THE UNIVERSITY *of* EDINBURGH

Surrogate models for laser powder bed fusion digital twins

Xiaohan Li

Doctor of Philosophy

School of Engineering

The University of Edinburgh

November 15, 2023

Declaration

I hereby declare that this thesis has been written by my own, and that the contents presented herein are solely my own work, unless explicitly acknowledged otherwise within the text. Furthermore, I affirm that this work has not been previously submitted for the purpose of obtaining any other academic degree or professional qualification.

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. While the nonlinear heat equation in the context of laser powder bed fusion is numerically solved by the finite element method, three time-efficient surrogates are proposed as fast alternatives with different trade-offs between model accuracy, robustness, offline preparation, and online execution time. The first one is the reduced Gaussian process surrogate, which is a data-driven model equipped with a nonlinear dimension reduction scheme. It outperforms in real-time execution online managing to predict temperature profiles almost instantly, though it is comparably less accurate, not robust to random anisotropy, and requires offline preparation of data generation, nonlinear dimension reduction, and training. The second one is the sketched surrogate with data-driven local projection. It projects the accurate but high-dimensional finite element method solution with a low-dimensional basis formed by subsampled training temperatures and then bypasses the majority of costly computations for the temperature-dependent matrices in the projected model by randomized sketching. It is the most accurate surrogate while lacking robustness, necessitating the same offline preparation, and taking more time compared with the first surrogate. The third one is the sketched surrogate with online local projection. Its 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 with-

out replacement based on approximate optimal sampling distributions. Both the projection and the sketching are designed to implement alongside the printing process, which makes this surrogate capable of handling different process parameters without requiring prior computations offline. The third surrogate, therefore, is accurate, robust, and requires no offline preparation, although it entails longer online execution time compared to the other two surrogates. A series of numerical experiments are carried out to present and compare the performance of the three surrogates, which assumes a two-layer printing process with a fixed laser beam trajectory using different printing attributes (laser power and scan speed) and arbitrary thermal conductivity anisotropy. All three surrogates are also principally feasible in other thermal-driven additive manufacturing to obtain better quality assurance with techniques like uncertainty management and closed-loop control.

Lay summary

In laser powder bed fusion, accurately modelling temperature distribution and phase transitions is crucial for ensuring the quality of the 3D printing process. The digital twin technology aims to create fast and accurate simulators that can replicate the printing process and improve its outcomes. Three alternatives to the traditional method of solving the heat equation have been proposed to reduce computation time. The first alternative is a reduced Gaussian process model, which uses data-driven techniques and dimension reduction. It can quickly predict temperature profiles in real-time, but it is slightly less accurate, not robust to random variations, and requires some offline preparation. The second alternative is the sketched surrogate with data-driven local projection. It projects the detailed but high-dimensional solution onto a lower-dimensional basis using selected training temperatures. By using randomized sketching, it reduces the computational workload. This model is the most accurate but lacks robustness, requires the same offline preparation as the first alternative, and takes more time to compute. The third alternative is the sketched surrogate with online local projection. It generates projection bases during the simulation by combining previous temperature profiles with specialized Gaussian functions, while its sketching process uses efficient sampling techniques. This model is accurate, robust, and does not require offline preparation, but it takes longer to compute compared to the other two alternatives. Numerical experiments are conducted using a two-layer printing process with different laser power, scan speed, and thermal conductivity variations. These three alternatives can also be applied to other thermal-driven additive manufacturing processes to enhance quality assurance using techniques like uncertainty management and closed-loop control.

Acknowledgement

I would like to thank my supervisor Dr. Nick Polydorides for his great help in my PhD project. I am also very grateful for the financial support by the Principal's Career Development and Edinburgh Global Research Scholarships from the School of Engineering at the University of Edinburgh, UK.

Contents

Declaration i

Abstract ii

Lay summary iv

Acknowledgement v

List of Figures x

List of Tables xiii

1 Introduction 1

1.1 Motivation 1

1.2 Objectives 3

1.2.1 Time-efficient surrogates with different advantages 3

1.3 Dissertation outline and notation 5

2 Literature Review 7

2.1 Laser powder bed fusion 7

2.1.1 Introduction to laser powder bed fusion 7

2.1.2 Modelings of LPBF 9

2.2 Thermal modelling of LPBF 12

2.3 Digital twin of AM 14

2.4 Surrogate model 16

3 Nonlinear Thermal Model of LPBF 20

3.1 Governing equations 21
3.2 Temperature-dependent thermal properties 22
3.2.1 Anisotropy of thermal conductivity 22
3.2.2 Latent heat 23
3.2.3 Polynomial fitting 23
3.2.4 Thermal properties of powder 24
3.3 The full-order numerical solver with FEM 24
3.4 Time consumption due to high dimensionality and nonlinearity . . 28

4 Reduced Gaussian process surrogate 32
4.1 Subsampling 33
4.1.1 Normalization and Featuring 33
4.1.2 Relative distance predictor 35
4.2 High-dimensional temperature prediction 37

5 Sketched surrogate with data-driven local projection 39
5.1 Subsampling-based local projection 40
5.2 Randomized sketching 41
5.3 High-dimensional temperature prediction 44

6 Sketched surrogate with online local projection 46
6.1 Randomized sketching with approximate sampling probability . . 48
6.1.1 Approximate sampling probability 50
6.2 Gaussian local projection 52
6.2.1 Online generation of projection basis 53
6.2.2 Online projection 55

7 Results 58
7.1 The thermal simulator with FEM 58
7.1.1 Parameter setup and validation 58
7.1.2 Thermal behaviours of LPBF 62
7.2 Reduced Gaussian process surrogate 68
7.2.1 Data generation and parameter setup 68

7.2.2 Dimension reduction 69
7.2.3 Model accuracy and robustness 70
7.2.4 Offline preparation 76
7.2.5 Time cost reduction 77
7.3 Sketched surrogate with data-driven local projection 78
7.3.1 Parameter setup 78
7.3.2 Model accuracy and robustness 79
7.3.3 Offline preparation 84
7.3.4 Time cost reduction 84
7.4 Sketched surrogate with online local projection 85
7.4.1 A simplified example 85
7.4.2 Parameter setup 87
7.4.3 Model accuracy and robustness 88
7.4.4 Offline preparation 93
7.4.5 Time cost reduction 93
7.5 Comparison and discussion 94
7.5.1 Model accuracy 94
7.5.2 Robustness 96
7.5.3 Time cost and offline preparation 96
7.5.4 Different strengths of each surrogate 97

8 Conclusion **99**

References **104**

List of Figures

2.1 The schematic of LPBF. 8
2.2 The connections between computational models of LPBF. 9
2.3 The schematic of DT operation. 15

3.1 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. 20

4.1 The framework of the reduced GP surrogate predicting high-dimensional temperatures with three control inputs (laser power, scan speed and time) via linear combinations of subsampled training temperatures. 32

5.1 The framework of the sketched surrogate with data-driven local projection where high-dimensional temperatures of given test inputs (laser power, scan speed, and time) are reconstructed by the projected results produced from the surrogate with subsampling-based local projection and randomized sketching. 39

6.1 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) can not be described as (anisotropic) Gaussian functions. 47

6.2 The framework of the sketched surrogate with online local projection where Gaussian local projection bases are generated online and randomized sketching is achieved by approximate optimal sampling probability. 48
6.3 The sketching procedure to compute the projected and sketched mass matrix. 49
6.4 The relative errors with different percentages p. 51
6.5 The projection procedure to compute the projected and sketched mass matrix. 53

7.1 The laser scanning pattern in [13] where position 1 and 2 used in validation are respectively the centre of the first and second layer. 60
7.2 The scanning pattern and the selected positions of the two-layer domain. 62
7.3 Temperature profiles at χ_1 in all six tests. 64
7.4 Temperature profiles at χ_2 in all six tests. 65
7.5 Temperature profiles at χ_3 in all six tests. 65
7.6 Temperature profiles at χ_4 in all six tests. 66
7.7 Thermal histories at the four selected positions in all tests. 66
7.8 The boxplot of relative errors between FEM and $\mathcal{F}_1$. 70
7.9 The boxplot of relative errors between FEM and $\mathcal{F}_1$ without the first five time steps of each scanning line. 71
7.10 The melt pool comparison between FEM and $\mathcal{F}_1$ in the test $\mathcal{S}_8$. . 73
7.11 The melt pool comparison between FEM and $\mathcal{F}_1$ in the test $\mathcal{S}_9$. . 74
7.12 The thermal history comparison between FEM and $\mathcal{F}_1$ in $\mathcal{S}_8$. . . . 75
7.13 The thermal history comparison between FEM and $\mathcal{F}_1$ in $\mathcal{S}_9$. . . . 75
7.14 The relative errors between FEM and $\mathcal{F}_2$. 79
7.15 The relative errors between FEM and $\mathcal{F}_2$ without the first 5 time steps of each scanning line. 79
7.16 The melt pool comparison between FEM and $\mathcal{F}_2$ in the test $\mathcal{S}_8$. . 81
7.17 The melt pool comparison between FEM and $\mathcal{F}_2$ in the test $\mathcal{S}_9$. . 82
7.18 The thermal history comparison between FEM and $\mathcal{F}_2$ in $\mathcal{S}_8$. . . . 84

7.19 The thermal history comparison between FEM and $\mathcal{F}_2$ in $\mathcal{S}_9$. . . . 84
7.20 Relative errors between FEM and SM. 87
7.21 The comparison of temperature distribution between FEM and SM. 87
7.22 The boxplot of relative errors between FEM and $\mathcal{F}_3$. 89
7.23 The melt pool comparison between FEM and $\mathcal{F}_3$ in the test $\mathcal{S}_8$. . 90
7.24 The melt pool comparison between FEM and $\mathcal{F}_3$ in the test $\mathcal{S}_9$. . 91
7.25 The thermal history comparison between FEM and $\mathcal{F}_3$ in $\mathcal{S}_8$. . . . 93
7.26 The thermal history comparison between FEM and $\mathcal{F}_3$ in $\mathcal{S}_9$. . . . 93

List of Tables

2.1 The list of some commonly-used metal materials in LPBF. 8
2.2 Inputs and outputs of the computational models of LPBF [8, 10]. 11

3.1 The integral definition of matrices in FEM and their Gaussian quadrature approximation. 26
3.2 The definition of matrices in Gaussian quadrature approximation. 27

7.1 The fitted polynomial functions of thermal properties [13, 119–122]. 59
7.2 Parameters in thermal properties. 60
7.3 The validation of the printing process with laser power 250W and scan speed 200mm/s. 61
7.4 Model parameters in the thermal model of LPBF [13, 14, 108, 123]. 62
7.5 Parameters of the 6 examples. 64
7.6 Highest temperatures and melt pool sizes at the selection positions in all tests. 67
7.7 Parameters of the 16 tests. 69
7.8 Parameters in $\mathcal{F}_1$. 69
7.9 The dimension reduction in $\mathcal{F}_1$. 70
7.10 The accuracy comparison of $\mathcal{F}_1$ between using a static or random λ. 72
7.11 The comparison of melt pool sizes between FEM and $\mathcal{F}_1$ in the test $\mathcal{S}_8$ and $\mathcal{S}_9$. 76
7.12 The offline time cost of $\mathcal{F}_1$. 77
7.13 The average execution time of FEM and $\mathcal{F}_1$. 78
7.14 Parameters of $\mathcal{F}_2$. 78
7.15 The accuracy comparison of $\mathcal{F}_2$ between using a static or random λ. 80

7.16 The comparison of melt pool sizes between FEM and $\mathcal{F}_2$ in the test $\mathcal{S}_8$ and $\mathcal{S}_9$. 83
7.17 The time cost reduction of $\mathcal{F}_2$. 85
7.18 Thermal model parameters in the simplified example. 86
7.19 The surrogate model parameters in the simplified example. 86
7.20 The comparison of average time cost for the simplified example. . 88
7.21 Parameters of $\mathcal{F}_3$. 88
7.22 The accuracy comparison of $\mathcal{F}_3$ between using a static or random λ. 89
7.23 The comparison of melt pool sizes between FEM and $\mathcal{F}_3$ in the test $\mathcal{S}_8$ and $\mathcal{S}_9$. 92
7.24 The time cost reduction of $\mathcal{F}_3$. 94
7.25 The comparison of model accuracy and robustness. 95
7.26 The comparison of the online execution time. 97

Chapter 1

Introduction

1.1 Motivation

Additive manufacturing (AM), popularly known as three-dimensional (3D) printing, has been one of the fast-emerging technologies expected to drive a revolution in prototyping and manufacturing at home, in a lab, and in the office. Since the first 3D printer using stereolithography was produced by Charles Hall in the 1980s, the common novelty of AM is successively adding materials instead of removing them as in traditional manufacturing [1]. It ultimately transforms a computer-aided design (CAD) into the layer-upon-layer fabrication of a three-dimensional object with various materials like polymers, ceramics, and metals [2]. Since the object is fabricated additively, AM enjoys the superiority in design freedom resulting in increasingly wide applications in many fields such as aerospace [3], automotive [4], food [5], and medical industry [6].

