{W}, \mathbf{b}} [\mathbf{u}_{NN}, p_{NN}, T_{NN}] \tag{13}$$

where the input  $[t, \mathbf{x}]$  are the collocation points (in both space and time) of interest. The output  $\mathbf{u}_{NN}, p_{NN}, T_{NN}$  are the velocity, pressure, and temperature fields we want to predict. To enable such a mapping, the hidden parameters of the neural network,  $\mathbf{W}$  and  $\mathbf{b}$ , need to be identified by optimizing a meticulously designed loss function, which will be given in the next section. Once  $\mathbf{W}$ , and  $\mathbf{b}$  are determined, the output prediction can be easily achieved by a feed-forward evaluation, which is very efficient since only a few matrix multiplications are needed in Eq. 11.

### 2.2.1 Loss function design

The loss function,  $L(\mathbf{W}, \mathbf{b})$ , in the PINN for thermal-fluid flows consists of two components:  $L_{data}(\mathbf{W}, \mathbf{b})$  and  $L_{pde}(\mathbf{W}, \mathbf{b})$ , which represents the constraint of matching existing labeled data and the constraints of satisfying fundamental physical principles. Their definitions are given as follows.

Let  $\hat{\mathbf{u}}, \hat{p}$ , and  $\hat{T}$  denote the available labeled data for velocity, pressure, and temperature, respectively. These labeled data can be either obtained by experiments or validated high-fidelity simulations. As a mean squared deviation (MSD) of the discrepancy between the prediction and labeled data, the component in the loss function from the data constraint,  $L_{data}$ , is defined as

$$L_{data}(\mathbf{W}, \mathbf{b}) = \frac{1}{N_u} \sum_{i=1}^{N_u} [\mathbf{u}_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}) - \hat{\mathbf{u}}(\mathbf{x}_i, t_i)]^2 + \frac{1}{N_p} \sum_{i=1}^{N_p} [p_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}) - \hat{p}(\mathbf{x}_i, t_i)]^2 + \frac{1}{N_T} \sum_{i=1}^{N_T} [T_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}) - \hat{T}(\mathbf{x}_i, t_i)]^2 \tag{14}$$

where  $N_u, N_p$ , and  $N_T$  are the number of labeled velocity, pressure, and temperature data points, respectively. Conventional off-the-shelf machine learning tools purely minimize this loss function to identify the hidden parameters. The success of this approach requires a massive amount of data-sets. However, considering the cost of experimental measurements and high-fidelity simulations, these labeled velocity, pressure, and temperature data points are expensive (sometimes impossible) to obtain. This limitation hinders the direct application of big-data based machine learning tools to metal AM process prediction.

To alleviate the dependence on big-data, we substitute extra expertise in fundamental physical principles into the loss function. These physical principles, often expressed as a set of PDEs with appropriate initial and boundary conditions, are highly condensed knowledge of fundamental physical mechanisms that can inform the neural network. For that, we first define the following residuals of conservation equations of momentum, mass, and energy (corresponding to Eqs. (1–3)) as

$$\begin{cases} r_M := \rho(\mathbf{u}_{,t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{g}) + \nabla p - 2\mu \Delta \mathbf{u} \\ r_C := \nabla \cdot \mathbf{u} \\ r_T := (\rho c_p T)_{,t} + \mathbf{u} \cdot \nabla(\rho c_p T) + (\rho L f_L)_{,t} \\ \quad + \mathbf{u} \cdot (\rho L \nabla f_L) - \kappa \nabla^2 T - Q_T \end{cases} \tag{15}$$

To have well-posed systems, appropriate initial and boundary conditions are often necessary. In this paper, the initial boundary conditions are treated as part of the labeled data constraint. For boundary conditions (BCs), many existing PINN frameworks utilize a “soft” approach by designing additional loss components defined on the collocation points of boundaries to constrain the BCs. The downsides of this approach are two-fold: (1) The accuracy of satisfying the BCs is not guaranteed ; (2) The assigned weight of BC loss can affect learning efficiency, and no theory is existed to guide choosing the weight at this point.

In this paper, we treat the Dirichlet BC in a “hard” way by using a particular portion of the neural network to purely satisfy the prescribed Dirichlet BC. For that, we first define a Heaviside function as

$$H_\epsilon(\mathbf{x}) = \begin{cases} 1 - \cos[d(\mathbf{x})\pi/\epsilon] & \text{if } d(\mathbf{x}) < \epsilon \\ 1 & \text{if } d(\mathbf{x}) \geq \epsilon \end{cases} \tag{16}$$

where  $d(\mathbf{x})$  is the distance to the Dirichlet boundary.  $\epsilon$  defines a artificial thickness of the boundary. With  $H_\epsilon(\mathbf{x})$ , the predictions of the neural networks are defined as

$$\mathbf{u}_{NN} = \mathbf{u}_{bc}[1 - H_\epsilon(\mathbf{x})] + \mathbf{u}H_\epsilon(\mathbf{x}) \quad (17)$$

$$p_{NN} = p_{bc}[1 - H_\epsilon(\mathbf{x})] + pH_\epsilon(\mathbf{x}) \quad (18)$$

$$T_{NN} = T_{bc}[1 - H_\epsilon(\mathbf{x})] + TH_\epsilon(\mathbf{x}) \quad (19)$$

where  $\mathbf{u}_{bc}$ ,  $p_{bc}$ , and  $T_{bc}$  are the prescribed velocity, pressure, and temperature.  $\mathbf{u}$ ,  $p$ , and  $T$  are the solutions that satisfy the PDEs in the interior. Since  $H_\epsilon(\mathbf{x})$  smoothly changes from 1 to 0 as  $d(\mathbf{x})$  approaches to 0, the prediction will automatically satisfy the prescribed values by definition, without needing additional constraint. Then, the loss term from the PDEs with embedded Dirichlet BCs is defined as.

$$\begin{aligned} L_{pde}^1(\mathbf{W}, \mathbf{b}) = & \frac{1}{N_{r1}} \sum_{i=1}^{N_{r1}} r_M [\mathbf{u}_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}), \\ & p_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}), T_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b})]^2 \\ & + \frac{1}{N_{r1}} \sum_{i=1}^{N_{r1}} r_C [\mathbf{u}_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}), \\ & p_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}), T_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b})]^2 \\ & + \frac{1}{N_{r1}} \sum_{i=1}^{N_{r1}} r_T [\mathbf{u}_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}), \\ & p_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}), T_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b})]^2 \quad (20) \end{aligned}$$

where  $N_{r1}$  denotes the number of collocation points to constrain the PDEs.

In the thermal-fluid flow model for metal AM processes, Neumann BCs incorporate surface tension for flow field and laser for thermal field. In this paper, the Neumann BCs are handled by the conventional way, where the following term is added in the loss function

$$\begin{aligned} L_{pde}^2(\mathbf{W}, \mathbf{b}) = & \frac{1}{N_{r2}} \\ & \sum_{i=1}^{N_{r2}} \left\{ 2\mu \nabla^S \mathbf{u}_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}) \cdot \mathbf{n} \right. \\ & \left. - p_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b})\mathbf{n} - \boldsymbol{\tau}(\mathbf{x}_i, t_i) \right\}^2 \\ & + \frac{1}{N_{r3}} \sum_{i=1}^{N_{r3}} [k \nabla T_{NN}(\mathbf{x}_i, t_i, \mathbf{W}, \mathbf{b}) \cdot \\ & \mathbf{n} - q(\mathbf{x}_i, t_i)]^2 \quad (21) \end{aligned}$$

where  $N_{r2}$  and  $N_{r3}$  denote the number of collocation points on the fluid and temperature Neumann boundaries, respectively.

With above definitions, The hidden parameters  $\mathbf{W}$  and  $\mathbf{b}$  are obtained by minimizing the following total loss function, which are a linear combination of data constraint of  $L_{data}$  and PDE constraints of  $L_{pde}^1$  and  $L_{pde}^2$ .

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}} L(\mathbf{W}, \mathbf{b}) = & (1 - \lambda_{pde}^1 - \lambda_{pde}^2) L_{data}(\mathbf{W}, \mathbf{b}) \\ & + \lambda_{pde}^1 L_{pde}^1(\mathbf{W}, \mathbf{b}) + \lambda_{pde}^2 L_{pde}^2(\mathbf{W}, \mathbf{b}) \quad (22) \end{aligned}$$

where  $\lambda_{pde}^1$  and  $\lambda_{pde}^2$  are two positive numbers between 0 and 1, which define the weight of the physical law constraints in the loss function. The choice of the weights influences both the learning process and the prediction accuracy. No universal guideline exists for choosing the optimal weights at this point. We select the weights based on the ratios between different components in the loss function in this paper. We firstly estimate the magnitude of the PDE residual loss, boundary condition loss, and data loss. The ratios of PDE residuals can be roughly evaluated by the high fidelity simulations that generate training and validation data-sets (the formulation will be presented next). Then we specify the  $\lambda_{pde}^1$  and  $\lambda_{pde}^2$  so that the ratios of the three components are at the balanced level.

