Fast heat transfer simulation for laser powder bed fusion

Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 412 (2023) 116107

Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Fast heat transfer simulation for laser powder bed fusion

Xiaohan Li[*], Nick Polydorides

Institute for Digital Communications, School of Engineering, University of Edinburgh, King's Buildings Campus, W Mains Rd, Edinburgh, EH9 3JW, UK

Received 16 January 2023; received in revised form 30 April 2023; accepted 2 May 2023
Available online 24 May 2023

Abstract

Accurate and fast modeling of the temperature distribution and phase transitions in laser powder bed fusion is a major milestone in achieving its quality assurance. Commonly referred to as digital twin technology, the goal is to find agile, fast-to-compute but also sufficiently accurate simulators that can replicate the 3D printing process while enhancing the quality of its outcomes. In this work, we propose a surrogate model for the nonlinear heat transfer equation coupled with subspace projection and randomized sketching, that exploits the accuracy and explainability of finite element time-domain simulation with the computational efficiency of Monte Carlo sampling, applied to the modality of laser powder bed fusion. Focusing on tackling the high-dimensionality imparted from the finite element approximation and the nonlinearity in the governing equations, our surrogate relies on low-dimensional projection with subspace selection and subsequently sub-samples the Picard iterations utilized to solve the projected non-linear system of equations. The projection bases are generated in the process of simulation by combining previous temperature profiles and locally deployed anisotropic Gaussian functions, while the sketching process utilizes efficient sampling without replacement based on approximate optimal sampling distributions. Both the projection and the sketching are designed to implement alongside the printing process, which makes the proposed surrogate capable of handling different process parameters without requiring prior computations offline. A series of numerical experiments are presented to validate the surrogate's accuracy and reduction in compute time compared to high-fidelity finite element simulations. Although the achieved speed-up can be as a high ten, computational times are still orders of magnitude away from what would be required for real-time computations. The presented methodology allows to handle different printing attributes (laser power and scan speed) and arbitrary thermal conductivity anisotropy.

Keywords: Dimensionality reduction; Finite element method; Nonlinear systems of equations; Randomized sketching; Transient heat transfer with phase changes

1. Introduction

As a heating-based additive manufacturing technology, laser powder bed fusion (LPBF) is aimed at printing metallic parts layer after layer guided by 3D geometrical designs. As such it can, in principle, perform better and more efficiently than traditional subtractive manufacturing in fabricating complex designs and reducing production costs in materials and time [1]. The process productivity and quality assurance, however, impede LPBF from

[*] Corresponding author.
E-mail addresses: xiaohan.li@ed.ac.uk (X. Li), n.polydorides@ed.ac.uk (N. Polydorides).

https://doi.org/10.1016/j.cma.2023.116107

realizing its full potential. One way to tackle these issues is by developing a digital twin (DT) of the printing process as a way to optimize the printing and enable closed-loop control that will in turn improve repeatability and efficiency while reducing waste and the need for human intervention. The concept is particularly appealing for a highly digitized process like LPBF, while its utility relies heavily on the speed and accuracy of surrogate and reduced models that capture the governing phenomena [2,3]. In particular, the efficient and accurate temperature profile near the heating source and the dimensions of the induced melt pools are critical for deducing the micro-structure and residual stresses of the part and can predict the final part's deformation, fatigue life, and other metallurgical properties [4,5]. Although high-fidelity numerical simulators for transient heat transfer with phase changes exist, mostly in the form of finite element time domain methods, running these models is very time-consuming due to the high dimensionality imparted by the fine spatial–temporal discretization as well as the intrinsic nonlinearity associated with the phase changes incurred. These technical challenges motivate the design of low-dimensional fast surrogates offering a prudent trade-off between time efficiency and model accuracy.

To achieve real-time implementation, various machine learning methods were proposed for constructing such surrogates in the context of LPBF. Mozaffar et al. proposed a recurrent neural network for the directed energy deposition process, capable of predicting temperature profiles for different geometries [6]. The approach, however, incurs a large cost in storage and compute time, due to the exorbitant amount of data required to train the model. Roy et al. predicted thermal histories from different part sizes with deep neural networks while using the design of heat influence zone to reduce the number of data [7]. In [8], a reduced Gaussian process surrogate was employed to predict high-dimensional temperatures by linear combination of temperatures predicted to be closed to the final prediction. Another approach to reduce compute times is that of model order reduction (MOR) categorized as structural, data-driven, and projection-based. An adaptive mesh refinement strategy was proposed in [9] as a structural MOR method where the area of the computational domain spanning the volume of the part was attributed a finer mesh and the remainder was more coarsely meshed. In [10], a convolutional autoencoder was applied as a data-driven MOR method to find the encoder and decoder for high-dimensional results of partial differential equations (PDEs). The surrogate was formed by cascading a feed-forward network and the decoder where the feed-forward network mapped training inputs to encoded low-dimensional representations. The projection-based method projects the full order model into the reduced model using orthonormal bases found via proper orthogonal decomposition (POD) [11], moment matching [12], or balanced truncation [13]. Note that although projection-based MOR works well in linear PDEs, a further approximation of nonlinearity like the discrete empirical interpolation method in [14] is additionally required for nonlinear PDEs solved by either Picard or Newton iteration. In [8], the full-order model was projected by data-driven local projection bases formed by selecting training temperatures deemed to be similar to the final result, and randomized sketching was directly applied to subsample the algebraic operations involved in solving the nonlinear model with Picard iterations. Though it was a projection-based approach, it required the same effort of data generation and training as a data-driven approach since its temperature selection relied on a data-driven distance predictor. In general, though a data-driven surrogate realizes real-time implementations, it is an exhaustive mapping highly relying on sufficient training data to yield an accurate response surface. It is challenging in finding representative data and expensive in storage and offline computation. The projection-based method, however, needs a valid projection basis with as little dimension as possible and an additional design to effectively reduce nonlinear computations. These two issues are addressed explicitly in this paper.

The thermal model of LPBF simulates the heat transfer of a laser beam following a specific scanning path, which yields melt pools with time-variant locations and has temperature distributions peaking at the laser beam centers. Inspired by the general shape of temperature distributions and melt pools, we design time-variant local projection bases by a set of 3D Gaussian functions and previous temperatures. The selected Gaussian functions are scaled and translated from a benchmark Gaussian function calibrated at each time step. Nevertheless, the number of nonlinear computations is not reduced after projection. To cut down this effort, we apply randomized sketching with approximate sampling probability to estimate the projected model with less sketching time. As the surrogate is generated and calibrated based on previous temperatures, for different printing parameters and anisotropy with some randomness the surrogate can still be feasible without altering its design. Both the projection and sketching processes are implemented in the process of simulation and are updated at each time step. Together with the additional time cost due to basis generation, sketching, and projection, the designed surrogate as a swift numerical solver with finite element method (FEM) still only needs 11.33% of the original model's execution time while keeping 99.62% of tests give temperature predictions with relative errors below 3%. With this performance, the surrogate is a reasonable choice for the DT framework of LPBF and is valuable in promoting quality assurance through process optimization and closed-loop control.

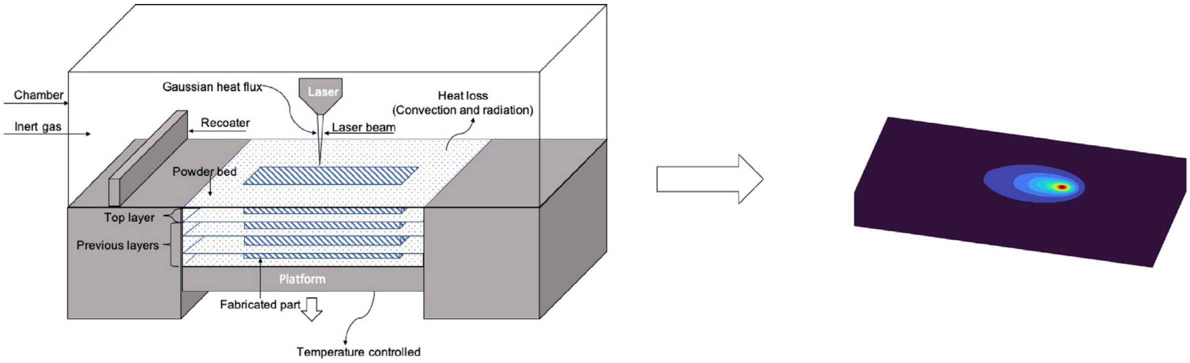


Fig. 1. Left, a schematic depicting the apparatus of a LPBF printing process and to the right a typical profile of the temperature distribution in the part when heated with a Gaussian heat source.

1.1. Notation and paper organization

In this paper, the matrices are presented in capital letters. For example, we denote a matrix as X, the ith row, the ith column, and the (i, j)-th entry of which are respectively expressed as X_{i*}, X_{*i}, and X_{ij}. Vectors that are not rows or columns of a matrix are in lowercase letters, with the ith entry of x denoted as x_i. Continuous quantities are also specified in lower case letters and tensors are denoted as $\vec{x}$.

In the next section we present the governing principles of the nonlinear, anisotropic thermal model in LPBF followed by its numerical implementation using FEM and the analysis of what makes the solver time-consuming. In Section 3, the details of the surrogate are explained with two subsections respectively illustrating the designs for sketching and projection. Then, the result section includes a simplified example with its codes provided in a Github repository and several numerical experiments in LPBF showing the performance in model accuracy and time cost reduction. Finally, the conclusions are laid out in Section 5.

2. Thermal model of LPBF

In LPBF, a trajectory is predetermined according to the object we want to print. Following this trajectory, a laser beam selectively scans the top surface of the powder bed yielding melt pools along this trajectory. The melt pools throughout the printing process fuse with their adjacent solidified material, and the final object will eventually be formed by layer-after-layer printing. The schematic of printing a cuboid object and the temperature distribution during printing are descriptively shown in Fig. 1.

2.1. Governing equations

Heat transfer of a t_N seconds scanning process is modeled as a nonlinear heat equation over a 3D computational domain $\Omega \in \mathbb{R}^3$, which is

$$\rho(u)c(u)\frac{\partial u(x,t)}{\partial t} - \nabla \cdot \vec{\kappa}(u)\nabla u(x,t) = 0, \quad (x,t) \quad \text{in } \Omega \times [0, t_N], \tag{1}$$

where $u(x, t)$ is the temperature at time t with 3D Cartesian coordinates $x = [x_1, x_2, x_3]^T$. The thermal properties of the material including thermal conductivity $\vec{\kappa}$, density ρ, and specific heat capacity c are temperature-dependent [15]. While experimental data of these three thermal properties in [16–20] are fitted with polynomials, the latent heat effect during the liquid–solid phase transition is considered in the model of specific heat capacity c, all of which are detailed in appendix A in [8]. The anisotropy due to the intricate melt pool convection, however, is simplified as an anisotropic enhanced thermal conductivity model which specifies the divergence term in Eq. (1) as

$$\nabla \cdot \vec{\kappa}(u)\nabla u := \sum_{i,j=1}^{3} \frac{\partial}{\partial x_i}\left[\kappa_{ij}(u)\right]\frac{\partial u}{\partial x_j} \tag{2}$$

where κ_{ii} satisfy $\kappa_{11}(u) : \kappa_{22}(u) : \kappa_{33}(u) = \lambda_1 : \lambda_2 : \lambda_3$, for some positive weights $\lambda \in \mathbb{R}^3$ while $\kappa_{ij} = 0$ when $i \neq j$ [15,21]. The vector λ can be considered as time invariant to be estimated from experimental data as in [16], or it may allow to vary randomly in a range which resembles more accurately the real process [22,23]. The boundary of Ω denoted as Γ is partitioned into three parts: the top surface Γ_t, the side surface Γ_s that is normal to the printing plane, and the bottom surface Γ_b. For these we have a Neumann boundary condition relating the heat source q, to the heat losses due to convection q_c, and radiation q_r as

$$\vec{\kappa}(u)\nabla u(x,t)\cdot\hat{n} = q(x,t) - q_c(u) - q_r(u), \quad x \text{ on } \Gamma_t \cup \Gamma_s, \tag{3}$$

where $\hat{n}$ is the outward unit normal on the boundary surface. The heat source function at the boundary $q(x,t)$ models the cross section of a Gaussian laser beam as it enters the top surface of the domain

$$q(x,t) = \frac{2aP}{\pi\varsigma^2}\exp\Big(-\frac{2(\|x-\mu(t)\|_2)^2}{\varsigma^2}\Big), \quad x \text{ on } \Gamma_t, \tag{4}$$

where $\mu(t) \in \mathbb{R}^3$ is the spatial coordinate of the laser beam center at time t, a is the absorptivity, P is the laser power, and ς is the effective radius of the laser beam [16]. For the heat losses due to convection and radiation we consider the nonlinear models

$$q_c(u) = h(u-u_a), \quad q_r(u) = \sigma_s\varepsilon(u^4 - u_a^4), \quad x \text{ on } \Gamma_t \cup \Gamma_s, \tag{5}$$

where $h > 0$, σ_s, ε, and u_a are respectively the heat convection coefficient, the Stefan–Boltzmann constant, the emissivity, and the ambient temperature, all of which are regarded as constants according to [24,25]. The bottom surface of the part is kept at a constant temperature via a temperature-controlled platform, thus we impose a Dirichlet condition there as [26]

$$u(x,t) = u_b, \quad x \text{ on } \Gamma_b, \ t \in [0, t_N]. \tag{6}$$

In effect, an initial condition

$$u(x,0) = u_0, \quad x \text{ in } \Omega, \tag{7}$$

suffices to yield a unique temperature solution $u(x,t)$ to simulate the heat transfer over the spatial and temporal domain.