While AM outperforms in realizing complicated and personalized designs, it faces the challenge of lacking quality assurance. The quality of printed parts is not consistent, and the proper values for process parameters mainly rely on "trial-and-error" efforts. These issues result in time and material waste and become essential obstacles hindering the full potential of AM. One commonly recognized way to address the problem is the digital twin (DT). It is a technology expected to facilitate smart industries in the era of Industry 4.0 by promoting process productivity and part quality [7]. Instead of being an isolated digital

model, it is a virtual replica with successive updates based on sensor data received from the physical twin, and it in turn makes proper decisions on diagnostics and control. This is a two-way communication between a physical twin and a digital twin which are respectively a printing process and correspondingly mechanical models in the context of AM [8]. While high-fidelity mechanical models are critical in reconstructing and understanding the dynamics in AM, they are also computationally expensive. In DT, however, large and/or fast implementations of these digital models are required for applications such as finding optimal prior values for process parameters and calibrating the digital model with sensor data. Therefore, it is a bottleneck problem for the DT framework to develop practical, accurate, and fast-solving surrogates for mechanical models.

Laser powder bed fusion (LPBF), as a thermal-driven metal AM, has intrinsically cyclic thermal behaviours with steep thermal gradients, high cooling rates, and intricate thermal histories, all of which would affect the final part's microstructure and mechanical properties [9]. The computational models of LPBF are categorized as powder bed model, heat source model, melt pool model, solidification model, residual stress model, and other macro-level models [8, 10]. The thermal modelling of LPBF is a critical mechanical model. On the one hand, it integrates the model of the powder bed, heat source, and melt pool. While the information on a powder bed and a heat source are hard to be explicitly obtained during a printing process, we can use temperature or melt pool size measurements to calibrate the thermal model to reduce or eliminate the uncertainty caused by all three categories of models in thermal modelling. The process of calibration, however, requires a large number of forward implementations of the thermal model, which increases the demand for fast-solving surrogates. On the other hand, the results of thermal modelling like temperature profiles, thermal histories, and melt pool sizes are required for the rest of the micro- and macro-scale models. Therefore, it is also necessary for other models to develop accurate and fast-computed surrogates of thermal modelling to facilitate the timely analysis and decision-making based on all quantities of interest. All in all, it is a meaningful research area to properly build a high-fidelity thermal model and design its time-efficient surrogates to dwell in the DT framework of LPBF. Since our aim is set to reduce

the online execution time, it is expected that there will be a compromise in model accuracy, robustness, and offline preparation. To cope with different scenarios, we determine to design the surrogates with different emphases. In specific, one surrogate is aimed to fulfil real-time implementation. Another surrogate is designed to keep a high level of accuracy. At last, we want a surrogate focusing on robustness and less offline preparation. Though they are observed with different advantages, the criterion of a valid design remains to maintain satisfactory accuracy while requiring less online running time.

1.2 Objectives

1.2.1 Time-efficient surrogates with different advantages

The thermal model of LPBF is first numerically solved by the finite element method (FEM) as a reference to evaluate the performance of designed surrogates. The thermal model of LPBF should simulate heat transfer during the fabrication of an object. The powder bed is paved layer after layer on a building platform which is temperature-controlled to improve the final object's quality like mechanical properties, microstructure, and residual stress. Following the trajectory pre-determined by a sliced CAD design, a laser beam selectively melts and fuses metal alloy powders under the protection of an inert gas atmosphere [1, 11]. As the printing process carries on, some heat is absorbed by powders yielding melt pools while heat loss due to convection and radiation occurs on the top and surrounding surfaces. Consequently, we model the heat transfer described above as a heat equation with Neuman boundary conditions including heat flux and heat loss, and a Dirichlet boundary condition imposing a constant bottom surface. The powder bed domain is modelled as a 3D cuboid-shape spatial domain which is vertically extended layer upon layer. When a new layer is spread, the information of remained temperature and material phases in all previous layers should be kept to ensure the continuity of printing. Moreover, it is necessary to build a high-fidelity simulation with additional designs related to the nonlinearity due to temperature-dependent thermal properties, the latent heat

effect, and the anisotropy due to the convection effect in melt pools [12–16]. Our thermal model should present the characteristics of LPBF and retain adjustable parameters including physical and nominal parameters such as laser power, scan speed, beam size, hatch space, scan path, building platform temperature, layer thickness, thermal properties, the scale factor of anisotropy, energy absorptivity, and ambient temperature. FEM, as a traditional and high-fidelity way to work out heat equations, is applied to build the numerical solver where the temporal domain is finely discretized with a small time interval and the spatial domain is discretized as tetrahedrons. To reduce computational complexity while maintaining accuracy, a mesh scheme with refinement is developed with fine mesh around scanning trajectories and coarse mesh elsewhere [17]. The heat equation, however, is still high-dimensional in order to accurately capture thermal behaviours. The heat equation is also nonlinear due to temperature-dependent thermal properties and boundary conditions. Hence, to cope with the nonlinearity, Picard iteration should be applied to converge to the final temperature result [18]. In short, we should design a numerical solver with FEM to properly discretize the heat equation and solve it iteratively with nonlinear loops nested within time loops, the model accuracy of which should be validated by published experimental data.

Surrogates are designed to improve time efficiency compared with FEM. While less online execution time is required, there will be some compromise in other aspects such as model accuracy, robustness, and offline preparation. In the context of thermal modelling in LPBF, model accuracy considers the accuracy of temperature distributions, melt pool sizes, and thermal histories. The robustness, however, means the ability to handle the random anisotropy of thermal conductivity yielding random melt pool size. In reality, the melt pool size at different time instants varies with some randomness even with the same printing parameters. Therefore, robustness is an important performance as the surrogate being robust in arbitrary thermal conductivity anisotropy can be more practical to model a real printing process. Offline preparation indicates the work needed to be accomplished before the simulations start. Generally, offline preparation aims to facilitate less online execution time and/or higher accuracy. However, it is also an advantage if a surrogate has no offline preparation required, which means that

the surrogate requires no offline cost in either time or storage.

Each surrogate model individually balances the trade-off between running time, model accuracy, robustness, and offline preparation. Therefore, we aim to build three surrogates, all of which require less online execution time but respectively bias on real-time implementation, high accuracy, and strong robustness without offline preparation. The first surrogate focus on real-time implementation. It can be less accurate, not robust, and require offline preparation, but its online implementation should be finished instantly. This surrogate is designed to swiftly produce rough temperature descriptions, which is suitable for applications with comparably low accuracy requirements. The second surrogate focus on model accuracy. It needs to keep high accuracy, but the compromise in online execution time, robustness, and offline preparation is acceptable. This surrogate targets the application scenarios where accuracy matters most. The third surrogate focus on robustness and no offline preparation. Though it has some compromises in model accuracy and online execution time, it is competent in handling arbitrary thermal conductivity anisotropy and saves the cost of offline preparation. This surrogate aims to be more practical in simulating a real printing process. Though the three surrogates have different advantages, they all need less online execution time while maintaining reasonable accuracy.

1.3 Dissertation outline and notation

The dissertation contains 8 chapters: introduction, literature review, nonlinear thermal model of LPBF, reduced Gaussian process surrogate, sketched surrogate with data-driven local projection, sketched surrogate with online local projection, results, and conclusion. In the chapter of introduction, we express what motivates us to develop time-efficient surrogates for the digital twin of LPBF. The objectives of this thesis are also delineated in the introduction part, which includes a reference numerical solver and three time-efficient surrogates with different advantages. The literature reviews are outlined in chapter 2, which contains the literature about laser powder bed fusion, thermal modelling of LPBF, digital twin of AM, and surrogate model. The third chapter describes the nonlinear thermal

model of LPBF. The nonlinear heat equation is set up in the context of LPBF where temperature-dependent thermal properties, phase changes, the anisotropy of thermal conductivity, and latent heat are considered. Correspondingly, the full-order numerical solver with FEM is developed with Gaussian quadrature approximation and Picard iteration. Three time-efficient surrogates are respectively explained in chapter 4, 5, and 6. In chapter 4, reduced Gaussian process surrogate is described by subsampling and a swift high-dimensional temperature predictor. Sketched surrogate with data-driven local projection is then introduced in chapter 5 where the three subsections respectively show subsampling-based local projection, randomized sketching, and high-dimensional temperature prediction. Another sketched surrogate is shown in chapter 6. It is a sketched surrogate with online local projection where Gaussian local projection bases are generated online and randomized sketching is implemented with approximate sampling probability. The results of the three surrogates are detailed in chapter 7 which include the performance comparison in model accuracy, robustness, offline preparation, and time cost reduction. The methods explained from chapter 3 to chapter 6 have been published either in [19] or in [20]. A simplified example of the sketched surrogate with online local projection is given with codes provided in the Github repository [21], the evaluation of which is detailed in section 7.4.1 of the result chapter and has been published in [20]. The rest of the results chapter, however, is different from the two published papers to let the performance of the three surrogates be compared under the same parameter settings. Conclusions are then detailed in chapter 8 where the performance of the three surrogates is summarized and some suggestions for their applications are given.

In this thesis, the matrices are presented in capital letters. For example, we denote a matrix as X, the i-th row, the i-th 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 i-th entry of x denoted as x_i. Continuous quantities are also specified in lowercase letters and tensors are denoted as $\vec{x}$.

Chapter 2

Literature Review

2.1 Laser powder bed fusion

2.1.1 Introduction to laser powder bed fusion

Generally, there are seven types of AM: vat photopolymerization (VPP), material extrusion (MEX), material jetting (MJT), binder jetting (BJT), powder bed fusion (PBF), sheet lamination (SHL), and direct energy deposition (DED), among which PBF is a prevalent metal AM technique selectively melting and fusing powders in the powder bed paved layer by layer [1]. LPBF, however, is a sub-category of PBF using a laser instead of an electron beam as the heat source. It is a prevalent AM technique in metal fabrication with superiority in realizing complex or personalized design resulting in increased application in fields like aerospace [3], automotive [4], and medical industry [6]. LPBF successively spreads and selectively melts thin layers of powder to fabricate physical parts from 3D geometrical designs [1]. In specific, a sequence of 2-dimensional layers is generated by slicing the CAD of a 3D object. One layer of powder is evenly paved on the temperature-controlled building platform. Under the protection of an inert gas atmosphere, the laser beam selectively scans the top of the powder bed with the trajectory predetermined by the slicing software. Then, the platform declines for the height of a layer thickness, and a new layer of powder is recoated above the previous layers. After repeating this cycle, the printed part is formed by fusing the melted material of adjacent trajectories and layers [22].

The schematic of LPBF is illustrated as figure 2.1.