## 2.2.2 Learning procedure

The PINN model is trained by minimizing the loss function defined in Eq. 22 with respect to  $\mathbf{W}$  and  $\mathbf{b}$ . The minimization is executed by the following procedures: (1) The coordinates of collocation points and training data are substituted into Eq. 22. (2) Take the derivatives of the loss function with respect to  $\mathbf{W}$  and  $\mathbf{b}$ . (3) Update  $\mathbf{W}$  and  $\mathbf{b}$  by a gradient descent. Most of current machine learning frameworks solve the optimization problem by a stochastic gradient descent (SGD) algorithm, which is a stochastic approximation of the gradient descent optimization [72]. SGD only uses a subset of collocation points, randomly sampled from the input space at each iteration, to calculate the directional gradient. Research shows that SGD works very well to skip bad local minima. One issue with SGD is the oscillation of gradient direction caused by the random selection of sampled collocation points. In this paper, the Adam method [73] that combines adaptive learning rate and momentum methods is used to improve convergence speed [73].

The PINN learning process needs the spatial and temporal derivatives of  $\mathbf{W}$  and  $\mathbf{b}$ , which can be accurately and efficiently calculated by using automatic differentiation (AD) [74]. The basic idea of AD is to use the chain rule to back-propagate derivatives from the output layer to the input layer since the connection between layers of a neural network is analytically defined. Compared to numerical differentiation techniques (e.g., finite difference and finite element), AD does not suffer from truncation or round-off errors, resulting in much higher accuracy. AD has been gaining increasing attention in the machine learning community and has been implemented in many modern deep learning frameworks, such as TensorFlow [39], PyTorch [40], Theano [41], and

Caffe [42]. In this paper, the PINN formulation is implemented in TensorFlow.

### 3 Training data generation: high fidelity FEM simulations

Due to the limited available experimental labeled data-sets, the high-fidelity thermal-fluid finite element model based on our previous work in [30] is utilized to generate the labeled data for AM processes. Only a small portion of simulated data will be used to train the PINN model, and the simulated data is also used to access the model’s accuracy. The thermal-fluid FEM model makes use of a residual-based variational multi-scale formulation (VMS). The core formulation is briefly presented as follows. Let  $\mathcal{W}$  denote the testing function space for the Navier-Stokes and energy conservation equations,  $\mathcal{V}$  denote the unknown velocity  $\mathbf{u}$ , pressure  $p$ , and temperature  $T$  fields. The RBVMS formulation of thermal-fluid flows is stated as:  $\forall \{\mathbf{v}, q, \eta\} \in \mathcal{W}$ , find  $\{\mathbf{u}, p, T\} \in \mathcal{V}$ , such that

$$\begin{aligned} & \int_{\Omega} \mathbf{v} \cdot \rho (\mathbf{u}_{,t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{g}) d\Omega + \int_{\Omega} p \nabla \cdot \mathbf{v} d\Omega \\ & - \int_{\Gamma} \mathbf{v} \cdot \mathbf{h} d\Gamma + \int_{\Omega} \nabla^s \mathbf{v} : \mu \nabla^s \mathbf{u} d\Omega + \int_{\Omega} q \nabla \cdot \mathbf{u} d\Omega \\ & + \int_{\Omega} \eta [(\rho c_p T)_{,t} + \mathbf{u} \cdot \nabla (\rho c_p T) \\ & + (\rho L f_L)_{,t} + \mathbf{u} \cdot \nabla (\rho L f_L)] d\Omega \\ & + \int_{\Omega} \nabla \eta \cdot \kappa \nabla T d\Omega - \int_{\Omega} \eta Q_T d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_M \left( \mathbf{u} \cdot \nabla \mathbf{v} + \frac{\nabla q}{\rho} \right) \\ & \cdot \mathbf{r}_M d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \rho \tau_C \nabla \cdot \mathbf{v} r_C d\Omega \\ & - \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_M \mathbf{v} \cdot [\mathbf{r}_M \cdot \nabla \mathbf{u}] d\Omega \\ & - \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{\nabla \mathbf{v}}{\rho} : (\tau_M \mathbf{r}_M \otimes \tau_M \mathbf{r}_M) d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_T (\mathbf{u} \cdot \nabla \eta) r_T d\Omega = 0 \end{aligned} \tag{23}$$

where  $\Omega = \bigcup_e \Omega^e$  is the computational domain, which is decomposed into  $n_{el}$  elements.  $\mathbf{h}$  is the applied traction on the Neumann boundary  $\Gamma$ .  $\mathbf{r}_M$ ,  $r_C$ , and  $r_T$  are the residuals of momentum, continuity, and energy conservation equations (see Eq. 15).  $\tau_M$ ,  $\tau_C$ , and  $\tau_T$  are the corresponding stabilization parameters [30], defined as.

$$\tau_M = \left( \frac{4}{\Delta t^2} + \frac{4\|\mathbf{u}\|^2}{h^2} + \frac{16\nu^2}{h^4} \right)^{-1/2} \tag{24}$$

$$\tau_C = \frac{h^2}{12\tau_M} \tag{25}$$

$$\tau_T = \left( \frac{4}{\Delta t^2} + \frac{4\|\mathbf{u}\|^2}{h^2} + \frac{16\kappa^2}{\rho^2 c_p^2 h^4} \right)^{-1/2} \tag{26}$$

where  $\Delta t$  is the time step,  $h$  is the characteristic element length, and  $\nu$  is the kinematic viscosity. Other widely used stabilization parameters can be found in [75,76,76,77,77,78]. The unknown velocity, pressure, and temperature are solved in a fully coupled fashion. Generalized- $\alpha$  method is used for time integration. The nonlinear equations are linearized by Newton’s method. The resulting linear systems are solved by a generalized minimal residual method (GMRES) with block preconditioning [79]. The formulation is implemented for parallel environments using the Message Passing Interface (MPI). The formulation has been validated with a series of metal manufacturing problems in [30]. It worths noting that the VMS and its extensions on moving fluid domains using Arbitrary Lagrangian-Eulerian technique (ALE-VMS) [80–85] and Space-Time (ST-VMS) technique [86–90] have been used as high-fidelity models to simulate a set of challenging fluid dynamics and fluid-structure interaction problems [91]. Several recent applications include environmental flows [92], wind energy [87,93–102], tidal energy [103–105], biomechanics [106,107,107,108], gas turbine [109–113] and transportation engineering [114,115].

## 4 Applications

### 4.1 Solidification of aluminum in a graphite mold without labeled data

The solidification process of aluminum in a graphite mold from the textbook *Solidification* by Dantzig and Rappaz [56] is investigated to assess the PINN formulation’s performance. Only thermodynamics with phase transition is considered in the simulations here. Figure 2 shows the problem setup, where the left half of the domain ( $-0.4 \text{ m} \leq x \leq 0.0 \text{ m}$ ) is occupied by a solid graphite mold with temperature  $T_{low} = 298.15 \text{ K}$ , and the right half of the domain ( $0.0 \text{ m} < x \leq 0.4 \text{ m}$ ) is occupied with liquid aluminum with temperature  $T_{high} = 973.15 \text{ K}$ , which is higher than the melting temperature of aluminum  $T_{melt} = 933.15 \text{ K}$ . The solidification process, depicted in Fig. 2 (lower), occurs by transferring heat from the aluminum into the mold, and the solid–liquid interface propagates towards the right end. The material properties of the graphite mold and aluminum are given in Table 1. The analytical solutions have been derived in [56] for this problem and are specified as follows.

$$x^* = 7.095 \times 10^{-3} \sqrt{t} \text{ m} \tag{27}$$

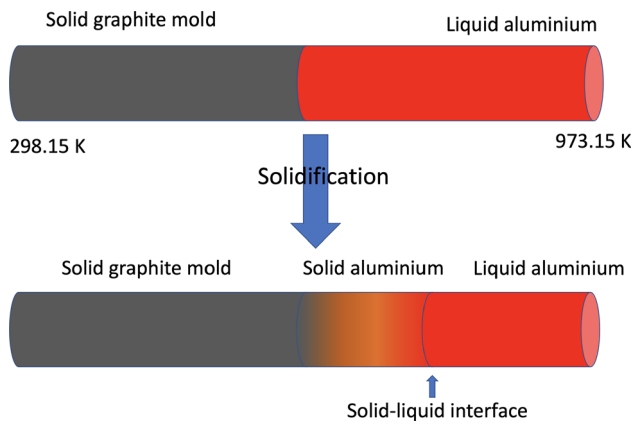


Fig. 2 1D solidification process from [56]

Table 1 Definition of material properties for the solidification problem

Materials	Graphite	Aluminum (solid)	Aluminum (liquid)
Density (kg/m <sup>3</sup> )	2200	2555	2555
Specific heat (J/(kg K))	1700	1190	1190
Thermal conductivity (W/(kg K))	100	211	91
Latent heat (J/kg)	–	398000	–

$$T_m = 769.95 + 471.8 \operatorname{erf} \left( \frac{96.69x}{\sqrt{t}} \right) \text{ K} \tag{28}$$

$$T_s = 769.95 + 360.2 \operatorname{erf} \left( \frac{60.02x}{\sqrt{t}} \right) \text{ K} \tag{29}$$

$$T_l = 973.15 - 111.4 \operatorname{erfc} \left( \frac{91.39x}{\sqrt{t}} \right) \text{ K} \tag{30}$$

where  $x^*$  is the solid–liquid interface location over time,  $T_m$ ,  $T_s$ , and  $T_l$  are the temperature distribution in the mold, solid aluminium, and liquid aluminium, respectively.