2.2. The full-order numerical solver with FEM

A high-fidelity numerical solver of the nonlinear thermal model as Eqs. (1)–(7) is developed using the FEM where fine discretization is applied in both temporal and spatial domains. While a t_N seconds printing process is discretized as N time steps with an interval Δt namely $t_n = n\Delta t$ for $n = 0, 1, \dots, N$, the spatial domain Ω is discretized into n_e linear tetrahedrons and a large number n_d degrees of freedom (DoF). The temperature at time t_n is represented by the high-dimensional vector $u_n \in \mathbb{R}^{n_d}$. Given u_{n-1}, the temperature at the next time step u_n satisfies

$$A(u_n)u_n = b(u_{n-1}, u_n), \quad \text{for} \quad n = 1, \dots, N, \tag{8}$$

where the temperature-dependent matrix $A \in \mathbb{R}^{n_d\times n_d}$ and vector $b \in \mathbb{R}^{n_d}$ are defined in Appendix A. The nonlinear system of equations (8) we apply Picard's iterative algorithm which converges to a unique solution subject to some mild assumptions, i.e. for a small Δt, $\|u_n - u_{n-1}\|_2$ is small [27,28].

2.3. Time consumption due to high dimensionality and nonlinearity

The full-order model with FEM is time-consuming since it is high-dimensional and nonlinear. While the high dimensionality is caused by fine spatial discretization, the nonlinearity is intrinsic to the material's thermal properties at the applied heat levels which trigger phase changes, as well as the radiation losses. On the one hand, the high n_d increases the dimension of the (8) system resulting in more compute time per Picard iteration. On the other hand, both the high dimensionality and nonlinearity increase the number of nonlinear computations required. Here we take the mass matrix $M(u_n) \in \mathbb{R}^{n_d \times n_d}$ as an example while summarizing the details of other matrices and vectors in Appendix A. The entries of M are defined as

$$M_{ij}(u_n) = \int_{\Omega} \rho(u_n) c(u_n) \phi_i(x) \phi_j(x) \mathrm{d}x \quad \text{for} \quad i, j = 1, \ldots, n_d, \tag{9}$$

where the integrand is the product of the density function $\rho(u_n)$, the specific heat capacity $c(u_n)$, and the local basis functions $\phi(x)$ of the ith and jth DoF. The integral in Eq. (9) is approximated via a n_m-point Gaussian quadrature rule as

$$\begin{aligned} M_{ij}(u_n) &= \sum_{e=1}^{n_e} \int_{\Omega_e} \rho(u_n) c(u_n)\, \phi_i(x) \phi_j(x)\, \mathrm{d}x, \quad i, j \in \operatorname{supp} \Omega_e \\ &\approx \sum_{e=1}^{n_e} \sum_{\iota=1}^{n_m} |J_e|\, |\Omega_e|\, w_\iota \rho(u_n(x_e^{(\iota)})) c(u_n(x_e^{(\iota)})) \phi_i(x_e^{(\iota)}) \phi_j(x_e^{(\iota)}), \end{aligned} \tag{10}$$

where $x_e^{(\iota)} \in \mathbb{R}^3$ and w_ι are respectively the coordinate and weight of the ι-th integration point within the eth element. $|J_e|$ is the determinant of the Jacobian of the eth element's coordinate transform. $|\Omega_e|$ is the eth element's volume. In matrix form, we can rewrite the approximation of M as

$$M(u_n) \approx \Phi^T W(u_n) \Phi, \tag{11}$$

where $\Phi \in \mathbb{R}^{n_m n_e \times n_d}$ are evaluations of the basis functions at all integration points and the diagonal matrix $W(u_n) \in \mathbb{R}^{n_m n_e \times n_m n_e}$ contains the temperature-dependent entries. Specifically, the ith column of Φ is

$$\Phi_{*i} = \left[\phi_i(X^{(1)}), \ldots, \phi_i(X^{(n_m)})\right]^T, \quad \text{for} \quad i = 1, \ldots, n_d, \tag{12}$$

where we have $\phi_i(X^{(\iota)}) = [\phi_i(x_1^{(\iota)}), \ldots, \phi_i(x_{n_e}^{(\iota)})]$ at the ι-th integration point. The temperature-dependent computations related to the ι-th integration point are held in the diagonal matrix $D_{ee}^n(X^{(\iota)}) = |J_e|\, |\Omega_e|\, w_\iota \rho(u_n(x_e^{(\iota)})) c(u_n(x_e^{(\iota)}))$ for $e = 1, \ldots, n_e$, and thus for all integration points this yields a diagonal

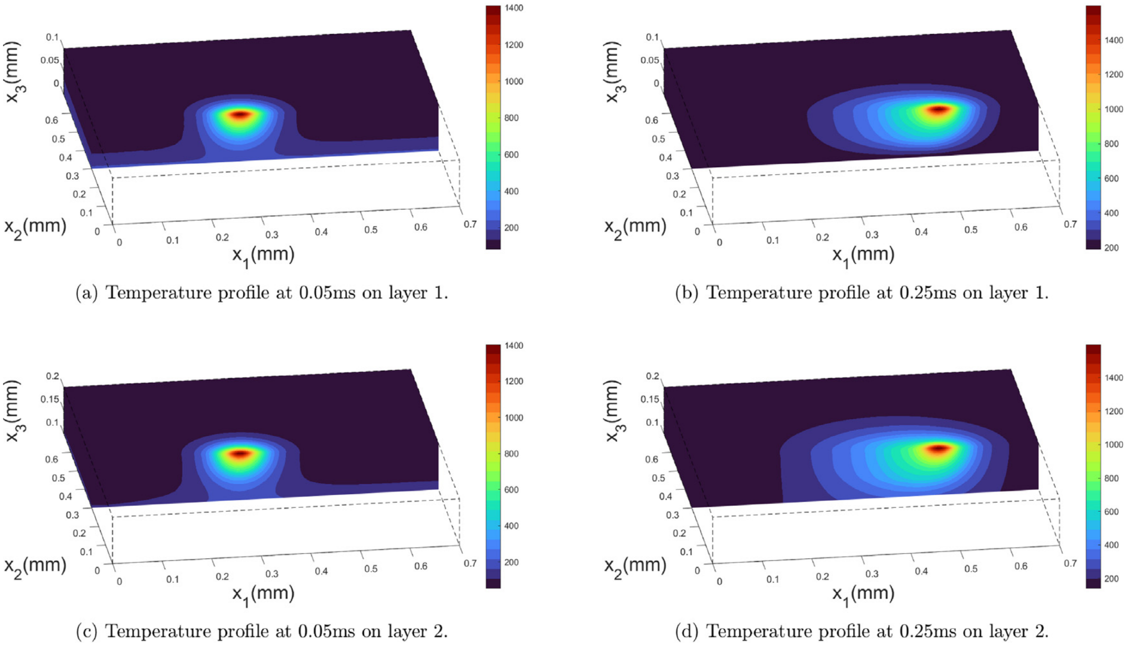
$$W(u_n) = \begin{bmatrix} D^n(X^{(1)}) & & \\ & \ddots & \\ & & D^n(X^{(n_m)}) \end{bmatrix}. \tag{13}$$

From (12) and (13), it is easy to see that the tall matrix Φ does not depend on temperature, and can thus be computed before the start of the simulation. During the simulation, the computation due to nonlinearity is reflected by the diagonal of $W(u_n)$ in each Picard iteration, which requires $n_m n_e$ nonlinear computations in density $\rho(u_n)$ and specific heat capacity $c(u_n)$. The degree of nonlinearity for the integrand is affected by the polynomial functions $\rho(u_n)$ and $c(u_n)$, both of which are fitted with experimental data and vary with different materials. The higher the degree of nonlinearity is the larger n_m is, and the finer the spatial discretization is the larger n_e becomes. Accordingly, it is worthwhile to figure out a way that manages to reduce both the dimensionality and nonlinearity of the thermal model as Eq. (8) but maintains the accuracy of temperature profiles.

3. Fast-computed surrogate

At the start of a new line of scanning in LPBF, the profile of the temperature is not regular for the first few time steps. This is partly due to the temperature-controlled building platform and the existing temperature from previous printing. More importantly, with a small time interval, there is not enough heat at the beginning to form steady melt pools. For example, a thermal simulation of scanning a straight line is shown in Fig. 2, from which we can tell that the steady temperature distributions are gradually formed with obvious anisotropy. Therefore, to improve time efficiency while securing accuracy, we first run the full order model for the first n_t time steps and then replace the subsequent thermal simulation with a surrogate. The surrogate is proposed to reduce the expensive time cost due to high dimensionality and nonlinearity. To begin with, we project the full-order thermal model with FEM to a comparably low dimension with a properly established orthonormal projection basis $\Psi_n \in \mathbb{R}^{n_d \times n_r}$ where $n_r \ll n_d$ and $u_n \approx \Psi_n r_n$. The Eq. (8) is thereby projected as

$$\mathbf{A}(u_n) r_n = \mathbf{b}(u_{n-1}, u_n), \tag{14}$$



(a) Temperature profile at 0.05ms on layer 1.

(b) Temperature profile at 0.25ms on layer 1.

(c) Temperature profile at 0.05ms on layer 2.

(d) Temperature profile at 0.25ms on layer 2.

Fig. 2. Temperature profiles whilst printing on a straight line trajectory. Notice that the temperature profiles at the start of the printing on each layer (left column) cannot be described as (anisotropic) Gaussian functions.

where $\mathbf{A}(u_n) = \Psi_n^T A(u_n) \Psi_n$ and $\mathbf{b}(u_{n-1}, u_n) = \Psi_n^T b(u_{n-1}, u_n)$. We hereafter denote the projected matrices and vectors with boldface. Though projection reduces the dimension of the matrix equation, the amount of nonlinear computation is still consistent with the full-order model. Accordingly, we further approximate Eq. (14) with randomized sketching to yield a swift surrogate as

$$\hat{\mathbf{A}}(\hat{u}_n)\hat{r}_n = \hat{\mathbf{b}}(\hat{u}_n, \hat{u}_{n-1}), \tag{15}$$

and get the projected solution $\hat{r}_n \in \mathbb{R}^{n_r}$ via Picard iterations. Backprojecting to the n_d dimension we then reconstruct $\hat{u}_n = \Psi_n \hat{r}_n$. The procedure of implementing the surrogate is outlined in algorithm 1 where both sketching and projection procedures have several parameters to control the trade-off between model accuracy and time cost. From algorithm 1, it is shown that we need to generate the projection basis before randomized sketching while the actual projection happens after the settlement of row selection in randomized sketching. To state the design clearly, we start from the assumption that we have a proper projection basis and illustrate the randomized sketching with approximated success probability in Section 3.1. Then, we explain Gaussian basis generation and projection in Section 3.2.