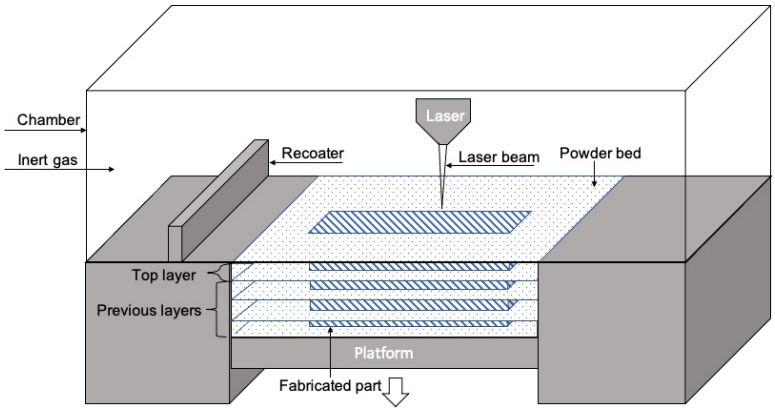


Figure 2.1: The schematic of LPBF.

As one promising technique of metal AM, a lot of research in material development has been undertaken. Numerous metal materials can be employed in LPBF, such as alloys [13, 23–32], steels [33–36], and superalloys [37–41]. Some commonly-used metal materials in LPBF are listed in table 2.1. Despite the

Table 2.1: The list of some commonly-used metal materials in LPBF.

Material type	Composition	Reference
Aluminum alloy	AlSi7Mg	[23]
	AlSi10Mg	[13, 24]
	Al-12Si	[25]
	Al-20Si	[26]
Titanium alloy	Ti6Al4V	[27, 28]
	TiAl	[29, 30]
Magnesium alloy	ZK60	[31]
	AZ91D	[32]
Steel	316L	[33, 34]
	17-4SS	[35]
	H11	[36]
Superalloy	IN718	[37]
	IN625	[38, 39]
	IN738LC	[40]
	247LC	[41]

choice of printing material, various shielding gases such as argon, nitrogen, and helium are commonly used to protect the printed part from oxidation [38, 42]. Like other types of AM, LPBF has the advantage of fabricating parts with designs deemed to be impractical or cumbersome in traditional manufacturing [43].

Though it enjoys the ability to fabricate parts with high complexity, one of the main challenges inhibiting the full potential of LPBF is the inconsistency of part quality [44]. The part quality is affected by various process parameters such as laser power and scan speed. Lee et al. proposed a model-free optimal controller to reach a target temperature by adjusting scan speed [45]. In [46], an adaptive control system was developed with a finite difference model of the heat equation to control melt pool sizes by adjusting laser power. To avoid the "trial-and-error" effort in process optimization and control, it is essential to build models and simulations to deepen the understanding of the physics in LPBF. Though LPBF is a conceptually straightforward process, the modelling of underlying physics is still intricate and of large spatial and temporal scale [47].

2.1.2 Modelings of LPBF

There are extensive researches about the LPBF's modelling and simulation which are mainly divided into five types: powder bed model, heat source model, melt pool model, solidification model, and residual stress model [8, 10]. The connections between these six categories of models are shown in figure 2.2. Powder bed

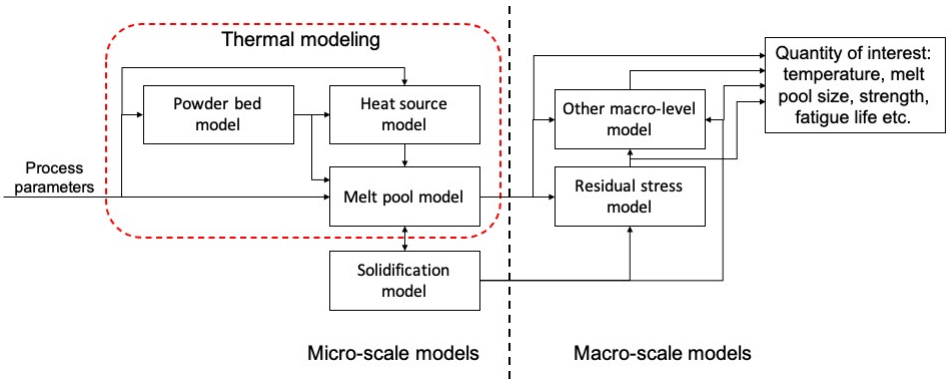


Figure 2.2: The connections between computational models of LPBF.

models are aimed to produce packing structures of powder beds like powder bed porosity, packing density, and radial distribution based on inputs like the size and shape of powder, layer thickness, re-coater shape, and velocity [48, 49]. Interactions occurring between individual powder particles such as adhesion, collision, and friction are modelled under the consideration of diverse forces such as friction, gravity, elastic forces, and van der Waal forces [50, 51]. Numerically, the discrete

element method (DEM) [50, 51] and the raindrop packing algorithm [52, 53] are utilized under the assumption of spherical particle shape and a particle size distribution such as Gaussian [50], uniform [51], monosized [53], or bimodal [52]. It is essential to properly establish a heat source model since it affects the melt pool geometries as well as the final parts' mechanical properties. A laser beam can be approximated as a two-dimensional surface heat source on the top surface such as an ellipsoidal heat source decaying exponentially [12], or a Gaussian heat source with normally distributed intensity [13, 54]. It can also be simulated as a volumetric heat source considering the penetration of laser scanning. Various three-dimensional geometries can be employed to model a heat source such as cylindrical [55], semi-spherical [56], semi-ellipsoidal [56], and conical shapes [57]. The melt pool model governed by a heat equation with corresponding boundary conditions describes the transient thermal fields and simulates the melt pool characteristics. In [58], an analytical solver was proposed to model thermal histories using specific enrichment and compensation functions. FEM is one of the most popular numerical solvers of heat transfer in LPBF, which provides high-fidelity results with fine spatial and temporal discretization [13, 59]. Another numerical solver of melt pool models is coupling FEM and finite volume method (FVM) with the software tool ALE3D employing arbitrary Lagrangian-Eulerian (ALE) techniques [60, 61]. While the powder bed model gives the porosity distribution and the heat source model gives the distribution of absorbed energy, the thermal model based on the above-mentioned two distributions produces temperature fields over the spatial and temporal domain which then become the inputs of solidification and residual stress model either directly by thermal histories or indirectly by melt pool sizes [10]. The solidification model depicts the formation of grain structure during the solidification process yielding microstructure characteristics including grain size, grain texture, and grain morphology [62]. With the information on melt pool geometries, temperature gradients, and cooling rate, the cellular automata (CA) and phase field (PF) methods are commonly used to model grain structures. Zhang et al. developed a predictive model by coupling cellular automaton and finite element method where coarse-scale temperatures produced by the finite element (FE) model were interpolated to the scale needed

for the CA model and the FE-CA model generated the evolution of grain morphology occurring as the cooling of deposition [63]. Azizi et al. studied grain growth with a multi-order parameter type PF model in the context of LPBF, the underlying thermal conditions of which considered different build directions [64]. Residual stress is induced by heating and cooling cycles with high thermal gra-

Table 2.2: Inputs and outputs of the computational models of LPBF [8, 10].

Model	Inputs	Outputs
Powder bed	Restitution coefficient Damping coefficient Young's modulus Sliding friction coefficient Rolling friction coefficient Radius distribution	Packing density Powder bed porosity Particle size distribution Particle radial distribution
Heat source	Beam diameter Laser power Scan speed Powder distribution Powder shape	Absorbed energy Vertical absorption distribution
Melt pool	Laser power Scan speed Absorption energy Thermal properties Layer thickness Preheating temperature Particle radial distribution	Melt pool size Thermal history Porosity Layer bonding defects
Solidification	Cooling rate Thermal history Melt pool size	Grain size Thermo-mechanical properties
Residual stress	Thermal history Thermo-mechanical properties	Residual stress Shrinkage Deformation Fatigue life

dients and rapid cooling, which may cause cracks and deformation of final parts. It is modelled either by simple equilibria of force and moment [65] or the thermo-mechanical analysis based on FEM [66, 67]. The accuracy of the residual stress model depends on the thermal history and microstructure while the accuracy of the microstructure depends on the melt pool geometries, thermal gradient, and

cooling rate. We thus can tell that the errors of the thermal model will propagate to the modelling and simulation of the properties of the final parts. The inputs and outputs of each computational model of LPBF are listed in table 2.2.

2.2 Thermal modelling of LPBF

The thermal modelling of LPBF should describe the transient heat transfer during rapid heating and cooling cycles of a printing process, from which thermal histories and melt pool sizes will be characterized. The thermal model is governed by a heat equation under the context of LPBF, the numerical domain of which is a layerwise-increased cuboid to simulate a powder bed paved layer after layer. A Dirichlet condition keeps the bottom surface constant to simulate the temperature-controlled building platform. Other surfaces are imposed with a Neumann boundary condition that considers heat source and heat loss. As the temperature changes are driven by the laser beam, the heat source model describing the heat flux caused by a laser beam is critical. Distinguished by whether the penetration depth is considered, the laser beam is modelled as either a volumetric or surface heat source. Foroozmehr et al. adopted a volumetric heat source where the laser beam penetrated the porous powder bed with an effective optical penetration depth [55]. The surface heat source serves as a Neumann boundary condition of the heat equation. In [13, 68], the heat flux density of the laser beam was deemed to obey a Gaussian distribution. For a more complicated surface heat source with anisotropy, Goldak et al. proposed a double-elliptical model where there are front and rear quadrants with different heat fractions and effective radius [56]. The heat loss, however, occurs on the surrounding and top surfaces, which includes heat convection and radiation [13].

The heat equation is nonlinear, the nonlinearity of which is introduced by both the radiation heat loss and the thermal properties including thermal conductivity, density, and specific heat capacity. These thermal properties depend on materials, phases, and temperatures. Experimental data on the thermal properties of metals are available so that we can model the variation of thermal properties in terms of temperature. In [14], thermal properties of AlSi10Mg were fitted by piecewise

polynomials separated as three temperature ranges respectively representing the solid, mushy, and liquid states. To consider the latent heat of the isothermal phase change between solid and liquid, Kollmannsberge et al. characterized the jump of specific heat capacity between two phases by a phase change function where the smoothness was controlled by a calibratable parameter and the specific value of latent heat was provided by experiments [12]. Another phase of metals in LPBF is powder, the effective thermal properties of which can be represented by the thermal properties of metals and shielding gas, and the porosity of the powder bed. The powder bed porosity as the volumetric fraction of void in the entire domain is employed to connect the temperature-dependent thermal properties of powder with the corresponding thermal properties of solid and inert gas. In [69, 70], the thermal conductivity of porous material was characterized by the Maxwell and the Bruggeman models using the thermal conductivity of solid and porosity. The density and specific heat capacity of powder bed can be approximated by the weighted sum of solid and inert gas according to their corresponding volumetric fractions [13, 14, 71, 72].

One major criticism of heat transfer simulation which is also a primary difference between a thermal model and computational fluid dynamics (CFD) models is the neglection of Marangoni convection in melt pool dynamics. In the review presented by Shu et al., the necessity to consider the effect of convection, especially the Marangoni convection, was shown as it makes the results consistent with experiments [73]. To mimic the Marangoni convection effect, Karayagiz et al. artificially increased the thermal conductivity of the material in the liquid phase [74]. An anisotropic enhanced thermal conductivity model was applied and validated in [75] which used directional correction factors to manipulate the thermal conductivity of liquid phase in the three Cartesian directions. There are a considerable number of existing research fulfilling numerical solvers of the thermal modelling of LPBF, some of which are validated by experiments. In [13], nonlinear thermal modelling of SLM was implemented where AlSi10Mg powder was melted in an argon atmosphere. The simulation solved by FEM was followed by experimental validations, and the variation of melt pool size with different laser power and scan speed was analyzed. Stefan et al. supplemented the anisotropy

of thermal conductivity in liquid phase [12]. A different material IN625 was utilized, and the experimental data set in AM benchmark marked as AMB2018-02 was employed as the reference for accuracy validation. It was shown that this thermal model was more predictive than the thermal models without the consideration of anisotropy. In LPBF, heat transfer in solid area is determined by heat conduction, while in liquid area heat is transferred through a combination of heat conduction and convection. Heat convection is the heat transport facilitated by the fluid movement which primarily results from temperature-induced surface tension gradients and the recoil pressure generated by the fluid evaporation at the fluid's free surface. A full CFD modelling is thus required to simulate the heat transfer of LPBF [73]. In this thesis, the thermal modelling accommodates the heat convection in liquid phase with the anisotropic enhanced thermal conductivity model proposed in [12, 75] which emulates heat convection by scaling the liquid thermal conductivity in the three Cartesian directions with three positive weights.