Despite the simplicity, solving this problem provides valuable insights into the solidification behavior and the machine learning model’s performance. The PINN model employs 5 hidden layers and 200 neurons of each layer, which provides better results over others based on our non-exhaustive investigation. The neural network is only informed with the energy conservation principle defined in Eq. 3 and trained without labeled data-set. No data component is not used in the loss function. The PINN model predicts the temperature distribution from  $t = 5$  s to  $t = 10$  s.

Figure 3 illustrates the PINN setup and the resulting temperature prediction in the space-time ( $x - t$ ) slab for the solidification prediction process. We compare the performance of the proposed “hard” approach with the conventional “soft” approach for the Dirichlet boundary condition in Fig. 4, which depicts the learning process and temperature predictions at 10 s. The plot shows that the “hard” approach can not only facilitate the learning process (see Fig. 4 (left)) but also produce more accurate temperature prediction (see Fig. 4 (right)).

One important question is how the PINN’s predictive capability compares with traditional numerical methods, such as the finite element method (FEM). To answer this question, we simulate this solidification problem by using PINN with four different numbers of collocation points and linear FEM with four equivalent resolutions ( $N_x = 50, 100, 150,$  and  $200$  along  $x$  direction). Figure 5 shows the predictions of PINN and FEM for the time history of the solid–liquid interface position with the four resolutions. The convergence rate of error of temperature prediction over the  $x - t$  slab is shown in Figure 6. The two plots indicate

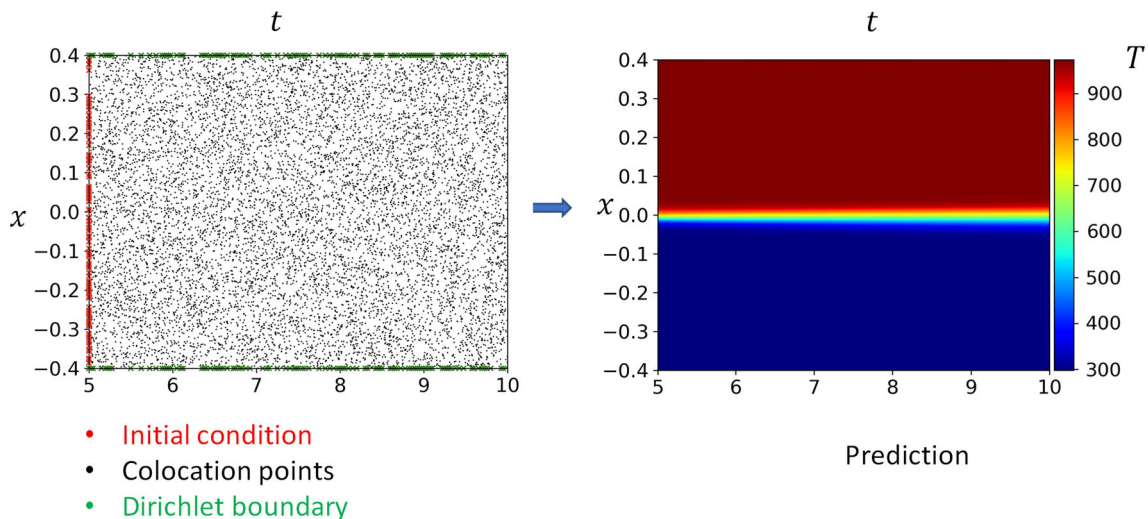


Fig. 3 PINN model for the solidification problem. Left: PINN setup. Right: Temperature prediction



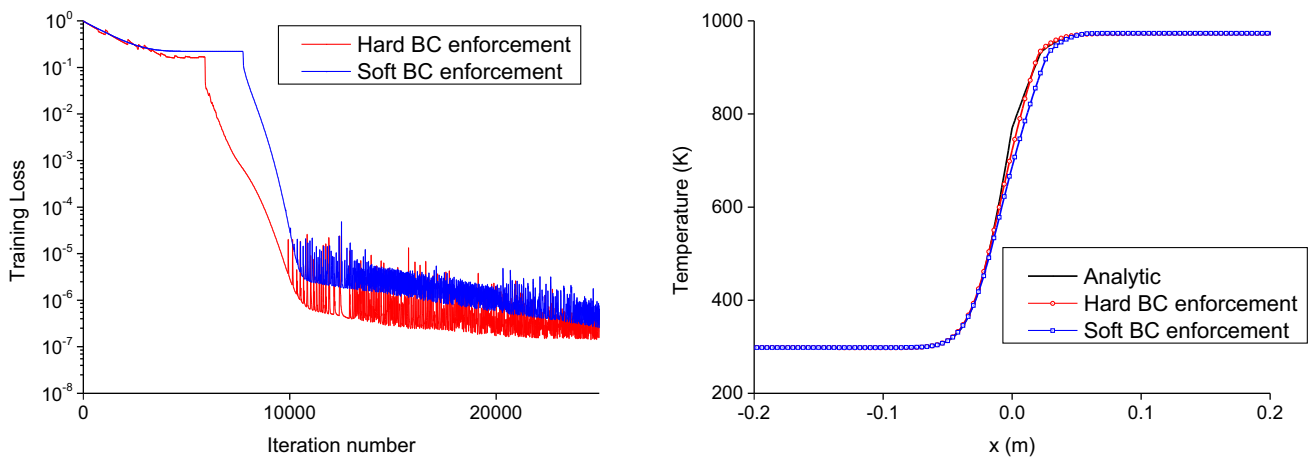


Fig. 4 Comparison between “hard” BC and “soft” BC. Left: learning speed. Right: The temperature distribution at 10 s

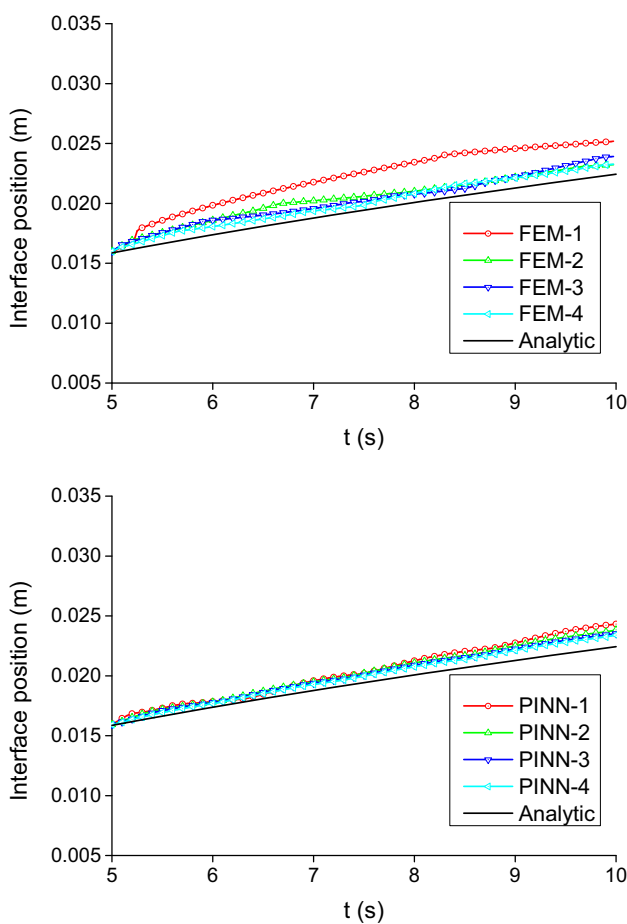


Fig. 5 Predicted time history of solid–liquid interface position. Left: refinement study of FEM. Right: refinement study of PINN

that PINN and FEM obtain similar convergence rates. However, when the resolution is low, the PINN still attains high accuracy while a noticeable discrepancy is observed for the standard FEM.