3.1. Randomized sketching with approximate sampling probability

We now explain the design by focusing on the mass matrix $M(u_n)$ defined in Eq. (9) as an example since the randomized sketching and projection of each part of the full-order model are implemented in a very similar way. Herein, we generally show the computational procedure of the projected and sketched mass matrix $\hat{\mathbf{M}}(u_n)$ as Fig. 3 where the modules related to randomized sketching are illustrated in detail and the module of Gaussian local projection is enriched later in Section 3.2. In the reduced model with a projection basis Ψ_n, the projected mass matrix $\mathbf{M}(u_n) \in \mathbb{R}^{n_r \times n_r}$ is

$$\mathbf{M}(u_n) = \Psi_n^T M(u_n) \Psi_n = \mathbf{\Phi}^T W(u_n) \mathbf{\Phi} = \sum_{i=1}^{n_m n_e} W_{ii}(u_n) \mathbf{\Phi}_{i*}^T \mathbf{\Phi}_{i*}, \tag{16}$$

where the tall matrix $\mathbf{\Phi} \in \mathbb{R}^{n_m n_e \times n_r}$ is the projected basis function matrix $\Phi \Psi_n$. We also tell that $\mathbf{M}(u_n)$ can be represented as the linear combination of $n_m n_e$ rank-one matrices $\mathbf{\Phi}_{i*}^T \mathbf{\Phi}_{i*}$ weighted by $W_{ii}(u_n)$ for $i = 1, \ldots, n_m n_e$.

Algorithm 1 The fast-computed surrogate.

```
Input: a total amount of time steps $N$, the full-order model as Eq. (8), an integer $n_t$.
Output: the temperature profiles $u_n$ for $n = 1, \cdots, N$.
for $n = 1$ to $N$ do
    if $n \leqslant n_t$ then
        Get $u_n$ with the full-order model $A(u_n)u_n = b(u_{n-1}, u_n)$.
    else
        Generate the Gaussian local projection basis $\Psi_n$ as Section 3.2.1.
        Implement randomized sketching with approximated sampling probability as Section 3.1.
        Implement projection as Section 3.2.2.
        Get $\hat{r}_n$ by solving the projected and sketched model $\hat{\mathbf{A}}(\hat{u}_n)\hat{r}_n = \hat{\mathbf{b}}(\hat{u}_n, \hat{u}_{n-1})$.
        Reconstruct the temperature estimation $\hat{u}_n = \Psi_n \hat{r}_n$.
    end if
end for
```

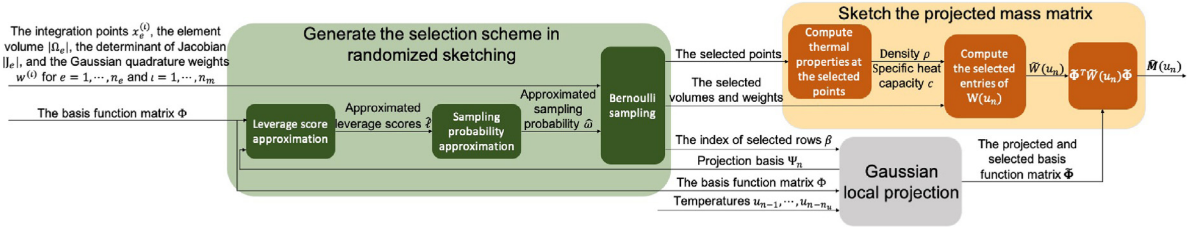

The integration points $x_e^{(\iota)}$, the element volume $|\Omega_e|$, the determinant of Jacobian $|J_e|$, and the Gaussian quadrature weights $w^{(\iota)}$ for $e = 1, \cdots, n_e$ and $\iota = 1, \cdots, n_m$
The basis function matrix Φ
Generate the selection scheme in randomized sketching
Leverage score approximation
Approximated leverage scores $\hat{\ell}$
Sampling probability approximation
Approximated sampling probability $\hat{\omega}$
Bernoulli sampling
The selected points
The selected volumes and weights
Sketch the projected mass matrix
Compute thermal properties at the selected points
Density ρ
Specific heat capacity c
Compute the selected entries of W(u_n)
$\tilde{W}(u_n)$
$\tilde{\Phi}^T\tilde{W}(u_n)\tilde{\Phi}$
$\hat{M}(u_n)$
The index of selected rows β
Projection basis Ψ_n
The basis function matrix Φ
Temperatures $u_{n-1}, \cdots, u_{n-n_u}$
Gaussian local projection
The projected and selected basis function matrix $\tilde{\Phi}$

Fig. 3. The sketching procedure to compute the projected and sketched mass matrix.

The time cost due to nonlinearity, however, is reflected by the diagonal matrix $W(u_n)$. It is shown in Eq. (16) that projection only reduces the columns of basis function matrix from n_d to n_r while all $n_m n_e$ evaluations of temperature-dependent functions to form the diagonal of $W(u_n)$ still remain, to bypass the majority of which we further approximate the projected model via randomized sketching. It manages to approximate $\mathbf{M}(u_n)$ by selecting and weighting some of the rows of $\mathbf{\Phi}$ and $W(u_n)$ based on some non-uniform Bernoulli sampling probabilities. In specific, the approximated mass matrix $\hat{\mathbf{M}}(u_n) \in \mathbb{R}^{n_r \times n_r}$ is

$$\hat{\mathbf{M}}(u_n) = \tilde{\mathbf{\Phi}}^T \tilde{W}(u_n)\tilde{\mathbf{\Phi}} = \sum_{i=1}^{n_g} \tilde{W}_{ii} \tilde{\mathbf{\Phi}}_{i*}^T \tilde{\mathbf{\Phi}}_{i*}, \tag{17}$$

where $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{n_g \times n_r}$ only contains the n_g selected rows of $\mathbf{\Phi}$. The total number of both rank-one matrices and corresponding weights are reduced from $n_m n_e$ to n_g. When the ith row of $\tilde{\mathbf{\Phi}}$ weighted by $\tilde{W}_{ii}(u_n)$ is selected from the jth row of $\mathbf{\Phi}$ weighted by $W_{jj}(u_n)$, we have the diagonal entries of $\tilde{W}(u_n) \in \mathbb{R}^{n_g \times n_g}$ as

$$\tilde{W}_{ii}(u_n) = \frac{1}{\omega_j} W_{jj}(u_n), \quad \text{for} \quad i = 1, \dots, n_g, \tag{18}$$

where $0 < \omega_j \leqslant 1$ is the probability of selecting the jth row of $\mathbf{\Phi}$. All n_g indexes j in Eq. (18) are recorded as a row index vector $\beta \in \mathbb{R}^{n_g}$ which is useful in the fast projection explained in Section 3.2.2. In randomized sketching stated above, it is essential to set all sampling probabilities $\omega \in \mathbb{R}^{n_m n_e}$ properly to ensure low sketching error while cutting down sketching time cost. Accordingly, we specify ω based on the fast-computed approximation of leverage scores which is detailed in the subsequent Section 3.1.1.

3.1.1. Approximate sampling probability

To significantly reduce nonlinear computations, it is critical to properly set the success probabilities $\omega \in \mathbb{R}^{n_m n_e}$ so that we manage to retain $n_g \ll n_m n_e$ and a small sketching error $\|\mathbf{M}(u_n) - \hat{\mathbf{M}}(u_n)\|_2$. The vector of probabilities

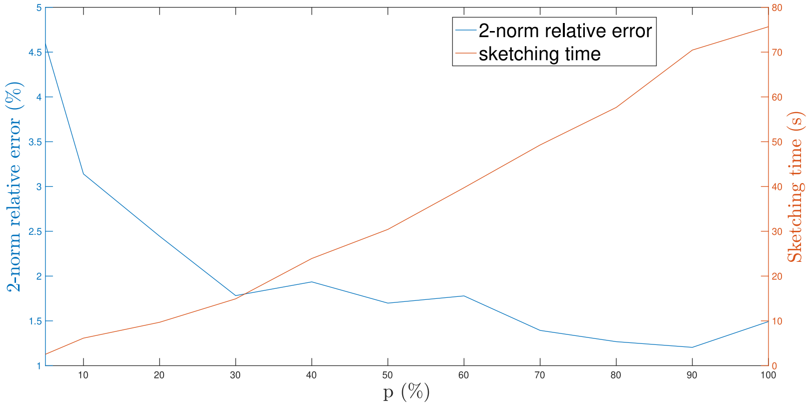


Fig. 4. The relative errors with different percentage p.

ω depends on the leverage scores of the rows of $\mathbf{\Phi} \in \mathbb{R}^{n_m n_e \times n_r}$ [29,30]. Namely, for $i = 1, \ldots, n_m n_e$ we have

$$\omega_i = \min(1, \xi / n_r \ell_i), \tag{19}$$

$$\ell_i = (\|U_{i*}\|_2)^2, \tag{20}$$

where the statistical leverage score $\ell \in \mathbb{R}^{n_m n_e}$ is the squared row norms of U, and $U \in \mathbb{R}^{n_m n_e \times n_r}$ is the left singular vectors in the compact SVD of $\mathbf{\Phi}$. The parameter $\xi > 0$ is a positive integer that bounds the number of rows selected. Though the exact leverage scores ℓ are needed for the randomized sketching, they are computationally expensive to obtain. The two most computationally expensive parts are the large matrix multiplication $\mathbf{\Phi} = \Phi \Psi_n$ with the computational complexity $O(n_m n_e n_d n_r)$ and the compact SVD of $\mathbf{\Phi}$. Therefore, we propose a way to approximate ℓ with less time in SVD and without implementing the exact product $\Phi \Psi_n$. As in Eq. (20), the exact leverage scores are computed as the squared row norms of the n_r left singular vectors in U. We hereby replace it with its first $\tilde{n}_r$ ($\tilde{n}_r < n_r$) columns that correspond to the $\tilde{n}_r$ largest singular values of $\mathbf{\Phi}$, and then use the squared row norms of the $\tilde{n}_r$ columns to approximate ℓ. In specific,

$$\hat{\ell}_i = (\|\tilde{U}_{i*}\|_2)^2, \tag{21}$$

where $\tilde{U} \in \mathbb{R}^{n_m n_e \times \tilde{n}_r}$ containing the $\tilde{n}_r$ left singular vectors is obtained from the $\tilde{n}_r$ rank approximation of $\mathbf{\Phi}$. We can choose $\tilde{n}_r$ as a small proportion of n_r without a high sacrifice in accuracy. As shown in Fig. 4, we take $p = \tilde{n}_r / n_r$ indicating the percentage we pick. Then, we compare the 2-norm relative errors of temperatures and the corresponding sketching time with p ranging from 5% to 100%. As in Fig. 4, we can tell that the approximation error has a comparably high tolerance for a small percentage p. The 2-norm relative errors of temperatures are below 2% when $p \geqslant 30\%$, while the increase of accuracy is not significant when $p \geqslant 80\%$. Therefore, an acceptable range of p is from 30% to 80%, which can be further specified according to how much sketching time we want to take. Given a reasonable percentage p, $\tilde{U}$ can be fast-computed via randomized SVD. While the standard procedure of randomized SVD in [31] is detailed by algorithm B.1 in Appendix B, it is modified as algorithm 2 to avoid computing $\mathbf{\Phi} = \Phi \Psi_n$ explicitly. In specific, as stated in the third step of algorithm 2 we compute $\Phi(\Psi \Upsilon_1)$ instead of $\Phi \Psi \Upsilon_1$ to avoid the expensive multiplication $\Phi \Psi$. The computational complexity of $\Phi(\Psi \Upsilon_1)$ and $\Phi \Psi \Upsilon_1$ are respectively $O(n_d n_r \tilde{n}_r + n_m n_e n_d \tilde{n}_r)$ and $O(n_m n_e n_d n_r + n_d n_r \tilde{n}_r)$. Since we have $\tilde{n}_r = \lfloor p n_r \rfloor$ and $1/n_r \leqslant p < 1$, it is validated that $O(n_d n_r \tilde{n}_r + n_m n_e n_d \tilde{n}_r) < O(n_m n_e n_d n_r + n_d n_r \tilde{n}_r)$. Moreover, the smaller the proportion p is the more time we can save by avoiding this large matrix multiplication. Once the fast-computed $\hat{\ell}$ is obtained, the approximated success probability $\hat{\omega}$ is accordingly generated via Eq. (19).