2.3 Digital twin of AM

Initially, DT is employed by NASA to predict the fatigue failure of critical components in 2003 [7]. It replicates a real system with a multi-physical and multi-variable simulation, the concept of which stretches from a single product to a machine or an entire environment of production over time [76]. For the elementary level, a digital model runs according to the sensor data from a real machine. The virtual replica at this level, also called digital shadow, is only a mono-directional interaction from a physical object to a digital model [77]. Ladj et al. proposed a digital shadow absorbing expert knowledge and artificial intelligence to characterize the operation of the physical system, the application of which was provided with an example in the aeronautical machining industry [78]. For the advanced level, a bi-directional interaction is established where data from a digital shadow is also delivered to its physical twin. Therefore, the loop of virtual feedback is closed and a full DT is formed. A DT at this level can be used in process optimization, defect detection, and achieving self-calibration [76]. Stojanovic et. al

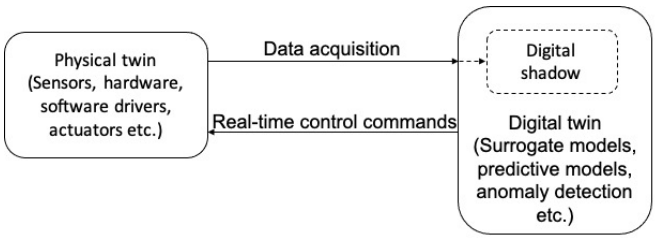


Figure 2.3: The schematic of DT operation.

presented a data-driven DT approach to enable root cause analysis and process control in multi-dimensional and large process space, in which a 3D laser cutting process was used as a showcase [79]. Tancredi et al. integrated machine learning algorithms into a DT of a real industrial plant to enable anomaly detection in industrial system operation and enhance employee safety [80]. The mono- and bi-directional interaction in digital shadow and digital twin is shown in figure 2.3.

In the era of Industry 4.0, intelligent decisions are made for production systems via real-time communication and cooperation to facilitate the mass and efficient production of high-quality personalized products [81]. Both AM and DT are two key concepts in Industry 4.0, the combination of which is an active research area to release the innovative potential of digital manufacturing. The fabrication process of AM involves complicated physical phenomena, and the quality of printed parts is affected by the process variables of the 3D printer. The number and size of key-hole pores are expected to increase with higher laser power, while the formation of which is restrained by a higher scan speed [82]. The defect of lack-of-fusion porosity can be overcome by narrowing hatch space, reducing layer thickness, lower scan speed, and higher laser power [83]. Residual stress/distortion resulting from large thermal gradients and fast cooling rates during the layer-wise solidification of LPBF can be relieved by higher laser power, lower scan speed, and a reduction of build plan area [84]. Therefore, the process optimization and efficient coordination of AM is profitable to fabricate structurally sound objects. However, it is time- and resource-consuming to optimize the AM process with trial-and-error efforts. Instead, we can alter printing attributes according to fast-computed high-fidelity models that simulate how process parameters affect the object's properties. These models upon validation are the components of the DT

of AM [85]. For the DT of LPBF, a numerical framework is thereby established to estimate the interaction between powders, melt pool sizes, heating and cooling rates, solidification parameters, the evolution of residual stress, and the formation of defects, etc [60, 86].

Though high-fidelity numerical solvers of digital models produce accurate simulations and predictions of a LPBF system, they are computationally intensive and thereby not applicable to connect with real-time sensor data and make intelligent decisions in time. The computational speed of digital models is essential in both realizing the one-way communication in a digital shadow to simulate the physical system with real-time sensor data and the two-way communication in a digital twin to perform corrective control commands, which excludes the option of utilizing high-fidelity numerical solvers and makes the fast-solving yet accurate enough alternatives become a fundamental part of developing a DT framework [87].

2.4 Surrogate model

A surrogate model (SM) is an alternative to computer models, which is normally swift in computation while remaining accurate. It is regarded as a better trade-off between time cost and model accuracy and is thus appreciated in applications requiring large and/or rapid simulations. Unlike other computer models, DT continuously tracks the physical twin by leveraging sensor data, machine learning, and the internet of things [88]. It is compelling to use surrogates in DT technology since detailed models are generally computationally demanding and thereby are not suitable to effectively fulfil the tasks such as optimization, control, and decision-making [89].

Data-driven surrogates are the most prevalent type of surrogate, which has low online evaluation cost while subjecting to the computational-expensive calibration, overfitting, and subjective structure [90]. Tang et al. trained a surrogate based on deep learning to describe the temporal dynamics of an oil-water subsurface flow model [91]. In the context of AM, a recurrent neural network was proposed by Mozaffar et al. to predict temperature profiles with different geome-

tries in the directed energy deposition process, which was expensive in storage and computational cost due to the requirement of an exorbitant amount of training data [92]. In [93], Roy et al. designed heat influence zones to cut down the data required to train deep neural network mapping from part sizes to thermal histories. For a complicated engineering process like AM, the full-order models are normally high-dimensional, the dimensionality of which increases with higher complexity and a larger number of system parameters. Dimension reduction is therefore commonly employed in surrogate models to reduce the computation required without overly sacrificing the representativeness of the whole process [94]. Liu et al. developed a surrogate that couples principal component analysis (PCA) and the Kriging model, which was employed to efficiently analyze uncertainty propagation of dynamic systems in engineering [95]. In [96], the reduced Gaussian process (GP) emulated the high-dimensional parametrized partial differential equations (PDEs) where the dimensionality of output space was dramatically reduced by Isomap and kernel Isomap method and the high-dimensional results were ultimately predicted as linear combinations of trained snapshots. In [97], the surrogate model consisted of three cascading steps: dimension reduction, low-dimensional mapping, and reconstruction. While the encoding and decoding of the autoencoder were separated for dimension reduction and reconstruction, the low-dimensional mapping was a feed-forward neural network trained to connect the encoder and decoder of the autoencoder.

Another type of surrogates is projection-based where the governing equations are projected to a low-dimensional subspace by an orthonormal basis which should be able to span a space covering the final results. Projection bases can be established via moment matching [98], balanced truncation [99], or proper orthogonal decomposition (POD) [100]. The POD method is a data-driven approach to build optimal bases where snapshots generated by the full order model are gathered to form orthonormal projection bases [100]. As described in [101], moment matching is an interpolation approach generating bases based on the Hermite interpolation condition and Krylov subspace technique. Balanced truncation, however, is a common notion in the community of control theory where bases are found based on the reachability and observability Gramian [99]. In [102], Youngsoo et al. de-

veloped the construction of the incremental space-time reduced basis for a large-scale linear Boltzmann transport problem. A further speed-up was achieved in [103] after the FEM for elliptic partial differential equations were projected onto a low-dimensional subspace. The projected model was further sketched with randomized sampling to accomplish a speed-up of two orders of magnitude with a small compromise in accuracy. The performance of projection-based surrogates is limited in nonlinear systems because the number of nonlinear computations is still consistent with the dimensionality of the original, not the reduced system. To tackle this issue, Boris et al. first lifted the nonlinear model and then applied POD bases. The lifting introduced auxiliary variables and transformed ordinary differential equations (ODEs) or differential-algebraic equations into polynomial systems by variable transformation [100]. Since the reduced-order operators of the polynomial system could be precomputed, the online computation was reduced without further approximation of nonlinear systems. In [104], the nonlinear parts of ODEs were approximated by the discrete empirical interpolation method (DEIM). With the combination of projection and interpolation, DEIM managed to cut down the number of nonlinear computations into a significantly less amount proportional to the model dimension reduced by POD.

Surrogates can also be multi-fidelity-based, which are established by simplifying the underlying physics and/or loosening the numerical resolution. To reduce the computational complexity for the thermal simulation of LPBF, the complicated melt pool convection can be ignored as in [13, 105], or be simplified to an anisotropic enhanced thermal conductivity model as in [12, 37]. To reduce the spatial nodes in FEM, adaptive meshing can be employed where a finer mesh is applied in the area with more complicated temperature fluctuation. Using the thermal simulation in [13] as the benchmark, a lower-fidelity model was developed in [14], the computational complexity of which was alleviated by a two-dimensional adaptive mesh where the computation time and memory were dramatically reduced by increasing the distance between lower nodes. In [17], a strategy of adaptive mesh was provided to refine the discretization around geometric components while the rest of the domain had coarser meshes. A multi-fidelity-based surrogate can also be established by combing a low-fidelity simulation with a

coarse mesh and a discrepancy surrogate describing the difference between low- and high-fidelity simulations. Zhang et al. constructed a multi-fidelity surrogate model using linear regression. This method effectively utilized data from multiple fidelity sources, leveraging low-fidelity models as basis functions while identifying their coefficients and discrepancy function through linear regression [106]. Han et al. proposed an alternative cokriging method for variable-fidelity surrogate modelling, in which a scale factor was incorporated into the cokriging predictor to accurately capture the impact of low-fidelity data on high-fidelity function predictions. Through illustrative examples, they demonstrated the potential of the proposed surrogate in facilitating efficient aerodynamic data generation, shape optimization, and other areas employing computer codes with diverse levels of fidelity [107].

Chapter 3

Nonlinear 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. The melt pools fuse with the adjacent printed part, 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 figure 3.1.

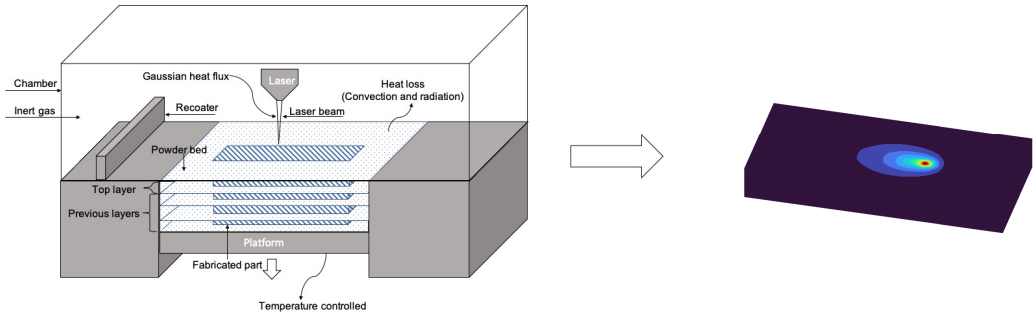


Figure 3.1: Left, a schematic depicting the apparatus of a LPBF printing process and to the right a typical profile of the temperature distribution in the part when heated with a Gaussian heat source.

3.1 Governing equations

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

$$\rho(u)c(u)\frac{\partial u(x,t)}{\partial t} - \nabla \cdot \vec{\kappa}(u)\nabla u(x,t) = 0, \quad (x,t) \quad \text{in } \Omega \times [0, t_N], \tag{3.1}$$

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 [12]. The boundary of Ω denoted as Γ is partitioned into three parts: the top surface Γ_t, the side surface Γ_s that's 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

$$\vec{\kappa}(u)\nabla u(x,t) \cdot \hat{n} = q(x,t) - q_c(u) - q_r(u), \quad x \text{ on } \Gamma_t \cup \Gamma_s, \tag{3.2}$$

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

$$q(x,t) = \frac{2\alpha P}{\pi \varsigma^2} \exp\Big(-\frac{2\|x - \mu(t)\|^2}{\varsigma^2}\Big), \quad x \text{ on } \Gamma_t, \tag{3.3}$$

where $\mu(t) \in \mathbb{R}^3$ is the spatial coordinate of the laser beam centre at time t, α is the absorptivity, P is the laser power, and ς is the effective radius of the laser beam [13]. For the heat losses due to convection and radiation, we consider the nonlinear models

$$q_c(u) = h(u - u_a), \quad x \text{ on } \Gamma_t \cup \Gamma_s, \tag{3.4}$$

$$q_r(u) = \sigma_s \varepsilon (u^4 - u_a^4), \quad x \text{ on } \Gamma_t \cup \Gamma_s, \tag{3.5}$$

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 [15, 16]. 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 [108]

$$u(x,t) = u_b, \quad x \text{ on } \Gamma_b,\ t \in [0, t_N]. \tag{3.6}$$

In effect, an initial condition

$$u(x,0) = u_0, \quad x \text{ in } \Omega, \tag{3.7}$$

suffices to yield a unique temperature solution $u(x,t)$ to simulate the heat transfer over the spatial and temporal domain. In this thesis, the laser beam is modeled perpendicular to the top surface of the powder bed. If a tilted angle is needed, the heat source model in equation (3.3) can be changed as an anisotropic surface heat source such as a double-elliptical shape featuring varied front and rear heat fractions and effective radius [56].