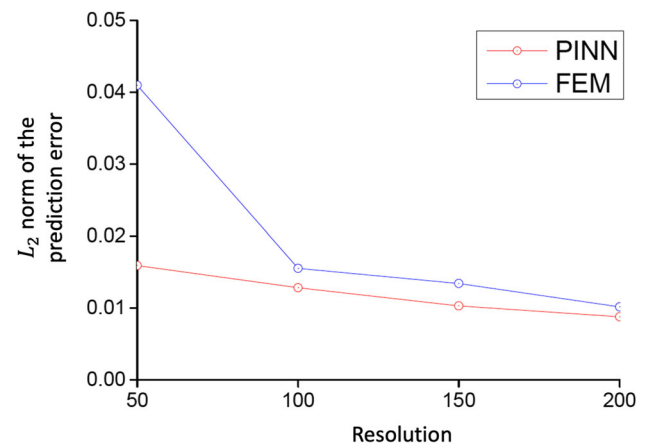


Fig. 6  $L_2$  norm of prediction error of PINN and FEM with different resolutions

### 4.2 NIST AM-bench test series

In this section, we apply the PINN framework to the Additive Manufacturing Benchmark (AM-Bench) test series conducted by the National Institute of Standards and Technology (NIST) [116,117]. In 2018, NIST performed a series of metal AM experiments with different manufacturing parameters, which attracted blind simulations to compare with the in-situ and ex-situ measurements, such as temperature, melt pool dimensions, and micro-structures [116]. The archived experimental measurements provide valuable benchmark data for modelers to test the predictive capabilities of simulation models. In this paper, we use the proposed PINN framework to predict the temperature, melt pool fluid dynamics, melt pool dimensions, and cooling rates during the NIST AM processes, which corresponds to the first challenge in the NIST AM-Bench test series [117]. To the authors’ best knowledge, this is the first application of PINN to three-dimensional metal AM processes.

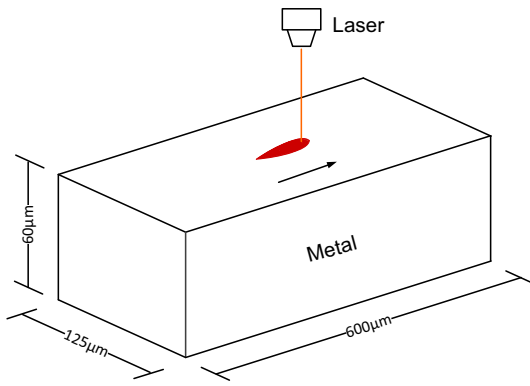


Fig. 7 NIST AM-bench test series

Figure 7 shows a schematic description of the NIST AM-Bench test, a selective laser beam melting process of a pure Inconel 625 (IN625) substrate. The material properties used in this paper are summarized in Table 2. The laser is applied by imposing the following moving heat flux on the substrate.

$$\kappa \nabla T \cdot \mathbf{n} = q_{laser} = \frac{2Q\eta}{\pi r_b^2} \exp\left(\frac{-2((x - V_s t)^2 + y^2)}{r_b^2}\right) \quad (31)$$

where  $Q$  is the laser power,  $\eta$  is the absorptivity,  $r_b$  is the laser beam radius,  $V_s$  is the laser scanning speed.  $\eta = 0.43$  and  $r_b = 50 \mu\text{m}$  are used in this paper. Table 3 lists the laser power and scanning speed of three cases used in the NIST experiments. Based on the fact that the top surface deformation is relatively small compared with the melt pool dimension, a flat top surface is assumed, and the following boundary condition is applied for the fluid field.

$$-pn + 2\mu \nabla^s \mathbf{u} \cdot \mathbf{n} = \boldsymbol{\tau} = \frac{d\sigma}{dT} [\nabla T - (\nabla T \cdot \mathbf{n})\mathbf{n}] \quad (32)$$

Table 2 Mechanical properties of IN625

Name	Notation (units)	Value
Density	$\rho$ ( $\text{kg m}^{-3}$ )	8440
Solidus temperature	$T_s$ (K)	1563
Liquid temperature	$T_l$ (K)	1623
Solid specific heat capacity	$c_{ps}$ ( $\text{J kg}^{-1} \text{K}^{-1}$ )	$0.2441T + 338.39$
Liquid specific heat capacity	$c_{pl}$ ( $\text{J kg}^{-1} \text{K}^{-1}$ )	709.25
Solid solid conductivity	$\kappa_s$ ( $\text{W m}^{-1} \text{K}^{-1}$ )	$3.0 \times 10^{-5} T^2 - 0.0366T + 18.588$
Liquid solid conductivity	$\kappa_l$ ( $\text{W m}^{-1} \text{K}^{-1}$ )	30.078
Latent heat of fusion	$c_L$ ( $\text{KJ kg}^{-1} \text{K}^{-1}$ )	290
Dynamic viscosity	$\mu$ (Pa s)	$7 \times 10^{-3}$
Marangoni coefficient	$\frac{\partial \gamma}{\partial T}$ ( $\text{N m}^{-1} \text{K}^{-1}$ )	$-2 \times 10^{-5}$
Reference temperature	$T_{ref}$ (K)	295

Table 3 Three laser parameters

Parameters	Case A	Case B	Case C
Laser power	150 W	195 W	195 W
Scan speed	0.4 m/s	0.8 m/s	1.2 m/s

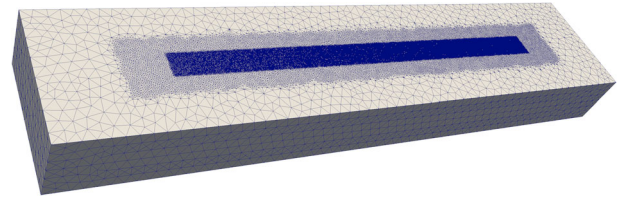
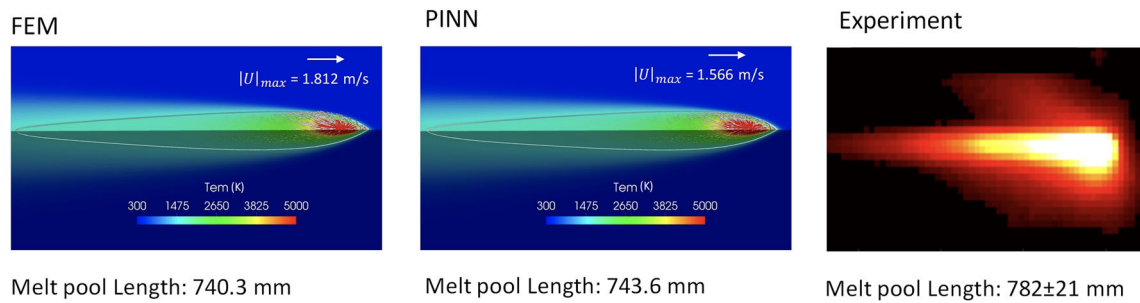


Fig. 8 The mesh employed in the high-fidelity FEM thermal-fluid simulations

where  $\frac{d\sigma}{dT}$  is the Marangoni coefficient, which is only effective in the tangential direction of the temperature gradient. Except for the top surface, no-slip and fixed reference temperature boundary conditions are used for other surfaces.

The PINN model employs a fully connected neural network with 5 hidden layers and 250 neurons per layer. The model is physically informed by the conservation laws of momentum, mass, and energy defined in Eqs. 1–3. A total of 788, 651 collocation points in the  $\mathbf{x} - t$  slab are used to penalize the PDE residuals (see Eq. 15) in the loss function.

Due to the limited point-wise measurements provided by NIST, we use the high-fidelity finite element thermal-fluid model presented in Sect. 3 to generate the labeled data-sets to facilitate the training. The FEM simulations use the following boundary conditions. For the fluid field, the no-slip boundary condition is used for all the surfaces except the top surface, where no penetration and the traction boundary condition defined in Eq. 32 is used. For the temperature field, zero heat flux is used for all the surfaces except the top surface, where heat laser defined in Eq. 31 is used. 4,464,276 linear tetrahedral elements are employed for the computa-



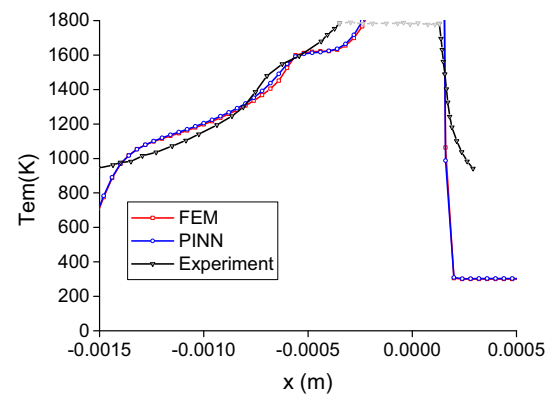
**Fig. 9** Comparison of the predictions of the temperature and melt pool fluid dynamics of FEM, PINN and experiment for case B (195 W, 0.8 m/s) at quasi-steady state (2 ms), when the melt pool shape is not chang-

ing. Left: FEM prediction. Middle: PINN prediction. Right: Thermal video frame based on radiance temperature from experiment[118]

tions. A local refinement, shown in Fig. 8, is designed to capture the moving laser. Time step  $\Delta t = 1 \times 10^{-6}$  s. These simulations are executed at Stampede2 at Texas Advanced Computing Center (TACC) with 192 CPU processors. The simulated results are treated as “ground truth” after being compared with the NIST experimental measurements. The FEM simulations have been run up to 2.0 ms for all the three cases listed in Table 3, but only a small portion of the simulated data between 1.2 and 1.5 ms is used as labeled training data in the PINN model, which then predicts the manufacturing processes for a wider time interval from 0 ms to 2.0 ms.