3.2. Gaussian local projection

To ensure model accuracy, it is essential to establish competent projection bases. A satisfactory projection basis should embody the temperature distributions we want while having as less dimension as possible. In LPBF, the laser

Algorithm 2 The approximation of sampling probability.

```
Input: the basis function matrix $\Phi \in \mathbb{R}^{n_m n_e \times n_d}$, the Gaussian projection basis $\Psi \in \mathbb{R}^{n_d \times n_r}$, a parameter $\xi > 0$, a proportion $1/n_r \leqslant p < 1$, an embedding dimension $k$.
Output: an approximation of success probability $\hat{\omega}$.
$\tilde{n}_r = \lfloor p n_r \rfloor$.
Generate two random sub-Gaussian matrices $\Upsilon_1 \in \mathbb{R}^{n_r \times \tilde{n}_r}$ and $\Upsilon_2 \in \mathbb{R}^{k \times n_m n_e}$.
Compute $\tilde{\Phi} \in \mathbb{R}^{n_m n_e \times \tilde{n}_r}$ by $\tilde{\Phi} = \Phi(\Psi \Upsilon_1)$.
Compute the orthonormal matrix $\Lambda \in \mathbb{R}^{n_m n_e \times \tilde{n}_r}$ in the QR-decomposition of $\tilde{\Phi}$.
Compute the left singular vectors of $(\Upsilon_2 \Lambda)^{\dagger} \Upsilon_2 \Phi \Psi$ as $\bar{U} \in \mathbb{R}^{\tilde{n}_r \times \tilde{n}_r}$.
$\tilde{U} = \Lambda \bar{U}$.
for $i = 1$ to $n_m n_e$ do
    $\hat{\ell}_i = (\|\tilde{U}_{i*}\|_2)^2$.
    $\hat{\omega}_i = \min(1, \xi / n_r \hat{\ell}_i)$.
end for
```

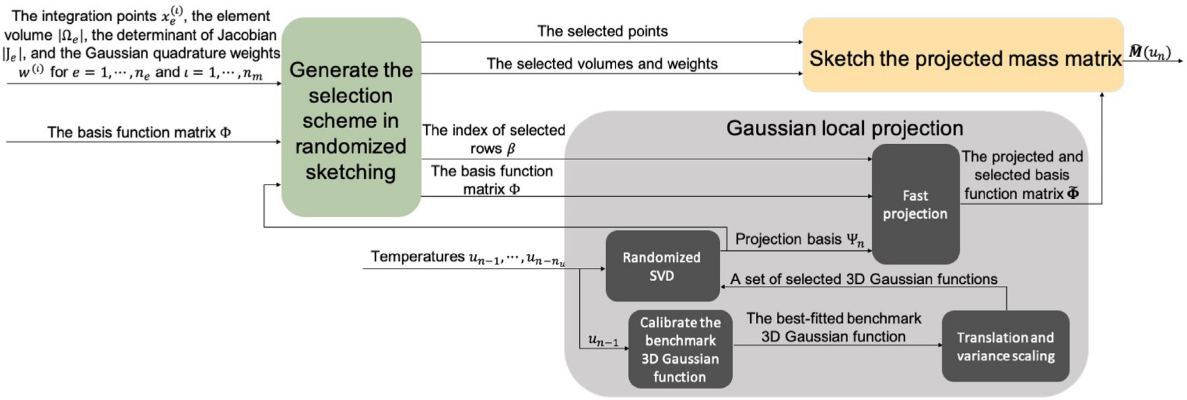

The integration points $x_e^{(\iota)}$, the element volume $|\Omega_e|$, the determinant of Jacobian $|J_e|$, and the Gaussian quadrature weights $w^{(\iota)}$ for $e = 1, \cdots, n_e$ and $\iota = 1, \cdots, n_m$
The basis function matrix Φ
Generate the selection scheme in randomized sketching
The selected points
The selected volumes and weights
Sketch the projected mass matrix
$\hat{M}(u_n)$
Gaussian local projection
The index of selected rows β
The basis function matrix Φ
Fast projection
The projected and selected basis function matrix $\tilde{\Phi}$
Projection basis Ψ_n
Temperatures $u_{n-1}, \cdots, u_{n-n_u}$
Randomized SVD
A set of selected 3D Gaussian functions
u_{n-1}
Calibrate the benchmark 3D Gaussian function
The best-fitted benchmark 3D Gaussian function
Translation and variance scaling

Fig. 5. The projection procedure to compute the projected and sketched mass matrix.

beam scans along the pre-determined trajectory yielding melt pools at different positions, the areas around which are also where the changes of temperature gradient concentrate. As a result, it makes sense to use local projection bases instead of a global basis since temperatures at different time coordinates are expected to be dissimilar. The local projection basis Ψ_n at the nth time step, though required to be updated as the printing process carries on, only needs to focus on the positions around the laser beam center $\mu(t_n)$ and thus requires fewer dimensions for an accurate delineation of $u(t_n)$. More specifically, the temperature $u(t)$ provoked by a moving Gaussian heat source as Eq. (4) is expected to have its peak at $\mu(t)$ and can be roughly outlined by a set of 3D Gaussian functions adjusted from a benchmark Gaussian function with operations including translation and variance scaling. The parameters of a benchmark Gaussian function, however, are calibrated by linear least square regression based on the latest temperature. The procedure of Gaussian local projection is exhibited as Fig. 5 in general, while the details of projection basis generation and projection are respectively explained in Sections 3.2.1 and 3.2.2.

3.2.1. *Projection basis generation*

Inspired by the Gaussian heat source, melt pool shapes, and anisotropic temperature distributions, we shall gather a set of 3D Gaussian functions to generally cover the temperature we want. The selected Gaussian functions are evaluated at all DoFs as $G \in \mathbb{R}^{n_d \times n_G}$ where n_G denotes the number of Gaussian functions. Considering the heat remained by the previous temperatures, the projection basis Ψ_n is then established as an orthonormal basis of the matrix $Q \in \mathbb{R}^{n_d \times (n_G + n_u + 1)}$ which is

$$Q = [G, u_{n-n_u}, \ldots, u_{n-1}, 1], \tag{22}$$

where the temperatures at the previous n_u time steps are included. The point, however, is how to properly select n_G Gaussian functions. We establish a Gaussian function with two parameters: the position of the mean and the variances. First, we can set n_μ mean positions located at the laser beam center at the nth time step and several positions ahead of the laser beam along the printing direction. In a straight line printing, the coordinates of the selected means are $\mu^n \in \mathbb{R}^{3\times n_\mu}$. $\mu^n_{*i} = \mu(t_n) + i[\Delta d, 0, 0]^T$ where $i = 0, 1, \ldots, n_\mu - 1$ in forward scanning and $i = 0, -1, \ldots, -(n_\mu - 1)$ in backward scanning. Then, we can find n_σ sets of standard deviations by properly scaling a set of benchmark values $\bar{\sigma} \in \mathbb{R}^3$ with an empirical vector $\eta \in \mathbb{R}^{n_\eta} > 0$. The benchmark standard deviations $\bar{\sigma}$ are obtained by roughly outlining the normalized latest temperature $u_{n-1}/\max(u_{n-1})$ as a 3D Gaussian function $f_{n-1}(x) = \exp(-\sum_{\tau=1}^{3}(x_\tau - \mu(t_{n-1})_\tau)^2/2\bar{\sigma}_\tau^2)$. The optimal $\bar{\sigma}$ satisfies

$$\operatorname*{argmin}_{\bar{\sigma}} \|u_{n-1}(\bar{X})/\max(u_{n-1}) - f_{n-1}(\bar{X})\|_2, \tag{23}$$

where $u_{n-1}(\bar{X}) \in \mathbb{R}^{\bar{n}_d}$ contains the DoF in $u_{n-1} \in \mathbb{R}^{n_d}$ that are greater than a temperature u_m. u_m is set empirically but is normally around the melting temperature of the material. The coordinates of these $\bar{n}_d$ nodes are specified as the rows in the matrix $\bar{X} \in \mathbb{R}^{\bar{n}_d\times 3}$. With linear least square regression, $\hat{\bar{\sigma}} \in \mathbb{R}^3$ is approximated by

$$1/\hat{\bar{\sigma}}^2 = (B(\bar{X})^T B(\bar{X}))^{-1} B(\bar{X})^T \ln(u_{n-1}(\bar{X})/\max(u_{n-1})), \tag{24}$$

where $B(\bar{X}) \in \mathbb{R}^{\bar{n}_d\times 3}$ is $B(\bar{X}) = \frac{1}{2}[(\bar{X}_{*1} - \mu(t_{n-1})_1\mathbf{1})^2, (\bar{X}_{*2} - \mu(t_{n-1})_2\mathbf{1})^2, (\bar{X}_{*3} - \mu(t_{n-1})_3\mathbf{1})^2]$. $\hat{\bar{\sigma}}$ is then properly scaled by η yielding n_η^3 different sets of standard deviations as

$$\sigma(\eta)_{*i} = [\exp(\eta_j \ln \bar{\sigma}_1), \exp(\eta_k \ln \bar{\sigma}_2), \exp(\eta_q \ln \bar{\sigma}_3)]^T, \quad \text{for} \quad i = 1, \ldots, n_\sigma, \tag{25}$$

where $n_\sigma = n_\eta^3$. The jth, kth, and qth entry of η constitute one possible arrangement in the permutations of η with repetition. Considering the designs above, we can collect a total number of $n_G = n_\mu n_\sigma$ Gaussian functions and gather the matrix G in Eq. (22) as

$$G_{di} = \exp(-\sum_{\tau=1}^{3} \frac{(X_{\tau d} - \mu^n_{\tau j})^2}{2\sigma(\eta)_{\tau k}{}^2}), \quad \text{for} \quad d = 1, \ldots, n_d, \quad \text{and} \quad i = 1, \ldots, n_G, \tag{26}$$

where $X \in \mathbb{R}^{3\times n_d}$ contains the spatial coordinate of the n_d DoF. The ith column of G arranges the jth of n_μ means and the kth of n_σ standard variances as one possible permutation. The projection basis $\Psi_n \in \mathbb{R}^{n_d\times n_r}$ is then formed as the left singular vectors of the n_r-rank approximation of Q. As the number of DoF n_d is large and the number of selected Gaussian functions n_G is typically a few hundred, the SVD of Q can be computationally expensive. As a result, we use randomized SVD as algorithm B.1 in Appendix B to reduce the generation time of the orthonormal basis Ψ_n [31].