3.2 Temperature-dependent thermal properties

3.2.1 Anisotropy of thermal conductivity

The influence of fluid flow, rather than heat conduction, on melt pool geometry and temperature distribution is more pronounced, necessitating the use of detailed CFD for precise predictions. However, employing detailed CFD models demands extensive computational resources and expertise [37]. As a practical alternative, the anisotropy due to the intricate melt pool convection is simplified as an anisotropic enhanced thermal conductivity model which specifies the divergence term in equation (3.1) as

$$\nabla \cdot \vec{\kappa}(u)\nabla u := \sum_{i,j=1}^{3} \frac{\partial}{\partial x_i}\Big[\kappa_{ij}(u)\Big]\frac{\partial u}{\partial x_j}. \tag{3.8}$$

where κ_{ii} satisfy $\kappa_{11}(u) : \kappa_{22}(u) : \kappa_{33}(u) = \lambda_1 : \lambda_2 : \lambda_3$, for some positive weights $\lambda \in \mathbb{R}^3$. To correct the liquid thermal conductivity in three Cartesian directions, we have κ_{ii} scaled by $\lambda = [\lambda_1, \lambda_2, \lambda_3]^T$ and $\kappa_{ij} = 0$ when $i \neq j$. Furthermore, λ_1,

λ_2, λ_3 are different in liquid phase, while $\lambda_1 = \lambda_2 = \lambda_3 = 1$ for the unmelted materials [12, 37]. The vector λ in liquid phase can be considered as time-invariant to be estimated from experimental data as in [13], or it may allow to vary randomly in a range which resembles more accurately a real process [75, 109].

3.2.2 Latent heat

The characterization of the latent heat effect during the isothermal phase change is incorporated through the specific heat capacity c. Since the phase change between solid and liquid in metals is not purely isothermal, the abrupt transition is smoothed within the temperature range defined by u_s and u_l which respectively represent the lower and upper temperature bounds of the mushy area [12]. To ensure a more realistic representation of the phase change process, the model of specific heat capacity c is

$$c = \bar{c} + L\frac{\partial f_p}{\partial u}, \tag{3.9}$$

where L is the specific latent heat of materials and

$$f_p = 1/(1 + \exp(-\varpi(u - (u_s + u_l)/2))), \tag{3.10}$$

is a temperature-dependent sigmoid function where $\varpi > 0$ is the logistic growth rate controlling smoothness. The calibration of the parameter ϖ is necessary since there is no available measurement to determine the exact process of the phase change.

3.2.3 Polynomial fitting

The temperature-dependent thermal properties including thermal conductivity κ, density ρ, and specific heat capacity $\bar{c}$ are modelled as polynomials by fitting with experimental data. The fitted polynomials are piecewise functions in three temperature ranges: $u \leqslant u_s$, $u_s < u < u_l$, and $u \geqslant u_l$, which respectively indicates the solid phase, the mushy area, and the liquid phase [13].

3.2.4 Thermal properties of powder

In the temperature range $u \leqslant u_s$, the materials have two phases in LPBF: powder and solid. Different from the thermal properties of solid which are fitted polynomials based on experimental data, thermal properties of powder denoted as κ_p, ρ_p, and $\bar{c}_p$ are modelled according to the porosity of powder bed ϕ_p, thermal properties of the solid material and the inert gas atmosphere [70, 110].

$$\kappa_p = (1 - \phi_p)^{1.5}\kappa, \tag{3.11}$$

$$\rho_p = (1 - \phi_p)\rho + \phi_p\rho_a, \tag{3.12}$$

$$\bar{c}_p = (1 - \phi_p)\bar{c} + \phi_p c_a, \tag{3.13}$$

where the density ρ_a and specific heat capacity c_a of the surrounded inert gas are similarly modelled as polynomials fitted by experimental data.

3.3 The full-order numerical solver with FEM

A high-fidelity numerical solver of the nonlinear thermal model as equations (3.1)-(3.7) 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 $n = 0, 1, \cdots, N$, the spatial domain Ω is discretized into n_s surface triangles, n_e tetrahedron elements, and $\tilde{n}_d = n_d + \bar{n}_d$ nodes including n_d degrees of freedom (DoF) and $\bar{n}_d$ nodes on the bottom surface Γ_b. The temperatures are approximated by linear combinations of basis functions ϕ_i for $i = 1, \cdots, \tilde{n}_d$. 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 temperatures $u_n \in \mathbb{R}^{n_d}$ at n_d DoF and the temperatures $\bar{u} = u_b \times \mathbf{1} \in \mathbb{R}^{\bar{n}_d}$ imposed as a constant temperature u_b according to the Dirichlet condition as equation (3.6). Given u_{n-1}, the temperature at the next time step u_n satisfies

$$A(u_n)u_n = b(u_{n-1}, u_n), \quad \text{for} \quad n = 1, \cdots, N, \tag{3.14}$$

where the temperature-dependent 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 equation (3.14) are composed by

$$A(u_n) = K(u_n) + \frac{1}{\Delta t} M(u_n) + Q_r(u_n) + Q_c, \tag{3.15}$$

$$b(u_{n-1}, u_n) = \frac{1}{\Delta t} M(u_n) u_{n-1} + l(t_n) + a - (\bar{K}(u_n) + \bar{Q}_r(u_n) + \bar{Q}_c)\bar{u}. \tag{3.16}$$

As stated in equation (3.15), the matrices on the left-hand side of equation (3.14) are defined on n_d DoF including the matrix of convection heat loss $Q_c \in \mathbb{R}^{n_d \times n_d}$ that is not temperature-dependent and three temperature-dependent matrices: the stiffness matrix $K(u_n) \in \mathbb{R}^{n_d \times n_d}$, the mass matrix $M(u_n) \in \mathbb{R}^{n_d \times n_d}$, and the matrix of radiation heat loss $Q_r(u_n) \in \mathbb{R}^{n_d \times n_d}$. On the right hand side of equation (3.14), there are the terms $\frac{1}{\Delta t} M(u_n) u_{n-1} + l(t_n) + a$ 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 the bottom surface 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 equation (3.15) and (3.16) are defined as the integrals over the domain or the boundary, the integrands of which are specified in table 3.1 where ϕ_i is the basis function of node i in FEM. The Gaussian quadrature approximations of these integrals are also listed in table 3.1. n_k, n_m, n_r, and n_c respectively represent the number of integration points sampled in each element to approximate the integrals in $K(u_n)$, $M(u_n)$, $Q_r(u_n)$, and Q_c by Gaussian quadrature rules. $|\Omega_p|$ is the volume of the pth tetrahedron element for $p = 1, \cdots, n_e$, and $|S_q|$ is the area of the qth triangle surface for $q = 1, \cdots, n_s$. The spatial coordinates and weights for the τth integration points of $K(u_n)$, $M(u_n)$, $Q_r(u_n)$, Q_c, a and $l(t_n)$ are respectively denoted as $\{\mathbf{x}_k^{(\tau)} \in \mathbb{R}^{n_e \times 3}, w_k^{(\tau)} \in \mathbb{R}^{n_e}\}_{\tau=1}^{n_k}$, $\{\mathbf{x}_m^{(\tau)} \in \mathbb{R}^{n_e \times 3}, w_m^{(\tau)} \in \mathbb{R}^{n_e}\}_{\tau=1}^{n_m}$, $\{\mathbf{x}_r^{(\tau)} \in \mathbb{R}^{n_s \times 3}, w_r^{(\tau)} \in \mathbb{R}^{n_s}\}_{\tau=1}^{n_r}$, $\{\mathbf{x}_c^{(\tau)} \in \mathbb{R}^{n_s \times 3}, w_c^{(\tau)} \in \mathbb{R}^{n_s}\}_{\tau=1}^{n_c}$, $\{\mathbf{x}_a \in \mathbb{R}^{n_s \times 3}, w_a \in \mathbb{R}^{n_s}\}$, and $\{\mathbf{x}_l^{(\tau)} \in \mathbb{R}^{n_s \times 3}, w_l^{(\tau)} \in \mathbb{R}^{n_s}\}_{\tau=1}^{n_l}$. As stated in algorithm 1, the nonlinear system of equations (3.14) is solved via Picard's iterations which takes several iterations of solving n_d-dimensional linear equations to converge to the solution of the original nonlinear equation when the error at the $(k+1)$th iteration satisfies

$\|A(u_n^{(k+1)})u_n^{(k+1)} - b(u_{n-1}, u_n^{(k+1)})\| < \epsilon_p$ or $\|u_n^{(k+1)} - u_n^{(k)}\|/\|u_n^{(k+1)}\| < \epsilon_p$ with a small tolerance ϵ_p, e.g. 10^{-5}, or the maximum number of iteration reaches [18, 111]. The steps of implementing the numerical solver with FEM are detailed in algorithm 2.

Algorithm 1 The Picard iteration of solving a nonlinear heat equation

Input: the temperature $u_{n-1} \in \mathbb{R}^{n_d}$, an error tolerance $\epsilon_p \in \mathbb{R}$, a maximum number of iteration n_p, the temperature-dependent matrix $A(u_n) \in \mathbb{R}^{n_d \times n_d}$, and vector $b(u_{n-1}, u_n) \in \mathbb{R}^{n_d}$.

Output: the temperature $u_n \in \mathbb{R}^{n_d}$.

1: $k = 0$.
2: $u_n^{(0)} = u_{n-1}$.
3: **do**
4: $\quad u_n^{(k+1)} = A(u_n^{(k)})^{-1} b(u_{n-1}, u_n^{(k)})$.
5: $\quad e_p = \|A(u_n^{(k+1)})u_n^{(k+1)} - b(u_{n-1}, u_n^{(k+1)})\|$ or $\|u_n^{(k+1)} - u_n^{(k)}\|/\|u_n^{(k+1)}\|$.
6: $\quad k = k + 1$.
7: **while** $e_p > \epsilon_p$ and $k < n_p$
8: $u_n = u_n^{(k)}$.

Table 3.1: The integral definition of matrices in FEM and their Gaussian quadrature approximation.