We first compare the predicted results of FEM and PINN with available experimental data for case B (195 W, 0.8 m/s). The purpose is two-fold: (1) Ensure the credibility of FEM data as the training data; (2) Validate the PINN model. Figure 9 shows the temperature field, melt pool fluid dynamics, and melt pool shape at 2.0 ms. The fast-moving laser, along with the effect of a negative Marangoni coefficient that drives the liquid metal from the higher temperature region to the lower temperature region, leading to a long and shallow melt pool. The predicted temperature profile along the scan track and experimental measurement extracted from [119] are plotted in Fig. 10 for comparison. The predicted results by both FEM and PINN show good agreement with available experimental data. Figures 9 and 10 also show that the PINN model, with a moderate amount of training data, can generate very similar predictions of temperature, melt pool length, and melt pool fluid velocity to those of FEM.

We then apply the PINN model to all the three cases listed in Table 3. The predicted melt pool shape and the fluid velocity field within the melt pool are presented in Fig. 11. The laser results in high velocity in the melt pool, which reaches up to 1.641 m/s, 1.566 m/s, and 1.446 m/s for the case A, B, and C, respectively. The predicted melt pool dimensions compared with the present FEM results, the thermal-fluid simulation results by Gan et al. [32] that won an award in the NIST AM-bench competition, and available

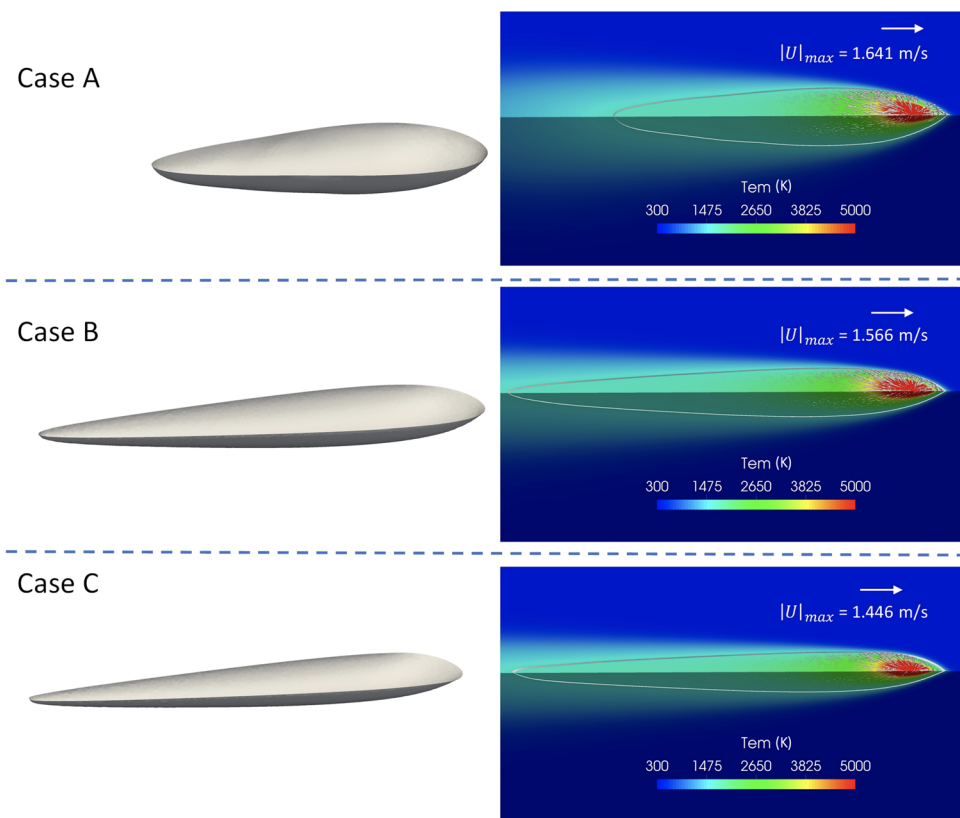


**Fig. 10** Temperature profile along the scan track for case B (195 W, 0.8 m/s) at quasi-steady state. Experimental data extracted from NIST AM-Bench test series [119] are also plotted for comparison. (Please note we cut off the temperature inside the melt pool since the experimental provides an almost constant temperature.)

NIST experimental measurements are listed in Table 4. For melt pool length, the relative discrepancy (with respect to the mean NIST experimental measurements) of PINN, FEM, and Gan’s predictions are 9.7%, 11.3%, and 17.8% for case A, 5.1%, 4.7% and 8.1% for case B, and 2.9%, 3.6% and 4.1% for case C. Similar accuracy is observed among these approaches. Based on the model predictions, we notice that case A generates the smallest melt pool length but the biggest width and depth, which could be due to the low scanning speed that gives the laser more time to melt the underneath metal. In contrast, case B and C have a slightly bigger laser power but much faster scanning speeds, which consequently result in longer melt pool lengths but smaller widths and depths. Between case B and case C with the same laser power, lower spanning (case B) speed results in bigger melt pool dimensions in all directions.

A critical factor in metal AM is the cooling rate, which profoundly influences dendrite arm spacing, grain structure, micro-segregation, and hot cracking. In this paper, the cooling rate is calculated as

**Fig. 11** Melt pool shape and the temperature and melt pool flow velocity predicted by PINN for case A, B and C at quasi-steady state (2 ms)



**Table 4** Melt pool dimensions of case A, B and C

Cases	Approaches	Length ( $\mu\text{m}$ )	Width ( $\mu\text{m}$ )	Depth ( $\mu\text{m}$ )
Case A	PINN	594.8 (9.7%)	193.3	64.0
	FEM	584.4 (11.3%)	190.8	62.8
	Gan et al. [32]	542 (17.8%)	–	–
	Experiment [119]	659 $\pm$ 21	–	–
Case B	PINN	740.3 (5.1%)	160.0	52.8
	FEM	743.6 (4.7%)	157.5	52.5
	Gan et al. [32]	843 (8.1%)	–	–
	Experiment [119]	782 $\pm$ 21	–	–
Case C	PINN	732.5 (2.9%)	131.5	43.2
	FEM	727.2 (3.6%)	130.3	42.6
	Gan et al. [32]	785 (4.1%)	–	–
	Experiment [119]	754 $\pm$ 46	–	–

$$R_c = \frac{T_s - 1273.15K}{t_c} \quad (33)$$

where  $t_c = (D_s - D_{1273.15})/V_s$ , the cooling time interval determined by dividing the distance between solidus temperature and 1273.15 K by the scanning speed  $V_s$ .

Table 5 presents the PINN prediction of cooling rate  $R_c$  for the case A, B, and C. The results of FEM, Gan's results, and experimental measurements are also listed for compar-

ison. Both modeling and experiment show that cooling rate increases from case A to case C. For the cooling rate, the relative discrepancy (with respect to the mean NIST measurements) of PINN, FEM, and Gan's predictions are 37.7%, 29.0% and 17.6% for case A, 8.1%, 12.7% and 26.3% for case B, and 7.8%, 7.8% and 11.7% for case C. Although all the models' prediction accuracy becomes lower compared with melt pool dimension prediction, we notice that the NIST measurements in cooling rates also exhibit significantly higher fluctuations than melt pool dimension measurements. Nevertheless, if using this discrepancy as an accuracy metric, the

**Table 5** Cooling rates of case A, B and C

Cases	Approaches	Solid cooling rate (K/s)
Case A	PINN	$8.54 \times 10^5$ (37.7%)
	FEM	$8.00 \times 10^5$ (29.0%)
	Gan et al. [32]	$5.11 \times 10^5$ (17.6%)
	Experiment [119]	$6.20 \times 10^5 \pm 7.99 \times 10^4$
Case B	PINN	$8.59 \times 10^5$ (8.1%)
	FEM	$8.16 \times 10^5$ (12.7%)
	Gan et al. [32]	$6.89 \times 10^5$ (26.3%)
	Experiment [119]	$9.35 \times 10^5 \pm 1.43 \times 10^5$
Case C	PINN	$1.38 \times 10^6$ (7.8%)
	FEM	$1.38 \times 10^6$ (7.8%)
	Gan et al. [32]	$11.30 \times 10^5$ (11.7%)
	Experiment [119]	$1.28 \times 10^6 \pm 3.94 \times 10^5$

proposed PINN model only underperforms in case A and outperforms in both case B and case C, which is advantageous compared with the other two high-fidelity FEM simulations that employ millions of elements.