3.2.2. *Fast projection*

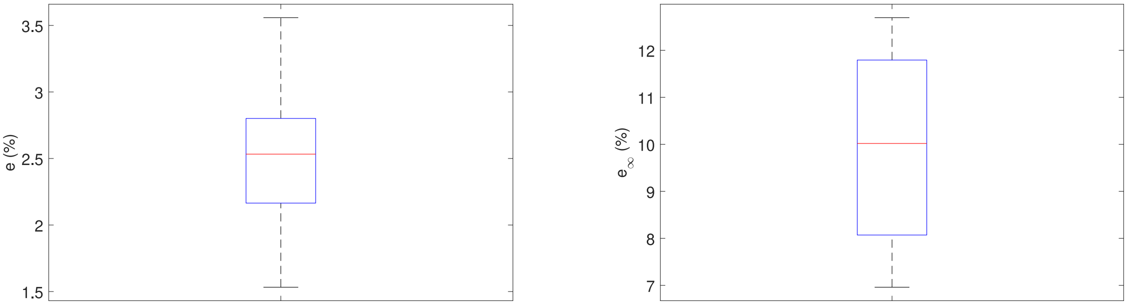
According to the design above, the projection basis Ψ_n is generated in the process of simulation and will not be established until u_{n-1} becomes available. Hence, the action of projection should also be implemented in the process of simulation. The complete projection should be finished by

$$\mathbf{\Phi} = \Phi \Psi_n, \tag{27}$$

which is the expensive large matrix multiplication that we want to avoid all along. We have avoided computing it explicitly in randomized sketching as stated in Section 3.1.1 and algorithm 2. Consequently, we can obtain the row indexes of n_g successful selections as β before fulfilling the projection. We can thereby skip $\mathbf{\Phi}$ and directly get $\tilde{\mathbf{\Phi}}$ as

$$\tilde{\mathbf{\Phi}} = \Phi_{\beta*} \Psi_n. \tag{28}$$

Since the majority of rows are bypassed via randomized sketching, we have $n_g \ll n_m n_e$ and thus the computational complexity is significantly reduced from $O(n_m n_e n_d n_r)$ for Eq. (27) to $O(n_g n_d n_r)$ for Eq. (28).



(a) 2-norm relative error

(b) infinity-norm of the component-wise relative error

Fig. 6. Relative errors between FEM and SM.

4. Results

The performance of the designed surrogate model (SM) is evaluated by comparing its model accuracy and time cost with the full-order model using FEM. In Section 4.1, a simplified example is evaluated, the codes of which are provided in the Github repository [32]. In Section 4.2, numerical experiments are carried out with the considerations under the context of LPBF such as intricate nonlinearity, phase changes, and anisotropy. All tests are implemented in Matlab R2020b on a computer with 2.6 GHz 6-Core Intel Core i7 processor and 16 GB RAM.

4.1. A simplified example

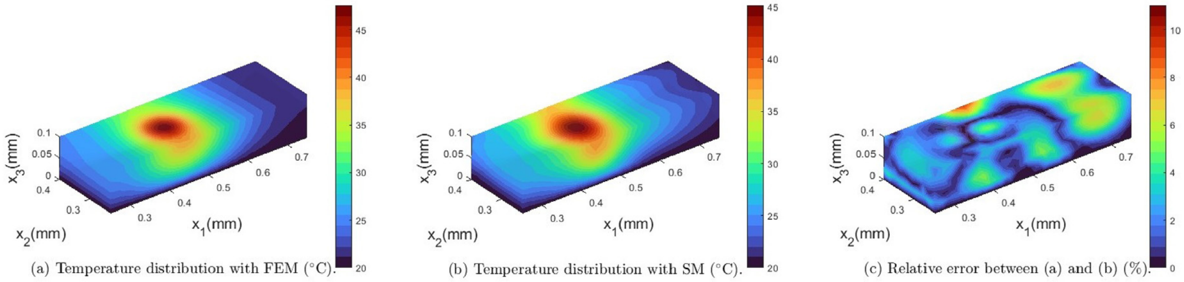
An example is provided to simulate the heat transfer with a Gaussian heat source moving forward with a speed v. The Gaussian function follows Eq. (4) with a fixed amplitude denoted as f_s in this section. The three-dimensional domain is discretized by mesh with less refinement compared with Section 4.2, and the thermal properties including thermal conductivity, density, and specific heat capacity are temperature-dependent but are only set as simple polynomials. As a result, this simplified example is comparably not high-dimensional and only has mild nonlinearity. The parameter details are listed in Appendix C, in which the parameters of the thermal model and the surrogate are respectively in Tables C.1 and C.2.

4.1.1. Model accuracy

The model accuracy is evaluated by relative errors which are computed in 2-norm and infinity-norm. Both are defined on the entire n_d-dimensional temperatures. Namely, when $u \in \mathbb{R}^{n_d}$ and $\hat{u} \in \mathbb{R}^{n_d}$ respectively denote the simulated temperature with FEM and the corresponding estimated temperature with the SM, we have the 2-norm relative error as $e = \|u - \hat{u}\|_2 / \|u\|_2$ and the infinity-norm of the component-wise relative error as $e_\infty = \|e^c\|_\infty$ where $e_i^c = |u_i - \hat{u}_i| / u_i$ for $i = 1, \ldots, n_d$. The boxplots as Fig. 6 show the two types of relative errors for the simplified example in this section. In each box, the red line represents the median while the bottom and top line are respectively the 25th percentile (Q_1) and 75th percentile (Q_3). The upper and lower whiskers of each box additionally extend with a distance of $1.5 \times (Q_3 - Q_1)$. We can tell that e range from 1.53% to 3.56% while e_∞ range from 6.96% to 12.69%. More specifically, we can compare the temperature distributions between FEM and SM. Take one time step ($t = 0.4$ ms) as an example, the temperature profiles and the relative errors are shown in Fig. 7.

4.1.2. Time cost reduction

The average time cost of FEM and SM are summarized in Table 1 where the time cost of SM consists of four parts: basis generation, sketching, projection, and simulation. It indicates that 33.64% of time cost is saved on average. The total time cost is reduced from 1.3414s to 0.8901s, which is not a significant reduction. However, a more remarkable improvement in time efficiency will be shown when the thermal model has a higher dimension and nonlinearity.



(a) Temperature distribution with FEM (°C).

(b) Temperature distribution with SM (°C).

(c) Relative error between (a) and (b) (%).

Fig. 7. The comparison of temperature distribution between FEM and SM.

Table 1
The comparison of average time cost.

FEM (s)	SM (s)					Reduction (%)
	Basis generation	Sketching	Projection	Simulation	Total	
1.3414	0.0066	0.2754	0.0077	0.6004	0.8901	33.64

4.2. Numerical experiments of LPBF

A series of numerical experiments are carried out to simulate two straight lines scanned back and forth in a two-layer AlSi10Mg powder bed surrounded by argon atmosphere where the laser beam starts from $x_s^{(i)}$ and ends at $x_e^{(i)}$ with a hatch distance Δh on the top of the ith layer. The powder bed domain numerically expressed as a cuboid is discretized to tetrahedron meshes with a conforming refinement within the printing area, in which fine meshes distribute around the scanning lines on the top layer (tetrahedrons with a side length of 0.005 mm) and the rest area has coarser mesh (tetrahedrons with a side length of 0.025 mm). Under this mesh scheme, the domain Ω evolving from the first to the second layer is spatially discretized into n_e tetrahedron elements, n_d DoFs, and $\bar{n}_d$ constant temperature nodes on the bottom surface. Specifically, we have $n_d = 86333$, $\bar{n}_d = 11011$, and $n_e = 510797$ for the first layer, and for the second layer $n_d = 96509$, $\bar{n}_d = 2134$, and $n_e = 568908$. It is worth noting that the printing area in a real LPBF process is normally larger and the mesh refinement used here may not be as efficient. For more realistic complexity or elaborate printing trajectories, more sophisticated mesh generation algorithms will be advantageous such as the area of fine mesh is set to be around and moving with the melt pool. The full-order model with FEM is used as a reference model, and we validate it with the experiment results in [16] which includes a selective laser melting process using the apparatus with a YLR-500-SM ytterbium fiber laser and a numerical simulation using the ANSYS multiphysics finite element package. The reference experiment was conducted with laser power $P = 250$ W and scan speed $v = 200$ mm/s using the same material (AlSi10Mg powders with 99.7% purity and an average size around 30μm) in the same atmosphere (argon). The scanning pattern of a laser beam is visualized as Fig. 8 where the positions at the center of each layer are used to validate our thermal simulation. We further detail the validation in Appendix D. There are a total amount of 18 tests including different laser power (P=200, 250, 300 W), scan speed (v=200, 600, 1000 mm/s), as well as a static or random anisotropy scale factor λ. While the static scale factor $\lambda = [2.5, 1, 3.2]^T$ is found by the trial-and-error calibration to fit the experiment results in [16], the three values in the random scale factor λ are respectively generated from uniform distributions $\lambda_1 \sim U(1, 4)$, $\lambda_2 \sim U(0.5, 1.5)$, and $\lambda_3 \sim U(1.5, 4.5)$ to simulate the randomness of anisotropy in reality. Therefore, with the 18 tests denoted as $\mathcal{S}_i$ for $i = 1, \ldots, 18$ we validate the feasibility of our surrogate to cope with different printing parameters and a certain level of anisotropy randomness. The parameter settings of each test are specified in Table E.1 in Appendix E. In Appendix C, the model parameters used in both FEM and the surrogate are specified in Table C.3, and the parameters specific to the surrogate are specified in Table C.4.

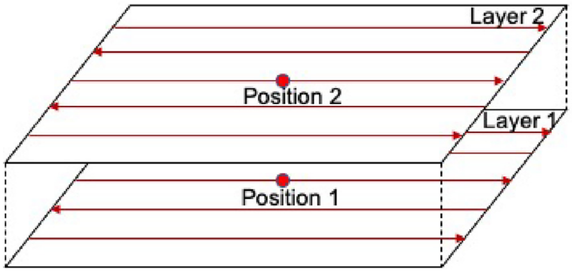
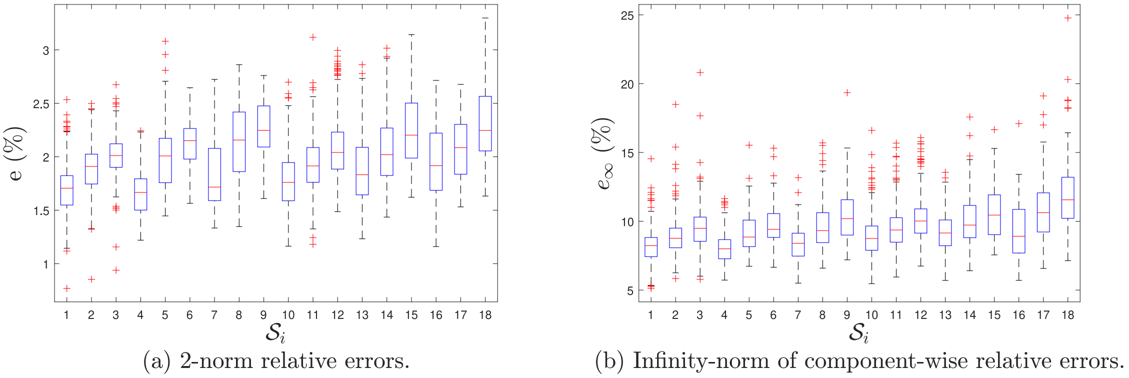


Fig. 8. The laser scanning pattern in [16] where position 1 and 2 used in validation are respectively the center of the first and second layer.



(a) 2-norm relative errors.

(b) Infinity-norm of component-wise relative errors.