Symbol	Size	Integral definition ($i, j = 1, \cdots, n_d$, $\bar{j} = 1, \cdots, \bar{n}_d$)	Gaussian quadrature approximation
$K(u_n)$	$\mathbb{R}^{n_d \times n_d}$	$K(u_n)_{ij} = \int_\Omega \vec{\kappa}(u_n)\nabla\phi_i \cdot \nabla\phi_j \mathrm{d}x$	$W_k^T D_k W_k$
$M(u_n)$	$\mathbb{R}^{n_d \times n_d}$	$M(u_n)_{ij} = \int_\Omega \rho(u_n)c(u_n)\phi_i\phi_j \mathrm{d}x$	$\sum_{\tau=1}^{n_m} W_m^{(\tau)^T} D_m^{(\tau)} W_m^{(\tau)}$
$Q_r(u_n)$	$\mathbb{R}^{n_d \times n_d}$	$Q_r(u_n)_{ij} = \int_{\Gamma_t \cup \Gamma_s} \sigma\varepsilon u_n^3 \phi_i\phi_j \mathrm{d}s$	$\sum_{\tau=1}^{n_r} W_r^{(\tau)^T} D_r^{(\tau)} W_r^{(\tau)}$
Q_c	$\mathbb{R}^{n_d \times n_d}$	$Q_{c_{ij}} = \int_{\Gamma_t \cup \Gamma_s} h\phi_i\phi_j \mathrm{d}s$	$\sum_{\tau=1}^{n_c} W_c^{(\tau)^T} D_c^{(\tau)} W_c^{(\tau)}$
a	$\mathbb{R}^{n_d}$	$a_i = \int_{\Gamma_t \cup \Gamma_s} (hu_a + \sigma\varepsilon u_a^4)\phi_i \mathrm{d}s$	$W_a^T d_a$
$l(t_n)$	$\mathbb{R}^{n_d}$	$l(t_n)_i = \int_{\Gamma_t} q(t_n)\phi_i \mathrm{d}s$	$\sum_{\tau=1}^{n_l} W_l^{(\tau)^T} d_l^{(\tau)}$
$\bar{K}(u_n)$	$\mathbb{R}^{n_d \times \bar{n}_d}$	$\bar{K}(u_n)_{i\bar{j}} = \int_\Omega \vec{\kappa}(u_n)\nabla\phi_i \cdot \nabla\phi_{\bar{j}} \mathrm{d}x$	$W_k^T D_k \bar{W}_k$
$\bar{Q}_r(u_n)$	$\mathbb{R}^{n_d \times \bar{n}_d}$	$\bar{Q}_r(u_n)_{i\bar{j}} = \int_{\Gamma_t \cup \Gamma_s} \sigma\varepsilon u_n^3 \phi_i\phi_{\bar{j}} \mathrm{d}s$	$\sum_{\tau=1}^{n_r} W_r^{(\tau)^T} D_r^{(\tau)} \bar{W}_r^{(\tau)}$
$\bar{Q}_c$	$\mathbb{R}^{n_d \times \bar{n}_d}$	$\bar{Q}_{c_{i\bar{j}}} = \int_{\Gamma_t \cup \Gamma_s} h\phi_i\phi_{\bar{j}} \mathrm{d}s$	$\sum_{\tau=1}^{n_c} W_c^{(\tau)^T} D_c^{(\tau)} \bar{W}_c^{(\tau)}$

As the definition of D_k, $D_m^{(\tau)}$, and $D_r^{(\tau)}$ stated in table 3.2, the number of temperature-dependent nonlinear computations relates to the number of tetrahedron elements n_e and the number of triangle surfaces n_s, both of which are

Table 3.2: The definition of matrices in Gaussian quadrature approximation.

Symbol	Size	Definition ($j = 1, \cdots, n_d$, $\bar{j} = 1, \cdots, \bar{n}_d$, $p = 1, \cdots, n_e$, $q = 1, \cdots, n_s$)
D_k	$\mathbb{R}^{3n_e \times 3n_e}$	$[D_{k_{3p-2,3p-2}}, D_{k_{3p-1,3p-1}}, D_{k_{3p,3p}}]^T = (\lvert\Omega_p\rvert \sum_{\tau=1}^{n_k} w_k^{(\tau)} \kappa(\mathbf{x}_{k_{p*}}^{(\tau)}))\lambda$
$D_m^{(\tau)}$	$\mathbb{R}^{n_e \times n_e}$	$D_{m_{pp}}^{(\tau)} = \lvert\Omega_p\rvert\, w_m^{(\tau)} \rho(\mathbf{x}_{m_{p*}}^{(\tau)}) c(\mathbf{x}_{m_{p*}}^{(\tau)})$
$D_r^{(\tau)}$	$\mathbb{R}^{n_s \times n_s}$	$D_{r_{qq}}^{(\tau)} = \lvert S_q\rvert\, \sigma_s \varepsilon w_r^{(\tau)} u(\mathbf{x}_{r_{q*}}^{(\tau)})^3$
$D_c^{(\tau)}$	$\mathbb{R}^{n_s \times n_s}$	$D_{c_{qq}}^{(\tau)} = \lvert S_q\rvert\, w_c^{(\tau)} h$
d_a	$\mathbb{R}^{n_s}$	$d_{a_q} = \lvert S_q\rvert\, w_a (h u_a + \sigma_s \varepsilon u_a^4)$
$d_l^{(\tau)}$	$\mathbb{R}^{n_s}$	$d_{l_q}^{(\tau)} = \lvert S_q\rvert\, w_l^{(\tau)} q(\mathbf{x}_{l_{q*}}^{(\tau)})$
W_k	$\mathbb{R}^{3n_e \times n_d}$	$[W_{k_{3p-2,j}}, W_{k_{3p-1,j}}, W_{k_{3p,j}}]^T = \nabla \phi_j$
$\bar{W}_k$	$\mathbb{R}^{3n_e \times \bar{n}_d}$	$[W_{k_{3p-2,\bar{j}}}, W_{k_{3p-1,\bar{j}}}, W_{k_{3p,\bar{j}}}]^T = \nabla \phi_{\bar{j}}$
$W_l^{(\tau)}$	$\mathbb{R}^{n_s \times n_d}$	$W_{l_{*j}}^{(\tau)} = \phi_j(\mathbf{x}_l^{(\tau)})$
$W_m^{(\tau)}$	$\mathbb{R}^{n_e \times n_d}$	$W_{m_{*j}}^{(\tau)} = \phi_j(\mathbf{x}_m^{(\tau)})$
$W_r^{(\tau)}$	$\mathbb{R}^{n_s \times n_d}$	$W_{r_{*j}}^{(\tau)} = \phi_j(\mathbf{x}_r^{(\tau)})$
$\bar{W}_r^{(\tau)}$	$\mathbb{R}^{n_s \times \bar{n}_d}$	$\bar{W}_{r_{*\bar{j}}}^{(\tau)} = \phi_{\bar{j}}(\mathbf{x}_r^{(\tau)})$
$W_c^{(\tau)}$	$\mathbb{R}^{n_s \times n_d}$	$W_{c_{*j}}^{(\tau)} = \phi_j(\mathbf{x}_c^{(\tau)})$
$\bar{W}_c^{(\tau)}$	$\mathbb{R}^{n_s \times \bar{n}_d}$	$\bar{W}_{c_{*\bar{j}}}^{(\tau)} = \phi_{\bar{j}}(\mathbf{x}_c^{(\tau)})$
W_a	$\mathbb{R}^{n_s \times n_d}$	$W_{a_{*j}} = \phi_j(\mathbf{x}_a)$

large numbers due to fine spatial discretization. Also, higher nonlinearity results in larger n_k, n_m, and n_r, which yields more nonlinear computations. These nonlinear computations are computed repeatedly in each Picard iteration as in algorithm 1. Therefore, it is time-consuming to solve a high-fidelity nonlinear heat equation due to the high dimensionality and nonlinearity. The high dimensionality caused by fine spatial discretization, however, can be alleviated by adaptive meshing. In the context of LPBF, the powder is successively and vertically piled. We correspondingly develop the vertically extended mesh with fine discretization around the scanning line while having coarse mesh elsewhere. The nonlinear thermal model with adaptive mesh is still computationally expensive, which hinders its application requiring rapid and/or massive thermal simulations like uncertainty quantification, process optimization, and closed-loop control.

3.4 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, a large n_d increases the dimension of the (3.14) 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 to form the temperature-dependent matrices. Here we take the mass matrix $M(u_n) \in \mathbb{R}^{n_d \times n_d}$ as an example to show the time consumption due to high dimensionality and nonlinearity. The entries of M are defined as

$$M_{ij}(u_n) = \int_\Omega \rho(u_n) c(u_n) \phi_i(x) \phi_j(x) \mathrm{d}x \quad \text{for} \quad i, j = 1, \cdots, n_d, \tag{3.17}$$

where the integrand is the product of the density function $\rho(u_n)$, the specific heat capacity function $c(u_n)$, and the local basis functions $\phi(x)$ of the i-th and j-th DoF. The integral in equation (3.17) is approximated via a n_m-point Gaussian quadrature rule as

$$\begin{aligned} M_{ij}(u_n) &= \sum_{e=1}^{n_e} \int_{\Omega_e} \rho(u_n) c(u_n)\, \phi_i(x) \phi_j(x)\, \mathrm{d}x, \quad i, j \in \operatorname{supp} \Omega_e \\ &\approx \sum_{e=1}^{n_e} \sum_{\tau=1}^{n_m} |\Omega_e|\, w_\tau \rho(u_n(\mathbf{x}_{m_{e*}}^{(\tau)})) c(u_n(\mathbf{x}_{m_{e*}}^{(\tau)})) \phi_i(\mathbf{x}_{m_{e*}}^{(\tau)}) \phi_j(\mathbf{x}_{m_{e*}}^{(\tau)}), \end{aligned} \tag{3.18}$$

where $\mathbf{x}_{m_{e*}}^{(\tau)}$ is the coordinate of the τ-th integration point within the e-th element. In matrix form, we can rewrite the approximation of M as

$$\begin{aligned} M(u_n) &\approx \sum_{\tau=1}^{n_m} {W_m^{(\tau)}}^T D_m^{(\tau)} W_m^{(\tau)}, \\ &= {\mathsf{W_m}}^T \mathcal{D}_m \mathsf{W_m}, \end{aligned} \tag{3.19}$$

where $\mathsf{W_m} \in \mathbb{R}^{n_m n_e \times n_d}$ are the evaluations of the basis functions at all integration points and the diagonal matrix $\mathcal{D}_m \in \mathbb{R}^{n_m n_e \times n_m n_e}$ contains the temperature-dependent entries. Specifically,

$$\mathsf{W_m} = \begin{bmatrix} W_m^{(1)} \\ \vdots \\ W_m^{(n_m)} \end{bmatrix}, \quad \text{and} \quad \mathcal{D}_m = \begin{bmatrix} D_m^{(1)} & & \\ & \ddots & \\ & & D_m^{(n_m)} \end{bmatrix}. \tag{3.20}$$

The matrix $\mathsf{W_m}$ does not depend on temperature, and can thus be computed before the start of the simulation. The online computation due to nonlinearity is reflected by the diagonal of $\mathcal{D}_m$ 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 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 equation (3.14) but maintains the accuracy of temperature profiles.

Algorithm 2 The numerical solver with FEM

Input: an total amount of layers n_L; an ambient temperature u_a; a building paltform temperature u_b; temperature-dependent functions of the material's thermal properties: $\vec{\kappa}(u)$, $\rho(u)$ and $c(u)$; a heat source function $q(x,t)$; an error tolerance ϵ_p; a time interval Δt; a total amount of simulation time steps n_t; a maximum number of Picard iteration n_p; an anisotropy scaling factor λ; Ω is the entire domain with the bottom Γ_b, side Γ_s, and top surface Γ_t.

Output: temperatures $U \in \mathbb{R}^{n_d \times n_t}$.

1: **for** $L \leftarrow 1$ to n_L **do**

2: Linear part of Neumann condition $Q_c \in \mathbb{R}^{n_d \times n_d}$, $\bar{Q}_c \in \mathbb{R}^{n_d \times \bar{n}_d}$ and $a \in \mathbb{R}^{n_d}$:

$$Q_{c_{ij}} \leftarrow \int_{\Gamma_t \cup \Gamma_s} h\phi_i\phi_j \mathrm{d}s \quad \text{for} \quad i,j = 1, \cdots, n_d,$$

$$a_i \leftarrow \int_{\Gamma_t \cup \Gamma_s} (hu_a + \sigma\varepsilon u_a^4)\phi_i \mathrm{d}s \quad \text{for} \quad i = 1, \cdots, n_d,$$

$$\bar{Q}_{c_{i\bar{j}}} \leftarrow \int_{\Gamma_t \cup \Gamma_s} h\phi_i\phi_{\bar{j}} \mathrm{d}s \quad \text{for} \quad i = 1 \cdots n_d, \quad \bar{j} = 1, \cdots, \bar{n}_d.$$

Initial condition $u_0 \leftarrow u_a \times \mathbf{1} \in \mathbb{R}^{n_d}$.