## 5 Conclusion

This paper presents the first attempt of using the physics-informed neural network (PINN) to predict the temperature and melt pool fluid dynamics in metal AM processes. We applied the PINN model to two representative metal manufacturing problems. The results show that the PINN can accurately predict the quantities of interest by only using a small amount of labeled training data. This paper is also the first few applications of scientific machine learning (SciML), currently confined to single-phase systems, to complex multi-scale and multi-physics problems that involve multi-phase fluid dynamics, heat transfer and phase transition. The two major technical contributions relevant to metal AM of the paper are:

- A SciML framework for metal AM processes, which can accurately predict temperature, pressure, and velocity field without relying on big-data.
- A “hard” approach for imposing Dirichlet boundary condition, which exactly imposes the prescribed value and speeds up the learning process.

Although deep learning models cannot replace conventional numerical tools that will continue to be the principal player, the initial success presented in this paper demonstrates PINN’s potential on the modeling and prediction of complicated metal AM processes and paves the way for the broad adoption in advanced manufacturing.

We also have to admit that this paper does not comprehensively handle the complexity of metal AM processes. To be precise, the PINN model here does not resolve the ambient gas phase, free-surface deformation of the melt pool, and the evaporation phenomenon, although the effects are not crucial for the applications considered in the paper. In terms of computational efficiency in AM simulations, we didn’t make a quantitative comparison between FEM and PINN in this paper because of the different computing machines: the FEM was performed in a CPU parallel environment with 192 processors at Stampede2, while the PINN was performed using GPU with only four cores at Frontera. The general observation is that the more training data is, the more efficient the PINN is. Under an extreme scenario with no training data, the PINN, as a PDE solver, is two times slower than FEM in a sequential computing environment, as tested in the 1D solidification problem.

In the future, the multi-phase Navier–Stokes will be enhanced with the evaporation model in the momentum equations, which was used in our control volume finite element model [14], to capture the heat loss, composition change, and fluid motion induced by evaporation. Additional PDEs, such as convection equation of level set or volume-of-fluid used in the previous works [23,30], will be incorporated into the PINN to enable modeling metal AM process at the powder scale.

**Acknowledgements** J. Yan is partially supported by ASME Robert M. and Mary Haythornthwaite Research Initiation Award and Singapore National Research Foundation (NRF2018-ITS004-0011). The PINN models were trained at the Texas Advanced Computing Center (Tacc) through a startup allocation on Frontera (CTS20014). These supports are greatly acknowledged.

## References

1. Zhao C, Fezzaa K, Cunningham R, Wen H, De Carlo F, Chen L, Rollett A, Sun T (2017) Real-time monitoring of laser powder bed fusion process using high-speed x-ray imaging and diffraction. *Sci Rep* 7(1):1–11
2. Cunningham R, Zhao C, Parab N, Kantzos C, Pauza J, Fezzaa K, Sun T, Rollett A (2019) Keyhole threshold and morphology in laser melting revealed by ultrahigh-speed x-ray imaging. *Science* 363(6429):849–852
3. Guo Q, Zhao C, Qu M, Xiong L, Hojjatzadeh S, Escano L, Parab N, Fezzaa K, Sun T, Chen L (2020) In-situ full-field mapping of melt flow dynamics in laser metal additive manufacturing. *Addit Manuf* 31:100939
4. NIST Additive Manufacturing Benchmark Test Series (AM-BENCH) (2020). <https://www.nist.gov/ambench>. Accessed 03 Aug 2020
5. AFRL Additive Manufacturing Modeling Challenge Series (2020). <https://www.americamakes.us/america-makes-and-afrl-announce-am-modeling-challenge/>. Accessed 07 July 2020
6. Noble C, Anderson A, Barton N, Bramwell J, Capps A, Chang M, Chou J, Dawson D, Diana E, Dunn T (2017) Ale3d: an arbitrary Lagrangian–Eulerian multi-physics code. Technical

- report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States)
7. Khairallah S, Anderson A, Rubenchik A, King W (2016) Laser powder-bed fusion additive manufacturing: physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones. *Acta Mater* 108:36–45
  8. Roehling TT, Wu SS, Khairallah SA, Roehling JD, Soezeri SS, Crumb MF, Matthews MJ (2017) Modulating laser intensity profile ellipticity for microstructural control during metal additive manufacturing. *Acta Mater* 128:197–206
  9. Khairallah S, Martin A, Lee J, Guss G, Calta N, Hammons J, Nielsen M, Chaput K, Schwalbach E, Shah M, Chapman G, Willey T, Rubenchik A, Anderson A, Wang Y, Matthews M, King W (2020) Controlling interdependent meso-nanosecond dynamics and defect generation in metal 3d printing. *Science* 368(6491):660–665
  10. Knapp G, Mukherjee T, Zuback J, Wei H, Palmer T, De A, DebRoy T (2017) Building blocks for a digital twin of additive manufacturing. *Acta Mater* 135:390–399
  11. Mukherjee T, Wei H, De A, DebRoy T (2018) Heat and fluid flow in additive manufacturing-part i: modeling of powder bed fusion. *Comput Mater Sci* 150:304–313
  12. Mukherjee T, Wei H, De A, DebRoy T (2018) Heat and fluid flow in additive manufacturing-part ii: Powder bed fusion of stainless steel, and titanium, nickel and aluminum base alloys. *Comput Mater Sci* 150:369–380
  13. Lin S (2019) Numerical methods and high performance computing for modeling metallic additive manufacturing processes at multiple scales. Ph.D. thesis, Northwestern University
  14. Lin S, Gan Z, Yan J, Wagner G (2020) A conservative level set method on unstructured meshes for modeling multiphase thermo-fluid flow in additive manufacturing processes. *Comput Methods Appl Mech Eng* 372:113348
  15. Attar E, Körner C (2011) Lattice Boltzmann model for thermal free surface flows with liquid–solid phase transition. *Int J Heat Fluid Flow* 32(1):156–163
  16. Körner C, Attar E, Heisl P (2011) Mesoscopic simulation of selective beam melting processes. *J Mater Process Technol* 211(6):978–987
  17. Körner C, Bauereiß A, Attar E (2013) Fundamental consolidation mechanisms during selective beam melting of powders. *Model Simul Mater Sci Eng* 21(8):085011
  18. Zohdi TI (2014) Additive particle deposition and selective laser processing—a computational manufacturing framework. *Comput Mech* 54(1):171–191
  19. Zohdi T (2014) A direct particle-based computational framework for electrically enhanced thermo-mechanical sintering of powdered materials. *Math Mech Solids* 19(1):93–113
  20. Ganeriwala R, Zohdi TI (2014) Multiphysics modeling and simulation of selective laser sintering manufacturing processes. *Procedia Cirp* 14:299–304
  21. Yan W, Ge W, Qian Y, Lin S, Zhou B, Liu WK, Lin F, Wagner GJ (2017) Multi-physics modeling of single/multiple-track defect mechanisms in electron beam selective melting. *Acta Mater* 134:324–333
  22. Yan W, Qian Y, Ge W, Lin S, Liu WK, Lin F, Wagner GJ (2018) Meso-scale modeling of multiple-layer fabrication process in selective electron beam melting: inter-layer/track voids formation. *Mater Des* 141:210–219
  23. Yan W, Lin S, Kafka O, Lian Y, Yu C, Liu Z, Yan J, Wolff S, Wu H, Ndip-Agbor E, Mozaffar M, Ehmann K, Cao J, Wagner G, Liu W (2018) Data-driven multi-scale multi-physics models to derive process-structure-property relationships for additive manufacturing. *Comput Mech* 61(5):521–541
  24. Yan W, Ge W, Smith J, Lin S, Kafka O, Lin F, Liu W (2016) Multi-scale modeling of electron beam melting of functionally graded materials. *Acta Mater* 115:403–412
  25. Chen H, Yan W (2020) Spattering and denudation in laser powder bed fusion process: multiphase flow modelling. *Acta Mater* 196:154–167
  26. Panwisawas C, Qiu C, Anderson MJ, Sovani Y, Turner RP, Attallah MM, Brooks JW, Basoalto HC (2017) Mesoscale modelling of selective laser melting: thermal fluid dynamics and microstructural evolution. *Comput Mater Sci* 126:479–490
  27. Li X, Zhao C, Sun T, Tan W (2020) Revealing transient powder-gas interaction in laser powder bed fusion process through multi-physics modeling and high-speed synchrotron x-ray imaging. *Addit Manuf* 35:101362
  28. Megahed M, Mindt H-W, Shula B, Peralta A, Neumann J (2016) Powder bed models-numerical assessment of as-built quality. In: 57th AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference, p 1657
  29. Mindt H-W, Desmaison O, Megahed M, Peralta A, Neumann J (2018) Modeling of powder bed manufacturing defects. *J Mater Eng Perform* 27(1):32–43
  30. Yan J, Yan W, Lin S, Wagner G (2018) A fully coupled finite element formulation for liquid–solid–gas thermo-fluid flow with melting and solidification. *Comput Methods Appl Mech Eng* 336:444–470
  31. Fan Z, Li B (2019) Meshfree simulations for additive manufacturing process of metals. *Integrat Mater Manuf Innov* 8(2):144–153
  32. Gan Z, Lian Y, Lin SE, Jones KK, Liu WK, Wagner GJ (2019) Benchmark study of thermal behavior, surface topography, and dendritic microstructure in selective laser melting of inconel 625. *Integrat Mater Manuf Innov* 8(2):178–193
  33. Liu Z, Wu C, Koishi M (2019) Transfer learning of deep material network for seamless structure-property predictions. *Comput Mech* 64(2):451–465
  34. Liu Z, Wu C, Koishi M (2019) A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials. *Comput Methods Appl Mech Eng* 345:1138–1168
  35. Liu Z, Wu C (2019) Exploring the 3d architectures of deep material network in data-driven multiscale mechanics. *J Mech Phys Solids* 127:20–46
  36. Liu Z, Kafka O, Yu C, Liu W (2018) Data-driven self-consistent clustering analysis of heterogeneous materials with crystal plasticity. In: Oñate E, Peric D, de Souza Neto E, Chiumenti M (eds) *Advances in computational plasticity*. Springer, pp 221–242
  37. Liu Z, Fleming M, Liu W (2018) Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials. *Comput Methods Appl Mech Eng* 330:547–577
  38. Liu Z, Bessa M, Liu W (2016) Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials. *Comput Methods Appl Mech Eng* 306:319–341
  39. Abadi M, Barham P, Chen J., Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, et al (2016) Tensorflow: a system for large-scale machine learning. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pp 265–283
  40. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al (2019) Pytorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (eds) *Advances in neural information processing systems*, pp 8024–8035
  41. Bastien F, Lamblin P, Pascanu R, Bergstra J, Goodfellow I, Bergeron A, Bouchard N, Warde-Farley D, Bengio Y Theano: new features and speed improvements. [arXiv:1211.5590](https://arxiv.org/abs/1211.5590)

42. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on multimedia, pp 675–678
43. Yang X, Barajas-Solano D, Tartakovsky G, Tartakovsky A (2019) Physics-informed Cokriging: a Gaussian-process-regression-based multifidelity method for data-model convergence. *J Comput Phys* 395:410–431
44. Raissi M, Perdikaris P, Karniadakis GE (2017) Machine learning of linear differential equations using Gaussian processes. *J Comput Phys* 348:683–693
45. Lagaris I, Likas A, Fotiadis D (1998) Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans Neural Netw* 9(5):987–1000
46. Raissi M, Yazdani A, Karniadakis G (2020) Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* 367(6481):1026–1030
47. Sun L, Gao H, Pan S, Wang J-X (2020) Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput Methods Appl Mech Eng* 361:112732
48. Zissis D, Xidias EK, Lekkas D (2015) A cloud based architecture capable of perceiving and predicting multiple vessel behaviour. *Appl Soft Comput* 35:652–661
49. Raissi M, Perdikaris P, Karniadakis GE Physics informed deep learning (part i): data-driven solutions of nonlinear partial differential equations. [arXiv:1711.10561](https://arxiv.org/abs/1711.10561)
50. He Q, Tartakovsky G, Barajas-Solano D, Tartakovsky A (2019) Physics-informed deep neural networks for multiphysics data assimilation in subsurface transport problems. *AGUFM 2019:H34B–02*
51. Tartakovsky A, Marrero C, Perdikaris P, Tartakovsky G, Barajas-Solano D (2020) Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resour Res* 56(5):e2019WR026731
52. Lu L, Dao M, Kumar P, Ramamurty U, Karniadakis GE, Suresh S (2020) Extraction of mechanical properties of materials through deep learning from instrumented indentation. *Proc Nat Acad Sci* 117(13):7052–7062
53. He Q, Chen J (2020) A physics-constrained data-driven approach based on locally convex reconstruction for noisy database. *Comput Methods Appl Mech Eng* 363:112791
54. Raissi M, Perdikaris P, Karniadakis G (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 378:686–707
55. Kissas G, Yang Y, Hwuang E, Witschey W, Detre J, Perdikaris P (2020) Machine learning in cardiovascular flows modeling: predicting arterial blood pressure from non-invasive 4d flow mri data using physics-informed neural networks. *Comput Methods Appl Mech Eng* 358:112623
56. Dantzig JA, Rappaz M (2016) *Solidification: revised & expanded*. EPFL Press, Lausanne
57. Khan P, Debroy T (1984) Alloying element vaporization and weld pool temperature during laser welding of alsi 202 stainless steel. *Metall Trans B* 15(4):641–644
58. Collur M, Paul A, Debroy T (1987) Mechanism of alloying element vaporization during laser welding. *Metall Trans B* 18(4):733–740
59. Voller V, Swaminathan C (1991) Eral source-based method for solidification phase change. *Numer Heat Transf Part B Fundam* 19(2):175–189
60. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi F (2017) A survey of deep neural network architectures and their applications. *Neurocomputing* 234:11–26
61. Schwing AG, Urtasun R Fully connected deep structured networks. [arXiv:1503.02351](https://arxiv.org/abs/1503.02351)
62. Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural-network approach. *IEEE Trans Neural Netw* 8(1):98–113
63. Mikolov T, Karafiát M, Burget L, Černocký J, Khudanpur S (2010) Recurrent neural network based language model. In: Eleventh annual conference of the international speech communication association
64. Sengupta N, Sahidullah M, Saha G (2016) Lung sound classification using cepstral-based statistical features. *Comput Biol Med* 75:118–129
65. Bishop CM (2006) *Pattern recognition and machine learning*. Springer, Berlin
66. Choy CB, Xu D, Gwak J, Chen K, Savarese S (2016) 3d-r2n2: a unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. Springer, pp 628–644
67. Han J, Pei J, Kamber M (2011) *Data mining: concepts and techniques*. Elsevier, Amsterdam
68. Liang S, Srikant R Why deep neural networks for function approximation? [arXiv:1610.04161](https://arxiv.org/abs/1610.04161)
69. Sibi P, Jones SA, Siddarth P (2013) Analysis of different activation functions using back propagation neural networks. *J Theor Appl Inf Technol* 47(3):1264–1268
70. Maas A, Hannun A, Ng A (2013) Rectifier nonlinearities improve neural network acoustic models. In: Proceedings ICML, vol 30, p 3
71. Eger S, Youssef P, Gurevych I Is it time to swish? comparing deep learning activation functions across nlp tasks. [arXiv:1901.02671](https://arxiv.org/abs/1901.02671)
72. Ruder S An overview of gradient descent optimization algorithms. [arXiv:1609.04747](https://arxiv.org/abs/1609.04747)
73. Kingma DP, Ba J Adam: a method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
74. Baydin A, Pearlmutter B, Radul A, Siskind J (2017) Automatic differentiation in machine learning: a survey. *J Mach Learn Res* 18(1):5595–5637
75. Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. *Adv Appl Mech* 28:1–44. [https://doi.org/10.1016/S0065-2156\(08\)70153-4](https://doi.org/10.1016/S0065-2156(08)70153-4)
76. Takizawa K, Tezduyar TE (2011) Multiscale space-time fluid-structure interaction techniques. *Comput Mech* 48:247–267. <https://doi.org/10.1007/s00466-011-0571-z>
77. Takizawa K, Tezduyar TE (2012) Space-time fluid-structure interaction methods. *Math Models Methods Appl Sci* 22(supp02):1230001. <https://doi.org/10.1142/S0218202512300013>
78. Takizawa K, Tezduyar TE, Kuraishi T (2015) Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. *Math Models Methods Appl Sci* 25:2227–2255. <https://doi.org/10.1142/S0218202515400072>
79. Saad Y, Schultz MH (1986) Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Stat Comput* 7(3):856–869
80. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Comput Mech* 43:3–37
81. Takizawa K, Bazilevs Y, Tezduyar TE (2012) Space-time and ALE-VMS techniques for patient-specific cardiovascular fluid-structure interaction modeling. *Arch Comput Methods Eng* 19:171–225. <https://doi.org/10.1007/s11831-012-9071-3>
82. Masud A, Calderer R (2009) A variational multiscale stabilized formulation for the incompressible Navier–Stokes equations. *Comput Mech* 44(2):145–160
83. Zhu L, Goraya S, Masud A (2019) Interface-capturing method for free-surface plunging and breaking waves. *J Eng Mech* 145(11):04019088