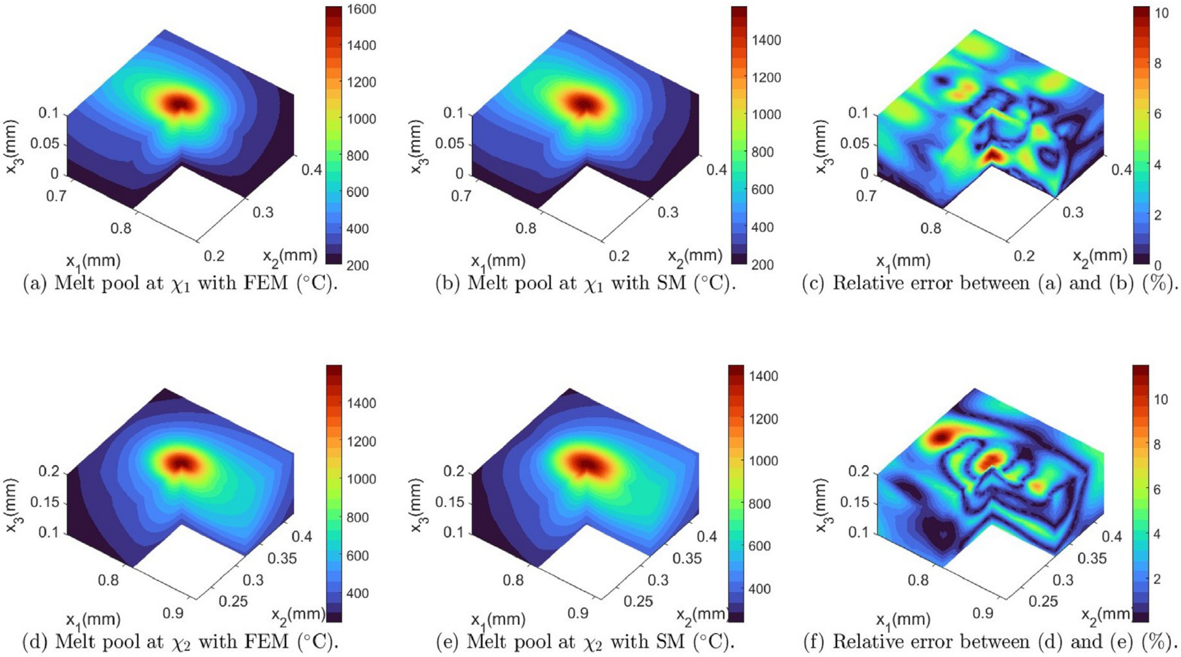
Fig. 9. Relative errors between FEM and SM.

Table 2
The melt pool size comparison between FEM and SM.

Test	Layer	FEM (μm)			SM (μm)			Relative error (%)		
		Length	Width	Depth	Length	Width	Depth	Length	Width	Depth
$\mathcal{S}_9$	1	162.50	98.77	62.50	167.50	103.34	60.76	3.08	4.63	2.78
	2	185.00	97.98	68.36	191.31	100.23	63.74	3.41	2.30	6.76
$\mathcal{S}_{10}$	1	115.00	93.90	58.39	112.71	93.90	54.37	1.99	< 0.01	6.88
	2	112.50	97.57	62.69	115.30	101.21	60.02	2.49	3.73	4.26

4.2.1. Model accuracy

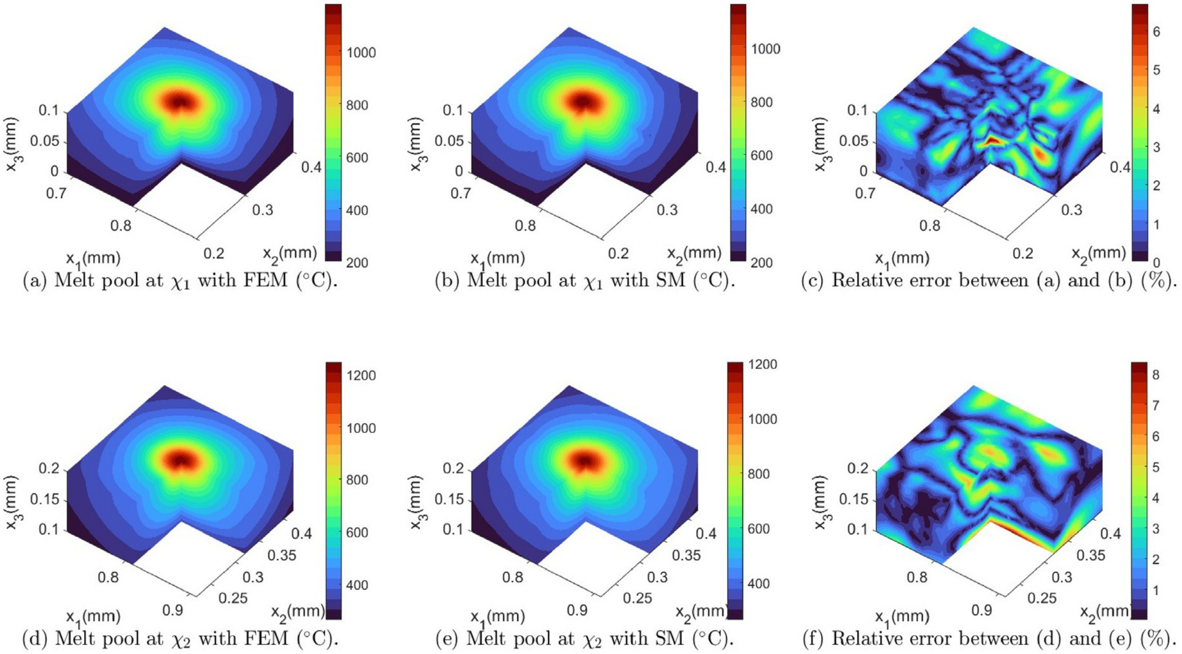
The model accuracy is compared in three aspects: relative errors, melt pools, and thermal histories. The spread of relative errors is shown in the boxplot as Fig. 9 where the ith box expresses the relative errors of the test $\mathcal{S}_i$ for $i = 1, \ldots, 18$. We can tell that most of the 2-norm relative errors are smaller than 3%. More specifically, there are respectively 99.62% and 60.64% of 2-norm relative errors below 3% and 2%, while the maximum 2-norm relative error among all tests is 3.30%. The infinity norm of component-wise relative errors, however, range from 5.12% to 24.77% where 98.79% and 70.42% of tests are respectively below 15% and 10%. In addition to the comparison between high-dimensional temperatures, we further focus on the area around melt pools where most temperature gradients concentrate. Take $\mathcal{S}_9$ and $\mathcal{S}_{10}$ as examples, χ_i denotes the middle position of the ith scanning line on the ith layer. The comparison of the melt pool size is listed in Table 2 where the maximum relative error of the four melt pool sizes are respectively 4.63%, 6.76%, 6.88%, and 4.26%. In Figs. 10 and 11, we show the contours of temperature distributions together with the relative errors of temperatures at individual nodes. The errors inhomogeneously distribute over the domain, and the maximum of the plotted errors which is also the infinity-norm of the component-wise relative errors are respectively 10.21%, 11.46%, 6.68%, and 8.36%. At the



(a) Melt pool at χ_1 with FEM (°C). (b) Melt pool at χ_1 with SM (°C). (c) Relative error between (a) and (b) (%).

(d) Melt pool at χ_2 with FEM (°C). (e) Melt pool at χ_2 with SM (°C). (f) Relative error between (d) and (e) (%).

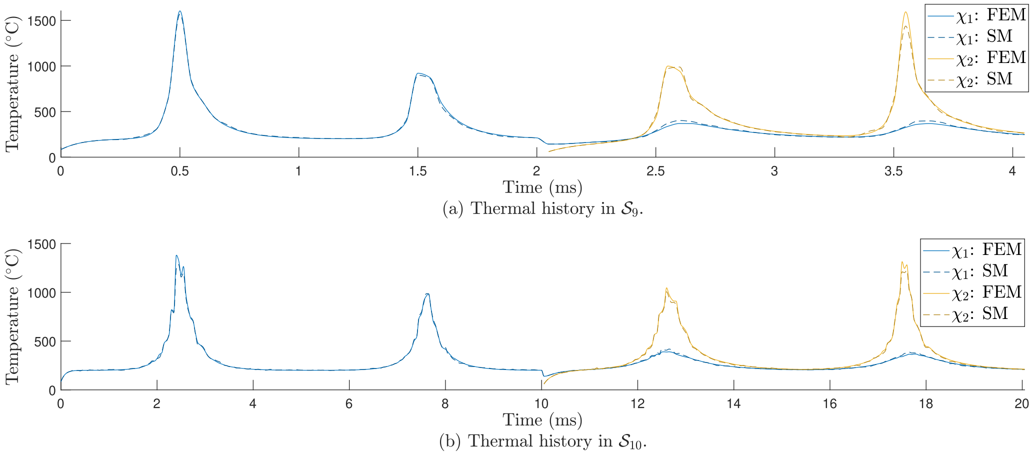
Fig. 10. The melt pool comparison of $\mathcal{S}_9$ between FEM and SM.



(a) Melt pool at χ_1 with FEM (°C). (b) Melt pool at χ_1 with SM (°C). (c) Relative error between (a) and (b) (%).

(d) Melt pool at χ_2 with FEM (°C). (e) Melt pool at χ_2 with SM (°C). (f) Relative error between (d) and (e) (%).

Fig. 11. The melt pool comparison of $\mathcal{S}_{10}$ between FEM and SM.



(a) Thermal history in $\mathcal{S}_9$.

(b) Thermal history in $\mathcal{S}_{10}$.

Fig. 12. The thermal history comparison between FEM and SM.

Table 3
The comparison of average time cost.

Layer	FEM (s)	SM (s)					Reduction (%)
		Basis generation	Sketching	Projection	Simulation	Total	
1	399.37	1.51	22.55	1.85	25.73	51.64	87.07
2	519.93	1.82	23.71	2.13	24.82	52.48	89.91

same four positions, the thermal histories are compared as Fig. 12. In $\mathcal{S}_9$, the relative errors of thermal histories at χ_1 and χ_2 are respectively 3.12% and 5.94%. For the thermal histories at χ_1 and χ_2 in $\mathcal{S}_{10}$, we can see the effect of anisotropy randomness while their relative errors are respectively 3.31% and 3.10%.

4.2.2. Time cost reduction

While the accuracy of SM is validated in Section 4.2.1, another important performance is the reduction of execution time. At each time step of the high-fidelity thermal model with FEM, the execution time consists of Picard iterations. Therefore, for one layer of printing in one test, we record and compute the average time cost of implementing the simulation where the nonlinear thermal properties and boundary conditions are repeatedly evaluated at each integration point in each Picard iteration. While the basis generation, sketching, and projection of SM respectively corresponding to Sections 3.2.1, 3.1, and 3.2.2 are successively executed in the process of simulation, the simulation part is also the Picard iterations but is implemented with the projected and sketched model where only a small amount of nonlinear computations are required. On average, as shown in Table 3, the SM manages to respectively reduce 87.07% and 89.91% of time cost for the one- and two-layer printing. Specifically, while the average execution time of FEM is respectively 399.37s for layer 1 and 519.93s for layer 2, the average execution time of SM is correspondingly 51.64s and 52.48s. Most of the execution time in SM is taken by sketching and simulation parts, though both of which are necessary they can be further reduced if more sacrifice in accuracy is acceptable. Namely, a smaller proportion p in algorithm 2 will reduce the sketching time required, and a smaller multiplier ξ will cut down the simulation time required.