Set up the flag of material phase $\xi \leftarrow \mathbf{0} \in \mathbb{R}^{n_e}$ where 0 and 1 respectively represent powder and the printed object.

if $L \neq 1$ **then**

Respectively replace u_0 and ξ at previous layers with the temperature and material phase at the n_tth time step in the $L-1$ layers domain.

end if

for $n \leftarrow 1$ to n_t **do**

Load vector $l \in \mathbb{R}^{n_d}$: $l_i \leftarrow \int_{\Gamma_t} q(n_t \Delta t)\phi_i \mathrm{d}s$ for $i = 1, \cdots, n_d$.

$k = 0$.

$u_n^{(k)} \leftarrow u_{n-1}$.

do

Get thermal properties with $u_n^{(k)}$: thermal conductivity $\vec{\kappa}(u_n^{(k)})$, density $\rho(u_n^{(k)})$, and specific heat capacity $c(u_n^{(k)})$.

Stiffness matrix $K(u_n^{(k)}) \in \mathbb{R}^{n_d \times n_d}$ and $\bar{K}(u_n^{(k)}) \in \mathbb{R}^{n_d \times \bar{n}_d}$:

$$K(u_n^{(k)})_{ij} \leftarrow \int_\Omega \vec{\kappa}(u_n^{(k)}) \nabla\phi_i \cdot \nabla\phi_j \mathrm{d}x \quad \text{for} \quad i, j = 1, \cdots, n_d,$$

$$\bar{K}(u_n^{(k)})_{i\bar{j}} \leftarrow \int_\Omega \vec{\kappa}(u_n^{(k)}) \nabla\phi_i \cdot \nabla\phi_{\bar{j}} \mathrm{d}x \quad \text{for} \quad i = 1, \cdots, n_d, \quad \bar{j} = 1, \cdots, \bar{n}_d.$$

Mass matrix $M(u_n^{(k)}) \in \mathbb{R}^{n_d \times n_d}$:

$$M(u_n^{(k)})_{ij} \leftarrow \int_\Omega \rho(u_n^{(k)}) c(u_n^{(k)}) \phi_i \phi_j \mathrm{d}x \quad \text{for} \quad i, j = 1, \cdots, n_d.$$

Radiation heat loss matrix $Q_r(u_n^{(k)}) \in \mathbb{R}^{n_d \times n_d}$ and $\bar{Q}_r(u_n^{(k)}) \in \mathbb{R}^{n_d \times \bar{n}_d}$:

$$Q_r(u_n^{(k)})_{ij} \leftarrow \int_{\Gamma_s \cup \Gamma_t} \sigma\varepsilon(u_n^{(k)})^3 \phi_i \phi_j \mathrm{d}s \quad \text{for} \quad i, j = 1, \cdots, n_d,$$

$$\bar{Q}_r(u_n^{(k)})_{i\bar{j}} \leftarrow \int_{\Gamma_s \cup \Gamma_t} \sigma\varepsilon(u_n^{(k)})^3 \phi_i \phi_{\bar{j}} \mathrm{d}s \quad \text{for} \quad i = 1, \cdots, n_d, \quad \bar{j} = 1, \cdots, \bar{n}_d.$$

$A(u_n^{(k)}) \leftarrow K(u_n^{(k)}) + \frac{1}{\Delta t} M(u_n^{(k)}) + Q_r(u_n^{(k)}) + Q_c$.

$b(u_{n-1}, u_n^{(k)}) \leftarrow \frac{1}{\Delta t} M(u_n^{(k)}) u_{n-1} + l(t_n) + a - (\bar{K}(u_n^{(k)}) + \bar{Q}_r(u_n^{(k)}) + \bar{Q}_c)\bar{u}$.

$u_n^{(k+1)} \leftarrow A(u_n^{(k)})^{-1} b(u_{n-1}, u_n^{(k)})$.

$e_p \leftarrow \|u_n^{(k+1)} - u_n^{(k)}\| / \|u_n^{(k+1)}\|$ or $\left\| A(u_n^{(k+1)}) u_n^{(k+1)} - b(u_{n-1}, u_n^{(k+1)}) \right\|$.

$k \leftarrow k + 1$.

while $e_p > \epsilon_p$ and $k < n_p$

Compute the temperature at the element center $u_c \in \mathbb{R}^{n_e}$ based on $u_n^{(k)}$.

Update the phase vector $\xi \leftarrow (u_c > u_s) \vee \xi$.

end for

end for

Chapter 4

Reduced Gaussian process surrogate

A data-driven surrogate is designed to obtain results almost in real time. It is trained to form the latent mapping from the printing attributes in LPBF (laser power and scan speed) and time to the high-dimensional temperature profiles. Directly employing machine learning algorithms like neural networks requires a huge amount of training data to guarantee fine response surfaces yielding accurate predictions, which is a major obstacle in the data-driven surrogate of thermal modelling in LPBF since for our high-dimensional nonlinear numerical solver with FEM it might already be computationally prohibitive to generate a large number of temperature snapshots. We thus propose a reduced GP surrogate leveraging nonlinear model order reduction (MOR) to alleviate the cost of data generation and training.

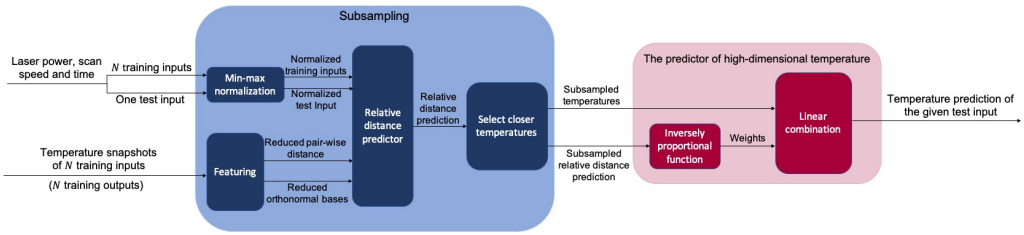


Figure 4.1: The framework of the reduced GP surrogate predicting high-dimensional temperatures with three control inputs (laser power, scan speed and time) via linear combinations of subsampled training temperatures.

The reduced GP surrogate is composed of two parts: subsampling and a

swift predictor of high-dimensional temperatures. In subsampling, several training temperatures are selected according to the relative distance prediction which is generated by a predictor formed by Gaussian process regression (GPR) and nonlinear MOR. The swift predictor of temperatures then estimates the high-dimensional temperature by the linear combination of the subsampled temperatures weighted according to the relative distance predictions. The framework of the reduced GP surrogate is as figure 4.1 where the modules of subsampling and the predictor of high-dimensional temperature are respectively explained in section 4.1 and 4.2.

4.1 Subsampling

The high-dimensional temperature of a given test input can be predicted by a linear combination of some training temperatures deemed as the closest ones to this prediction, in which the weights of the linear combination are inversely proportional to relative distances between the temperature prediction and the selected training temperatures. We thus develop a subsampling scheme to select training temperature snapshots based on a relative distance predictor mapping input parameters to relative distances.

4.1.1 Normalization and Featuring

As a data-driven method, its prediction accuracy relies on finding a representative training data set which is generated by the heat simulator with FEM. The training inputs are selected from the controllable parameters of a LPBF process including laser power, beam size, scan speed, time, preheating temperature, and so on. In this thesis, the inputs matrix $X \in \mathbb{R}^{n_x \times N}$ contains N training points for a triplet ($n_x = 3$) of laser power P, scan speed v, and time t. For each $X_{*i} \doteq [P_i, v_i, v_i t_i]^T$, one can readily run the heat simulator with FEM as equation (3.14) by tuning the appropriate boundary conditions for times $t = t_1, \ldots, t_i$ until we obtain $u(X_{*i})$. Repeating the process for $N \gg 3$ sampling points produces a corresponding temperature snapshots matrix $U \in \mathbb{R}^{n_d \times N}$ where $U_{*i} = u(X_{*i})$ and N is typically

chosen to be smaller than n_d. To make the learned parameters in GPR scale-invariant, we normalise the input quantities in the rows of X to be in the $[0, 1]$ range

$$\tilde{X}_{i*} = \frac{X_{i*} - \min(X_{i*})}{\max(X_{i*}) - \min(X_{i*})}, \quad \text{for} \quad i = 1, \cdots, n_x. \tag{4.1}$$

As for the feature engineering of the training output, we approximate the pair-wise distances of temperatures in U with nonlinear dimensionality reduction [112]. The relative distance between U_{*i} and U_{*j} for $i, j = 1, \cdots, N$, denoted as Q_{ij} in the pair-wise distance matrix $Q \in \mathbb{R}^{N \times N}$, is the shortest distance between the ith and jth node of a weighted g-neighborhood graph. We first construct a weighted g-neighborhood graph with N vertices representing the N training temperature profiles and with edges weighted as the adjacency matrix $G \in \mathbb{R}^{N \times N}$ which is given by

$$G_{ij} = \begin{cases} \|U_{*i} - U_{*j}\| & \text{if } U_{*j} \text{ is a neighbour of } U_{*i} \\ \infty & \text{otherwise} \end{cases}, \quad \text{for} \quad i, j = 1, \cdots, N, \tag{4.2}$$

where U_{*j} is a neighbour of U_{*i} if their Euclidean distance belongs to the g smallest values in $\|U_{*i} - U_{*j}\|$ for $j = 1, \cdots, N$. The pair-wise distance matrix Q is computed as the shortest path matrix of this graph via the Floyd–Warshall algorithm [113]. We then find a reduced representation of Q by a rank $\bar{n}_r$ truncation of its singular value decomposition (SVD) which is

$$Q = E \Lambda V^T \approx E_r Z, \tag{4.3}$$

where $Z = \Lambda_r V_r^T$ is the $\bar{n}_r \times N$ reduced representation after featuring. Here, $E \in \mathbb{R}^{N \times N}$, $V \in \mathbb{R}^{N \times N}$, and $\Lambda \in \mathbb{R}^{N \times N}$ are respectively the left, right singular vectors, and the diagonal matrix of singular values. $E_r = [E_{*1}, \cdots, E_{*\bar{n}_r}] \in \mathbb{R}^{N \times \bar{n}_r}$, $V_r = [V_{*1}, \cdots, V_{*\bar{n}_r}] \in \mathbb{R}^{N \times \bar{n}_r}$, and the diagonal matrix $\Lambda_r \in \mathbb{R}^{\bar{n}_r \times \bar{n}_r}$ respectively contain the left, right singular vectors, and singular values corresponding to the $\bar{n}_r$ largest singular values in Λ. To guarantee the accuracy of reduction, the value of $\bar{n}_r$ is chosen to satisfy $(\sum_{i=1}^{\bar{n}_r} \Lambda_{ii})/(\sum_{i=1}^{N} \Lambda_{ii}) > 99\%$. After normalisation and featuring, the resulting training data set is $\{\tilde{X}_{*i}, Z_{*i}\}_{i=1}^{N}$ which represents a map-

ping from n_x-dimensional normalised inputs to $\bar{n}_r$-dimensional reduced outputs with $\bar{n}_r \ll N$.