84. Calderer R, Zhu L, Gibson R, Masud A (2015) Residual-based turbulence models and arbitrary Lagrangian–Eulerian framework for free surface flows. *Math Models Methods Appl Sci* 25(12):2287–2317
85. Masud A, Calderer R (2013) Residual-based turbulence models for moving boundary flows: hierarchical application of variational multiscale method and three-level scale separation. *Int J Numer Meth Fluids* 73(3):284–305
86. Takizawa K, Tezduyar TE (2012) Computational methods for parachute fluid–structure interactions. *Arch Comput Methods Eng* 19:125–169. <https://doi.org/10.1007/s11831-012-9070-4>
87. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Computational fluid–structure interaction: methods and applications. Wiley, London. <https://doi.org/10.1002/9781118483565>
88. Takizawa K, Fritze M, Montes D, Spielman T, Tezduyar TE (2012) Fluid–structure interaction modeling of ringsail parachutes with disreefing and modified geometric porosity. *Comput Mech* 50:835–854. <https://doi.org/10.1007/s00466-012-0761-3>
89. Takizawa K, Tezduyar TE, Boben J, Kostov N, Boswell C, Buscher A (2013) Fluid–structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity. *Comput Mech* 52:1351–1364. <https://doi.org/10.1007/s00466-013-0880-5>
90. Takizawa K, Tezduyar TE, Boswell C, Tsutsui Y, Montel K (2015) Special methods for aerodynamic-moment calculations from parachute FSI modeling. *Comput Mech* 55:1059–1069. <https://doi.org/10.1007/s00466-014-1074-5>
91. Takizawa K, Bazilevs Y, Tezduyar TE, Korobenko A (2020) Computational flow analysis in aerospace, energy and transportation technologies with the variational multiscale methods. *J Adv Eng Comput* 4(2):83–117
92. Ravensbergen M, Helgedagsrud T, Bazilevs YY, Korobenko A (2020) A variational multiscale framework for atmospheric turbulent flows over complex environmental terrains. *Comput Methods Appl Mech Eng* 368:113182
93. Bazilevs Y, Hsu M-C, Akkerman I, Wright S, Takizawa K, Henicke B, Spielman T, Tezduyar TE (2011) 3D simulation of wind turbine rotors at full scale. Part I: geometry modeling and aerodynamics. *Int J Numer Methods Fluids* 65:207–235. <https://doi.org/10.1002/flid.2400>
94. Takizawa K, Henicke B, Tezduyar TE, Hsu M-C, Bazilevs Y (2011) Stabilized space-time computation of wind-turbine rotor aerodynamics. *Comput Mech* 48:333–344. <https://doi.org/10.1007/s00466-011-0589-2>
95. Takizawa K, Henicke B, Montes D, Tezduyar TE, Hsu M-C, Bazilevs Y (2011) Numerical-performance studies for the stabilized space-time computation of wind-turbine rotor aerodynamics. *Comput Mech* 48:647–657. <https://doi.org/10.1007/s00466-011-0614-5>
96. Takizawa K, Tezduyar TE, McIntyre S, Kostov N, Kolesar R, Habluetzel C (2014) Space-time VMS computation of wind-turbine rotor and tower aerodynamics. *Comput Mech* 53:1–15. <https://doi.org/10.1007/s00466-013-0888-x>
97. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C, Øiseth O, Mathisen KM, Kostov N, McIntyre S (2014) Engineering analysis and design with ALE-VMS and space-time methods. *Arch Comput Methods Eng* 21:481–508. <https://doi.org/10.1007/s11831-014-9113-0>
98. Takizawa K (2014) Computational engineering analysis with the new-generation space-time methods. *Comput Mech* 54:193–211. <https://doi.org/10.1007/s00466-014-0999-z>
99. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Kostov N, McIntyre S (2014) Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. *Arch Comput Methods Eng* 21:359–398. <https://doi.org/10.1007/s11831-014-9119-7>
100. Takizawa K, Tezduyar TE, Mochizuki H, Hattori H, Mei S, Pan L, Montel K (2015) Space-time VMS method for flow computations with slip interfaces (ST-SI). *Math Models Methods Appl Sci* 25:2377–2406. <https://doi.org/10.1142/S0218202515400126>
101. Korobenko A, Bazilevs Y, Takizawa K, Tezduyar TE (2018) Recent advances in ALE-VMS and ST-VMS computational aerodynamic and FSI analysis of wind turbines. In: Tezduyar TE (ed) *Frontiers in computational fluid–structure interaction and flow simulation: research from lead investigators under forty–2018, modeling and simulation in science, engineering and technology*. Springer, Berlin, pp 253–336. [https://doi.org/10.1007/978-3-319-96469-0\\_7](https://doi.org/10.1007/978-3-319-96469-0_7)
102. Otoguro Y, Mochizuki H, Takizawa K, Tezduyar T (2020) Space-time variational multiscale isogeometric analysis of a tsunami-shelter vertical-axis wind turbine. *Comput Mech* 66(6):1443–1460
103. Ravensbergen M, Mohamed A, Korobenko A (2020) The actuator line method for wind turbine modelling applied in a variational multiscale framework. *Comput Fluids* 201:104465
104. Mohamed A, Bear C, Bear M, Korobenko A (2020) Performance analysis of two vertical-axis hydrokinetic turbines using variational multiscale method. *Comput Fluids* 200:104432
105. Bayram A, Korobenko A (2020) Variational multiscale framework for cavitating flows. *Comput Mech* 66:49–67
106. Takizawa K, Tezduyar TE, Buscher A, Asada S (2014) Space-time fluid mechanics computation of heart valve models. *Comput Mech* 54(4):973–986
107. Terahara T, Takizawa K, Tezduyar T, Bazilevs Y, Hsu M (2020) Heart valve isogeometric sequentially-coupled fsi analysis with the space-time topology change method. *Comput Mech* 65:1167–1187
108. Terahara T, Takizawa K, Tezduyar T, Tsushima A, Shiozaki K (2020) Ventricle-valve-aorta flow analysis with the space-time isogeometric discretization and topology change. *Comput Mech* 65:1343–1363
109. Bazilevs Y, Takizawa K, Wu M, Kuraishi T, Avsar R, Xu Z, Tezduyar T (2020) Gas turbine computational flow and structure analysis with isogeometric discretization and a complex-geometry mesh generation method. *Comput Mech*. <https://doi.org/10.1007/s00466-020-01919-w>
110. Bazilevs Y, Takizawa K, Tezduyar T, Hsu M, Otoguro Y, Mochizuki H, Wu M (2020) Wind turbine and turbomachinery computational analysis with the ale and space-time variational multiscale methods and isogeometric discretization. *J Adv Eng Comput* 4(1):1–32
111. Kozak N, Rajanna M, Wu M, Murugan M, Bravo L, Ghoshal A, Hsu M, Bazilevs Y (2020) Optimizing gas turbine performance using the surrogate management framework and high-fidelity flow modeling. *Energies* 13(17):4283
112. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Avsar R, Zhang Y (2019) Space-time vms flow analysis of a turbocharger turbine with isogeometric discretization: computations with time-dependent and steady-inflow representations of the intake/exhaust cycle. *Comput Mech* 64(5):1403–1419
113. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Mei S (2019) Turbocharger turbine and exhaust manifold flow computation with the space-time variational multiscale method and isogeometric analysis. *Comput Fluids* 179:764–776. <https://doi.org/10.1016/j.compfluid.2018.05.019>
114. Kuraishi T, Takizawa K, Tezduyar T (2019) Space-time computational analysis of tire aerodynamics with actual geometry, road contact, tire deformation, road roughness and fluid film. *Comput Mech* 64(6):1699–1718
115. Kuraishi T, Takizawa K, Tezduyar TE (2019) Tire aerodynamics with actual tire geometry, road contact and tire deformation.



- Comput Mech 63:1165–1185. <https://doi.org/10.1007/s00466-018-1642-1>
116. Levine L, Lane B, Heigel J, Migler K, Stoudt M, Phan T, Ricker R, Strantza M, Hill M, Zhang F, Seppala J, Garboczi E, Bain E, Cole D, Allen A, Fox J, Campbell C (2020) Outcomes and conclusions from the 2018 am-bench measurements, challenge problems, modeling submissions, and conference. *Integr Mater Manuf Innov* 9(1):1–15
117. Heigel J, Lane B, Levine L (2020) In situ measurements of melt-pool length and cooling rate during 3d builds of the metal am-bench artifacts. *Integr Mater Manuf Innov* 9(1):31–53
118. Brandon L, Jarred H, Richard R, Ivan Z, Vladimir K, Jordan W, Thien P, Mark S, Sergey M, Lyle L (2020) Measurements of melt pool geometry and cooling rates of individual laser traces on in625 bare plates. *Integr Mater Manuf Innov* 9:16–30
119. Heigel J, Lane B (2018) Measurement of the melt pool length during single scan tracks in a commercial laser powder bed fusion process. *J Manuf Sci Eng* 140(5):5–12

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.