5. Conclusions

A fast surrogate model is proposed as a swift alternative to the high-fidelity thermal model of LPBF. The thermal model is governed by a nonlinear and anisotropic heat equation considering temperature-dependent thermal properties and phase changes. It is numerically solved by FEM with Gaussian quadrature and Picard iteration. To build the surrogate, we first run the full-order model with FEM for the first several time steps until the melt pool becomes stable. Then, the full-order model is projected on a local projection basis generated in the process of simulation. The local projection basis is formed by gathering several previous temperatures and a set of Gaussian functions calibrated by the temperature at the last time step. The projected model is then randomly sketched with the approximate sampling probability which requires less sketching time and avoids explicitly implementing full projection. After generating the row selection scheme by randomized sketching, we implement the projection with only the selected rows. To validate the performance of the surrogate, we make a simplified example and several numerical experiments of LPBF. The simplified example has relatively low dimensionality and uncomplicated nonlinearities. The codes of its implementation are available in the Github repository [32]. While the 2-norm relative errors of the simplified example are below 3.56%, its infinity-norm relative errors are less than 12.69%. The average running time is reduced from 1.34s to 0.89s saving 33.64% of time cost. The performance of time cost reduction becomes more remarkable in the numerical experiments of LPBF where we simulate the printing process of two straight lines scanned back and forth in a two-layer AlSi10Mg powder bed surrounded by an argon atmosphere. 18 numerical experiments are carried out with different laser power, scan speed, and static/random scale factor of anisotropy. On average, 88.67% of execution time is saved while respectively suppressing the 2-norm relative errors for 60.64% and 99.62% of tests below 2% and 3%. Therefore, the designed surrogate can be used as a fast-solving alternative that is competent enough to deal with different printing parameters and some anisotropy randomness. Since it swiftly and accurately emulates temperature profiles, melt pool sizes, and thermal histories, it is essential in facilitating the prediction and analysis of the quality of the printed part such as its deformation and mechanical properties. Compared with the two surrogates proposed in [8], its advantages are reflected in two aspects. First, its basis generation, projection, and randomized sketching do not rely on a data-driven model trained and prepared before simulations start. Therefore, its performance does not rely on the representativeness of training data and it saves the preparation effort of data generation and training. Second, since the projection bases of the surrogate in this paper are generated from hundreds of calibrated and adjusted Gaussian functions, they are feasible to cover a larger range of temperature distributions which makes the surrogate become more robust in some anisotropy randomness. Both advantages make the surrogate more practical and applicable in a real LPBF process. In principle, the surrogate is also valid for other types of thermal-driven AM if we properly adjust the designs like the evolution of the domain and the model of the heat source. Future work can focus on the application of the surrogate in solving problems requiring large and/or fast thermal simulations such as possible process optimizations and controls.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

XL would like to acknowledge the financial support for the PhD studies provided by the Principal's Career Development and Edinburgh Global Research Scholarship from the School of Engineering at the University of Edinburgh, UK. NP is grateful to EPSRC, United Kingdom for their support via the grant EP/V028618/1 (RAPID:Real-time process modelling and Diagnostics). For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising from this submission.

Appendix A. Finite element method

To solve the nonlinear thermal model of LPBF defined as Eqs. (1)–(7), we develop a high-fidelity numerical solver with FEM. The temporal domain is discretized as $t_n = n\varDelta t$ for $n = 0, 1, \ldots, N$, while the spatial domain is discretized into n_s surface triangles, n_e tetrahedron elements, and $\tilde{n}_d = n_d + \bar{n}_d$ nodes including n_d DoF and $\bar{n}_d$ nodes on the bottom surface $\varGamma_b$. The temperatures are approximated by linear combinations of basis functions ϕ_i for $i = 1, \ldots, \tilde{n}_d$. In other words, the temperature at the nth time step is approximated as $\sum_{i=1}^{\tilde{n}_d} \tilde{u}_{n_i}\phi_i$ where $\tilde{u}_n \in \mathbb{R}^{\tilde{n}_d}$ contains the temperature $u_n \in \mathbb{R}^{n_d}$ at n_d DoF and the temperatures $\bar{u} = u_b \times \mathbf{1} \in \mathbb{R}^{\bar{n}_d}$ imposed as a

Table A.1
The details of FEM.

Symbol	Details $(j = 1, \dots, n_d,\ \bar{j} = 1, \dots, \bar{n}_d,$ $e = 1, \dots, n_e,\ i = 1, \dots, n_s)$	Symbol	Details $(j = 1, \dots, n_d,\ \bar{j} = 1, \dots, \bar{n}_d,$ $e = 1, \dots, n_e,\ i = 1, \dots, n_s)$
$K(u_n)$	$\Phi_k^T W_k(u_n) \Phi_k$	$D^n(X^{(\iota)})$	$D^n_{ee} = \lvert J_e\rvert \, \lvert\Omega_e\rvert \, w_\iota \rho(u_n(x_e^{(\iota)}))c(u_n(x_e^{(\iota)}))$
$M(u_n)$	$\Phi^T W(u_n) \Phi$	$D_r^n(X^{(\iota)})$	$D^n_{r_{ii}} = \lvert J_i\rvert \, \lvert\Gamma_i\rvert \, \sigma_s \varepsilon w_\iota u(x_i^{(\iota)})^3$
$Q_r(u_n)$	$\Phi_r^T W_r(u_n) \Phi_r$	$D_c(X^{(\iota)})$	$D_{c_{ii}} = \lvert J_i\rvert \, \lvert\Gamma_i\rvert \, w_\iota h$
Q_c	$\Phi_c^T W_c \Phi_c$	d_s	$d_{s_i} = \lvert J_i\rvert \, \lvert\Gamma_i\rvert \, w(hu_a + \sigma_s \varepsilon u_a^4)$
s	$\Phi_s^T d_s$	Φ_s	$\Phi_{s_{*j}} = \phi_j(X)$
$l(t_n)$	$\Phi_l^T d_l(t_n)$	Φ_l	$\Phi_{l_{*j}} = [\phi_j(X^{(1)}), \dots, \phi_j(X^{(n_l)})]^T$
$\bar{K}(u_n)$	$\Phi_k^T W_k(u_n) \bar{\Phi}_k$	$\bar{\Phi}_r$	$\bar{\Phi}_{r_{*\bar{j}}} = [\phi_{\bar{j}}(X^{(1)}), \dots, \phi_{\bar{j}}(X^{(n_r)})]^T$
$\bar{Q}_r(u_n)$	$\Phi_r^T W_r(u_n) \bar{\Phi}_r$	$\bar{\Phi}_c$	$\bar{\Phi}_{c_{*\bar{j}}} = [\phi_{\bar{j}}(X^{(1)}), \dots, \phi_{\bar{j}}(X^{(n_c)})]^T$
$\bar{Q}_c$	$\Phi_c^T W_c \bar{\Phi}_c$	Φ	$\Phi_{*j} = [\phi_j(X^{(1)}), \dots, \phi_j(X^{(n_m)})]^T$
$d_l(t_n)$	$\left[\mathrm{d}_l^n(X^{(1)}), \dots, \mathrm{d}_l^n(X^{(n_l)})\right]^T$	Φ_r	$\Phi_{r_{*j}} = [\phi_j(X^{(1)}), \dots, \phi_j(X^{(n_r)})]^T$
$\mathrm{d}_l^n(X^{(\iota)})$	$\mathrm{d}^n_{l_{1i}} = \lvert J_i\rvert \, \lvert\Gamma_i\rvert \, w_\iota q(x_i^{(\iota)}, t_n)$	Φ_c	$\Phi_{c_{*j}} = [\phi_j(X^{(1)}), \dots, \phi_j(X^{(n_c)})]^T$
$W(u_n)$	$\begin{bmatrix} D^n(X^{(1)}) & & \\ & \ddots & \\ & & D^n(X^{(n_m)}) \end{bmatrix}$	$W_r(u_n)$	$\begin{bmatrix} D_r^n(X^{(1)}) & & \\ & \ddots & \\ & & D_r^n(X^{(n_r)}) \end{bmatrix}$
$W_c(u_n)$	$\begin{bmatrix} D_c(X^{(1)}) & & \\ & \ddots & \\ & & D_c(X^{(n_c)}) \end{bmatrix}$	$W_k(u_n)$	$\begin{bmatrix} W_{k_{3e-2,3e-2}} \\ W_{k_{3e-1,3e-1}} \\ W_{k_{3e,3e}} \end{bmatrix} = \lvert J_e\rvert \, \lvert\Omega_e\rvert \sum_{\iota=1}^{n_k} w_\iota \kappa(u_n(x_e^{(\iota)}))\lambda$
Φ_k	$\begin{bmatrix} \Phi_{k_{3e-2,j}} \\ \Phi_{k_{3e-1,j}} \\ \Phi_{k_{3e,j}} \end{bmatrix} = \nabla \phi_j$	$\bar{\Phi}_k$	$\begin{bmatrix} \bar{\Phi}_{k_{3e-2,\bar{j}}} \\ \bar{\Phi}_{k_{3e-1,\bar{j}}} \\ \bar{\Phi}_{k_{3e,\bar{j}}} \end{bmatrix} = \nabla \phi_{\bar{j}}$

constant temperature u_b according to the Dirichlet condition as Eq. (6). The matrix $A(u_n) \in \mathbb{R}^{n_d \times n_d}$ and the vector $b(u_{n-1}, u_n) \in \mathbb{R}^{n_d}$ in Eq. (8) are composed by

$$A(u_n) = K(u_n) + \frac{1}{\Delta t} M(u_n) + Q_r(u_n) + Q_c, \tag{A.1}$$

$$b(u_{n-1}, u_n) = \frac{1}{\Delta t} M(u_n) u_{n-1} + l(t_n) + s - (\bar{K}(u_n) + \bar{Q}_r(u_n) + \bar{Q}_c)\bar{u}. \tag{A.2}$$

As stated in Eq. (A.1), the matrices on the left hand side of Eq. (8) are defined on DoF including the matrix of convection heat loss $Q_c \in \mathbb{R}^{n_d \times n_d}$ that are not temperature-dependent and three temperature-dependent matrices: the stiffness matrix $K \in \mathbb{R}^{n_d \times n_d}$, the mass matrix $M \in \mathbb{R}^{n_d \times n_d}$, and the matrix of radiation heat loss $Q_r \in \mathbb{R}^{n_d \times n_d}$. On the right hand side of Eq. (8), there are the terms $\frac{1}{\Delta t} M(u_n) u_{n-1} + l(t_n) + s$ defined on n_d DoF and the terms $\bar{K}(u_n) + \bar{Q}_r(u_n) + \bar{Q}_c$ being the coefficients of $\bar{n}_d$ nodes on Γ_b. We subtract the terms relating to the Dirichlet condition $(\bar{K}(u_n) + \bar{Q}_r(u_n) + \bar{Q}_c)\bar{u}$ to balance the left hand side that only contains the n_d DoF. The matrices and vectors in Eq. (A.1) and (A.2) are integrals approximated by Gaussian quadrature rules. There are respectively $n_k n_e$, $n_m n_e$, $n_r n_s$, $n_c n_s$, n_s, and $n_l n_s$ integration points and weights are appropriately chosen for $K(u_n)$, $M(u_n)$, $Q_r(u_n)$, Q_c, s, and $l(t_n)$ according to [33,34]. Further details are specified by Table A.1. The nonlinear heat equation is then solved via Picard iterations where the final temperature solution is obtained after several iterations solving $A^{-1}b$. As state in algorithm B.1 in [8], the iteration stops when the error $\|A(u_n^{(i+1)})u_n^{(i+1)} - b(u_{n-1}, u_n^{(i+1)})\|_2$ or $\|u_n^{(i+1)} - u_n^{(i)}\|_2 / \|u_n^{(i+1)}\|_2$ below a small value like 10^{-5} or the maximum number of iteration reaches [27].

Appendix B. Randomized SVD with sub-Gaussian random matrices

We deploy the randomized SVD as algorithm B.1 in [31] where the embedding parameter k in our case can be set as 1 while maintaining the accuracy of projection bases.

Table C.1
Thermal model parameters in the simplified example.

Symbol	Definition	Value	Unit
n_d	the number of degrees of freedom	2986	–
$\bar{n}_d$	the number of nodes on Γ_b	561	–
n_e	the number of elements	17453	–
–	the size of domain	0.50 × 0.17 × 0.10	mm
–	the side length of refined tetrahedron mesh	0.01	mm
–	the side length of coarse tetrahedron mesh	0.025	mm
Δt	A time step in temporal discretization	0.05	ms
N	The total amount of time steps	15	–
v	The moving speed of Gaussian heat source	400	mm/s
f_s	The amplitude of Gaussian heat source	45000	–
ε	Emissivity	0.04	–
h	Heat convection coefficient	10	W/(m^2 K)
ς	The effective radius of Gaussian heat source	50	µm
u_a	Ambient temperature	20	°C
u_b	The constant temperature on Γ_b	20	°C
κ	Thermal conductivity	$u^3 + u^2 + u + 1$	W/m K
ρ	Density	$u + 1$	kg/m^3
c	Specific heat capacity	$u + 1$	J/kg K

Table C.2
The surrogate model parameters in the simplified example.

Symbol	Definition	Value	Unit
η	the scale factor of Gaussian function variance	$[0.5, 1, 1.5]^T$	–
u_m	the temperature threshold of node selection	30	°C
r	the dimension of projection	40	–
ξ	the multiplier of leverage score	4000	–
n_t	the number of FOM time steps	5	–
n_u	the number of previous temperatures	5	–
n_μ	the number of mean positions	2	–
p	the selection proportion in randomized sketching	60%	–
Δd	the distance interval of translation	50	µm

Algorithm B.1 Randomized SVD with sub-Gaussian random matrices

```
Input: a matrix $Q \in \mathbb{R}^{n_d \times n_Q}$, an approximation rank $n_r$, an embedding dimension $k$ .
Output: the orthonormal basis $\Psi \in \mathbb{R}^{n_d \times n_r}$.
Generate two random sub-Gaussian matrices $\Upsilon_1 \in \mathbb{R}^{n_Q \times n_r}$ and $\Upsilon_2 \in \mathbb{R}^{k \times n_d}$.
$\Lambda_1 = Q \times \Upsilon_1$.
Compute the orthogonal matrix $\Lambda_2$ in the QR-decomposition of $\Lambda_1$.
Compute the left singular vectors of $(\Upsilon_2 \Lambda_2)^\dagger \Upsilon_2 Q$ as $\bar{\Psi}$.
$\Psi = \Lambda_2 \bar{\Psi}$.
```

Appendix C. The model parameters

C.1. Model parameters of the simplified example in Section 4.1

For the simplified model in Section 4.1, the parameters of the thermal model and the surrogate model are respectively listed as Tables C.1 and C.2.

Table C.3
Model parameters in the thermal model of LPBF [16,26,35,36].

Symbol	Definition	Value	Unit
$x_s^{(i)}$	The start position of the i-layer simulations	$[0.29, 0.30, 0.10i]^T$	mm
$x_e^{(i)}$	The end position of the i-layer simulations	$[0.29, 0.32, 0.10i]^T$	mm
Δh	Hatch distance	0.02	mm
–	The size of the i-layer domain	$1.54 \times 0.70 \times 0.10i$	mm
a	Absorptivity	0.09	–
ς	The effective laser beam radius	35	µm
u_b	The constant temperature on Γ_b	200	°C
u_a	Ambient temperature	20	°C
h	Heat convection coefficient	10	W/(m^2 K)
ε	Emissivity	0.04	–

Table C.4
Parameters of the surrogate.

Symbol	Definition	Value	Unit
η	the scale factor of Gaussian function variance	$[0.5, 0.8, 1, 1.2, 1.5]^T$	–
u_m	the temperature threshold of node selection	500	°C
r	the dimension of projection	200	–
ξ	the multiplier of leverage score	80000	–
n_t	the number of FOM time steps	5	–
n_u	the number of previous temperatures	5	–
n_μ	the number of mean positions	3	–
p	the selection proportion in randomized sketching	40%	–
Δd	the distance interval of translation	35	µm

Table D.1
The validation of the printing process with laser power 250 W and scan speed 200 mm/s.

Layer	Type of results (Unit)	Average simulation results	Results in [16]
1	Highest Temperature (°C)	1480	1482
	Melt Pool Size (µm)	126.4 × 100.3 × 63.3	129.1 × 94.2 × 61.7
2	Highest Temperature (°C)	1535	1548
	Melt Pool Size (µm)	152.8 × 115.3 × 80.2	148.3 × 111.4 × 67.5

C.2. Model parameters of the numerical experiments in Section 4.2

For the numerical experiments in Section 4.2, the parameters of the thermal model and the surrogate model are respectively listed as Tables C.3 and C.4.

Appendix D. The validation of the thermal model with FEM using published experiments

See Table D.1.

Appendix E. The parameter setting of tests

See Table E.1.

Table E.1
Parameters of the 18 tests.

Test	Laser power (W)	Scan speed (mm/s)	λ
$\mathcal{S}_1$	200	200	$[2.5, 1, 3.2]^T$
$\mathcal{S}_2$	200	600	$[2.5, 1, 3.2]^T$
$\mathcal{S}_3$	200	1000	$[2.5, 1, 3.2]^T$
$\mathcal{S}_4$	250	200	$[2.5, 1, 3.2]^T$
$\mathcal{S}_5$	250	600	$[2.5, 1, 3.2]^T$
$\mathcal{S}_6$	250	1000	$[2.5, 1, 3.2]^T$
$\mathcal{S}_7$	300	200	$[2.5, 1, 3.2]^T$
$\mathcal{S}_8$	300	600	$[2.5, 1, 3.2]^T$
$\mathcal{S}_9$	300	1000	$[2.5, 1, 3.2]^T$
$\mathcal{S}_{10}$	200	200	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{11}$	200	600	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{12}$	200	1000	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{13}$	250	200	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{14}$	250	600	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{15}$	250	1000	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{16}$	300	200	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{17}$	300	600	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{18}$	300	1000	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$

References

[1] M. Russell, A. Souto-Iglesias, T. Zohdi, Numerical simulation of laser fusion additive manufacturing processes using the SPH method, Comput. Methods Appl. Mech. Engrg. 341 (2018) 163–187.

[2] D. Gunasegaram, A. Murphy, A. Barnard, T. DebRoy, M. Matthews, L. Ladani, D. Gu, Towards developing multiscale-multiphysics models and their surrogates for digital twins of metal additive manufacturing, Addit. Manuf. 46 (2021) 102089.

[3] T. Zohdi, A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments, Comput. Methods Appl. Mech. Engrg. 363 (2020) 112907.

[4] Y. Zhang, G. Guillemot, M. Bernacki, M. Bellet, Macroscopic thermal finite element modeling of additive metal manufacturing by selective laser melting process, Comput. Methods Appl. Mech. Engrg. 331 (2018) 514–535.

[5] Z. Hu, S. Mahadevan, Uncertainty quantification and management in additive manufacturing: Current status, needs, and opportunities, Int. J. Adv. Manuf. Technol. 93 (5) (2017) 2855–2874.

[6] M. Mozaffar, A. Paul, R. Al-Bahrani, S. Wolff, A. Choudhary, A. Agrawal, K. Ehmann, J. Cao, Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks, Manuf. Lett. 18 (2018) 35–39.

[7] M. Roy, O. Wodo, Data-driven modeling of thermal history in additive manufacturing, Addit. Manuf. 32 (2020) 101017.

[8] X. Li, N. Polydorides, Time-efficient surrogate models of thermal modeling in laser powder bed fusion, Addit. Manuf. 59 (2022) 103122.

[9] S. Zhang, A.L. Gain, J.A. Norato, Adaptive mesh refinement for topology optimization with discrete geometric components, Comput. Methods Appl. Mech. Engrg. 364 (2020) 112930.

[10] S. Nikolopoulos, I. Kalogeris, V. Papadopoulos, Non-intrusive surrogate modeling for parametrized time-dependent partial differential equations using convolutional autoencoders, Eng. Appl. Artif. Intell. 109 (2022) 104652.

[11] B. Kramer, K.E. Willcox, Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition, AIAA J. 57 (6) (2019) 2297–2307.

[12] A. Carracedo Rodriguez, S. Gugercin, J. Borggaard, Interpolatory model reduction of parameterized bilinear dynamical systems, Adv. Comput. Math. 44 (6) (2018) 1887–1916.

[13] P. Benner, P. Goyal, Balanced truncation model order reduction for quadratic-bilinear control systems, 2017, arXiv preprint arXiv:1705.00160.

[14] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.

[15] S. Kollmannsberger, M. Carraturo, A. Reali, F. Auricchio, Accurate prediction of melt pool shapes in laser powder bed fusion by the non-linear temperature equation including phase changes, Integr. Mater. Manuf. Innov. 8 (2) (2019) 167–177.

[16] Y. Li, D. Gu, Parametric analysis of thermal behavior during selective laser melting additive manufacturing of aluminum alloy powder, Mater. Des. 63 (2014) 856–867.

[17] C. Liu, C. Li, Z. Zhang, S. Sun, M. Zeng, F. Wang, Y. Guo, J. Wang, Modeling of thermal behavior and microstructure evolution during laser cladding of AlSi10Mg alloys, Opt. Laser Technol. 123 (2020) 105926.

[18] W.J. Little, Tables of Thermodynamic Properties of Argon from 100 to 3000 K, Tech. rep., Arnold Engineering Development Center Arnold AFB TN, 1964.

[19] S. Lucas, Measurement of Cp/Cv for argon, nitrogen, carbon dioxide and an argon+ nitrogen mixture, University College London, 2010.

[20] S. Chen, S. Saxena, Thermal conductivity of Argon in the temperature range 350 to 2500 K, Mol. Phys. 29 (2) (1975) 455–466.

[21] S. Safdar, A.J. Pinkerton, L. Li, M.A. Sheikh, P.J. Withers, An anisotropic enhanced thermal conductivity approach for modelling laser melt pools for Ni-base super alloys, Appl. Math. Model. 37 (3) (2013) 1187–1195.

[22] S. Schmid, J. Krabusch, T. Schromm, S. Jieqing, S. Ziegelmeier, C.U. Grosse, J.H. Schleifenbaum, A new approach for automated measuring of the melt pool geometry in laser-powder bed fusion, Progr. Addit. Manuf. 6 (2) (2021) 269–279.

[23] S. Nikam, H. Wu, R. Harkin, J. Quinn, R. Lupoi, S. Yin, S. McFadden, On the application of the anisotropic enhanced thermal conductivity approach to thermal modelling of laser-based powder bed fusion processes, Addit. Manuf. 55 (2022) 102870.

[24] M. Sheikhi, F.M. Ghaini, H. Assadi, Prediction of solidification cracking in pulsed laser welding of 2024 aluminum alloy, Acta Mater. 82 (2015) 491–502.

[25] K. Ramanathan, S. Yen, High-temperature emissivities of copper, aluminum, and silver, JOSA 67 (1) (1977) 32–38.

[26] J.G.S. Macías, T. Douillard, L. Zhao, E. Maire, G. Pyka, A. Simar, Influence on microstructure, strength and ductility of build platform temperature during laser powder bed fusion of AlSi10Mg, Acta Mater. 201 (2020) 231–243.

[27] M.G. Larson, F. Bengzon, The finite element method: Theory, implementation, and practice, Texts Comput. Sci. Eng. 10 (2010).

[28] V. Silva, P. Salinas, M. Jackson, C. Pain, Machine learning acceleration for nonlinear solvers applied to multiphase porous media flow, Comput. Methods Appl. Mech. Engrg. 384 (2021) 113989.

[29] R. Lung, Y. Wu, D. Kamilis, N. Polydorides, A sketched finite element method for elliptic models, Comput. Methods Appl. Mech. Engrg. 364 (2020) 112933.

[30] I.C. Ipsen, T. Wentworth, The effect of coherence on sampling from matrices with orthonormal columns, and preconditioned least squares problems, SIAM J. Matrix Anal. Appl. 35 (4) (2014) 1490–1520.

[31] Y. Aizenbud, A. Averbuch, Matrix decompositions using sub-Gaussian random matrices, Inform. Inference: J. IMA 8 (3) (2019) 445–469.

[32] Xiaohan3795, Fast_heat_transfer_simulation_for_laser_powder_bed_fusion, 2023, URL https://github.com/Xiaohan3795/Fast_heat_transfer_simulation_for_laser_powder_bed_fusion.

[33] L.M. van den Bos, B. Sanderse, W. Bierbooms, Adaptive sampling-based quadrature rules for efficient Bayesian prediction, J. Comput. Phys. 417 (2020) 109537.

[34] P. Keast, Moderate-degree tetrahedral quadrature formulas, Comput. Methods Appl. Mech. Engrg. 55 (3) (1986) 339–348.

[35] P. Foteinopoulos, A. Papacharalampopoulos, P. Stavropoulos, On thermal modeling of additive manufacturing processes, CIRP J. Manuf. Sci. Technol. 20 (2018) 66–83.

[36] L. Chen, H. Li, S. Liu, S. Shen, T. Zhang, Y. Huang, G. Zhang, Y. Zhang, B. He, C. Yang, Simulation of surface deformation control during selective laser melting of AlSi10Mg powder using an external magnetic field, AIP Adv. 9 (4) (2019) 045012.