4.1.2 Relative distance predictor

As the preprocessed training inputs $\tilde{X}$ and outputs Z are passed to the GPR module, $\bar{n}_r$ GPs are respectively trained to fit $\{\tilde{X}, Z_{i*}\}_{i=1}^{\bar{n}_r}$. GP is promising in providing accurate predictions with corresponding variance using less lengthy training compared with some other machine learning algorithms like neural networks. The multiple inputs and single output regression between $\tilde{X} \in \mathbb{R}^{n_x \times N}$ and each row of $Y \in \mathbb{R}^{\bar{n}_r \times N}$ defined as

$$Y_{i*} = Z_{i*} - \frac{1}{N} Z_{i*} \mathbf{1}\mathbf{1}^T, \quad \text{for} \quad i = 1, \cdots, \bar{n}_r, \tag{4.4}$$

is a discrete GP with zero mean and positive definite covariance matrix $\Sigma^{(i)} \in \mathbb{R}^{N \times N}$, which is

$$Y_{i*} \approx \mathcal{GP}(0, \Sigma^{(i)}), \tag{4.5}$$

where the covariance matrix $\Sigma^{(i)}(\Theta_{*i})$ arises from the discretisation of a covariance function (kernel) $k(x, x'; \theta^{(i)})$ with hyper-parameters $\theta^{(i)}$. More specifically

$$\Sigma^{(i)} = \bar{\Sigma}^{(i)} + \varepsilon_i^2 I, \tag{4.6}$$

where

$$\bar{\Sigma}_{pq}^{(i)} \approx k(\tilde{X}_{*p}, \tilde{X}_{*q}; \theta^{(i)}) \doteq \theta_0^{(i)} \exp\Big(-\frac{1}{2} \sum_{j=1}^{n_x} \frac{(\tilde{X}_{jp} - \tilde{X}_{jq})^2}{\theta_j^{(i)}}\Big), \quad \text{for} \quad p, q = 1, \cdots, N, \tag{4.7}$$

outlining the smoothness of GP as a squared exponential function of $\tilde{X}$ [96]. The strictly positive hyper-parameters $\Theta_{*i} \in \mathbb{R}^{n_x+2}$ defined as

$$\Theta_{*i} = [\theta_0^{(i)}, \theta_1^{(i)}, \cdots, \theta_\ell^{(i)}, \varepsilon_i]^T, \tag{4.8}$$

is optimally found via training. We hereafter respectively denote the ith column of the strictly positive matrix $\hat{\Theta} \in \mathbb{R}^{(n_x+2) \times \bar{n}_r}$ and the covariance matrix $\hat{\Sigma}^{(i)}$ as

the trained hyper-parameters and the trained covariance matrix of the ith GP for $i = 1, \cdots, \bar{n}_r$. Assume Θ_{*i} is uniformly distributed, then with Baye's rule we have $f(\Theta_{*i}|Y_{i*}) \propto f(Y_{i*}|\Theta_{*i})$ where the likelihood function $f(Y_{i*}|\Theta_{*i})$ as defined in equation (4.5) is

$$f(Y_{i*}|\Theta_{*i}) = (2\pi)^{-\frac{N}{2}} |\Sigma^{(i)}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} Y_{i*} (\Sigma^{(i)})^{-1} Y_{i*}^T\right). \tag{4.9}$$

Therefore, the maximum likelihood estimation of Θ_{*i} is

$$\begin{aligned} \hat{\Theta}_{*i} &= \underset{\Theta_{*i}}{\operatorname{argmax}}\, f(\Theta_{*i}|Y_{i*}), \\ &= \underset{\Theta_{*i}}{\operatorname{argmax}}\, f(Y_{i*}|\Theta_{*i}) = \underset{\Theta_{*i}}{\operatorname{argmax}} \ln f(Y_{i*}|\Theta_{*i}), \\ &= \underset{\Theta_{*i}}{\operatorname{argmax}} \left(-\frac{1}{2} \ln |\hat{\Sigma}^{(i)}| - \frac{1}{2} Y_{i*} (\hat{\Sigma}^{(i)})^{-1} Y_{i*}^T - \frac{N}{2} \ln (2\pi)\right). \end{aligned} \tag{4.10}$$

For a given test input $x \in \mathbb{R}^{n_x}$ that is not in the training data set, we first normalized it as $\tilde{x}$. Then, the prediction of its reduced representation $\hat{z} \in \mathbb{R}^{\bar{n}_r}$ and the corresponding variances $\mathrm{Var}(\hat{z})$ are

$$\hat{z}_i = B_{*i}^T S_{*i} + \frac{1}{N} Z_{i*} \mathbf{1}, \quad \text{for} \quad i = 1, \cdots, \bar{n}_r, \tag{4.11}$$

$$\mathrm{Var}(\hat{z}_i) = \hat{\Theta}_{1i} - B_{*i}^T (\hat{\Sigma}^{(i)})^{-1} B_{*i}, \quad \text{for} \quad i = 1, \cdots, \bar{n}_r, \tag{4.12}$$

where the matrix $B \in \mathbb{R}^{N \times \bar{n}_r}$ and the matrix $S \in \mathbb{R}^{N \times \bar{n}_r}$ are respectively

$$B_{*i} = [k(\tilde{X}_{*1}, \tilde{x}; \hat{\theta}^{(i)}), \cdots, k(\tilde{X}_{*N}, \tilde{x}; \hat{\theta}^{(i)})]^T, \quad \text{for} \quad i = 1, \cdots, \bar{n}_r. \tag{4.13}$$

$$S_{*i} = (\hat{\Sigma}^{(i)})^{-1} Y_{i*}^T, \quad \text{for} \quad i = 1, \cdots, \bar{n}_r. \tag{4.14}$$

It is worth noting that only the matrix B is necessary to be computed online as it depends on the normalized test input $\tilde{x}$, while the more computationally expensive matrix S can be pre-computed offline. Therefore, the prediction of $\hat{z}$ as equation (4.11) can be implemented instantaneously. We then reconstruct the

full relative distance $\hat{m} \in \mathbb{R}^N$ as

$$\hat{m} = E_r \hat{z}, \tag{4.15}$$

where E_r is the $\bar{n}_r$-dimensional orthonormal bases in equation (4.3), and $\hat{m}_i$ is the relative distance prediction between the temperature $u(x)$ and the ith training temperature U_{*i}. In other words, the smaller $\hat{m}_i$ is the closer the training temperature U_{*i} might be. We thereby could select a subset of training temperatures that are deemed to be more similar to the final temperature prediction $u(x)$.

4.2 High-dimensional temperature prediction

According to the prediction of relative distances $\hat{m}$, we select $n_r^{'}$ closest training data in U as $\tilde{U} \in \mathbb{R}^{n_d \times n_r^{'}}$ where $n_r^{'}$ is normally selected as a comparably small value (like 10 or 20). To extrapolate the high-dimensional temperature we compute the weights $w \in \mathbb{R}^{n_r^{'}}$ of $\tilde{U}$ as

$$w_i \propto e^{-\varrho \hat{m}_j}, \quad \text{such that} \quad \sum_{i=1}^{n_r^{'}} w_i = 1, \tag{4.16}$$

where the ith column $\tilde{U}_{*i}$ weighted by w_i is selected from the jth column in the original matrix of training temperatures U, and ϱ is a constant adjusting the decline of weights. Eventually, the temperature prediction of the test input x is

$$\hat{u}(x) = \tilde{U} w. \tag{4.17}$$

The pseudo-code of the reduced GP surrogate is as algorithm 3.

Algorithm 3 The reduced GP surrogate

Input: the training input $X \in \mathbb{R}^{n_x \times N}$, the training output $U \in \mathbb{R}^{n_d \times N}$, a test input $x \in \mathbb{R}^{n_x}$, a neighborhood size g, the number of selected temperature $n_r^{'}$.

Output: the high-dimensional temperature prediction $\hat{u}(x) \in \mathbb{R}^{n_d}$.

1: Normalize X as $\tilde{X}$, and normalize x as $\tilde{x}$.

2: Compute the adjacency matrix $G \in \mathbb{R}^{N \times N}$ of the weighted g-neighborhood graph with the training output U.

3: Compute the pair-wise shortest distance $Q \in \mathbb{R}^{N \times N}$ via Floyd-Warshall algorithm.

4: Get $E_r \in \mathbb{R}^{N \times \bar{n}_r}$ and $Z \in \mathbb{R}^{\bar{n}_r \times N}$ by the rank $\bar{n}_r$ truncation of the SVD of Q where $\bar{n}_r$ is chosen to cover more than 99% of the largest singular values.

5: Train $\bar{n}_r$ GP to respectively fit $\{X, Z_{i*}\}_{i=1}^{\bar{n}_r}$.

6: Predict $\hat{z} \in \mathbb{R}^{\bar{n}_r}$ by the $\bar{n}_r$ trained GP with the normalized test input $\tilde{x}$.

7: Reconstruct the full relative distance prediction $\hat{m} \in \mathbb{R}^N$ as $\hat{m} = E_r \hat{z}$.

8: Select the $n_r^{'}$ closest temperatures $\tilde{U} \in \mathbb{R}^{n_d \times n_r^{'}}$ based on $\hat{m}$.

9: Compute the weights $w_i \propto e^{-\varrho \hat{m}_j}$ such that $\sum_{i=1}^{n_r^{'}} w_i = 1$ where $\tilde{U}_{*i}$ is selected from U_{*j}.

10: Get the prediction of the high-dimensional temperature $\hat{u}(x) = \tilde{U} w$

Chapter 5

Sketched surrogate with data-driven local projection

While the reduced GP surrogate is entirely data-driven, the sketched surrogate with data-driven local projection preserves the structure of the original heat simulator with FEM as equation (3.14) but expedites it by data-driven local projection and randomized sketching. Data-driven local projection targets at addressing the issue of high dimensionality with the help of subsampling described in section 4.1. Randomized sketching, on the other hand, bypasses the majority of burdensome computation caused by high nonlinearity. The framework of the sketched surrogate with data-driven local projection is as figure 5.1 where the modules of local projection, randomized sketching and the high-dimensional temperature predictor are respectively explained in section 5.1, 5.2, and 5.3.

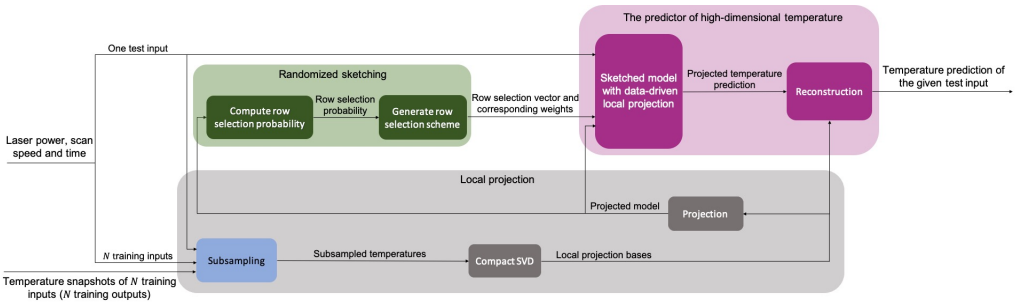


Figure 5.1: The framework of the sketched surrogate with data-driven local projection where high-dimensional temperatures of given test inputs (laser power, scan speed, and time) are reconstructed by the projected results produced from the surrogate with subsampling-based local projection and randomized sketching.

5.1 Subsampling-based local projection

Recall the numerical solver with FEM as in section 3.3 where the temperature at $t_n = n\Delta t$ time instance is denoted as $u_n \in \mathbb{R}^{n_d}$ on the discrete spatial domain with n_d DoF. Instead of directly computing equation (3.14), we can assume

$$u_n \approx \Psi_n r_n, \tag{5.1}$$

for a low-dimensional vector $r_n \in \mathbb{R}^{n_r}$ and a basis $\Psi_n \in \mathbb{R}^{n_d \times n_r}$. The FEM equation is then projected to the subspace spanned by the columns of Ψ_n to yield

$$\Psi_n^T A(u_n)\Psi_n r_n = \Psi_n^T b(u_{n-1}, u_n), \tag{5.2}$$

where the projection basis $\Psi_n \in \mathbb{R}^{n_d \times n_r}$ is orthonormal with $n_r \ll n_d$. The basis Ψ_n should span a space containing the final temperature result u_n with as less dimension n_r as possible. In the reduced GP surrogate, we select a subset of training temperatures and weight them according to the relative distance prediction. It indicates that the selected subset forms a basis of the final temperature prediction. Therefore, we develop a similar subsampling scheme to generate local projection bases. Given a test input $x_n = [P, v, vt_n]^T$, we predict its relative distance $\hat{m}$ following equation (4.11) - (4.15). Then, temperature snapshots in the training temperature matrix U corresponding to the n_r smallest values in $\hat{m}$ are selected to form a subset $\mathcal{U} \in \mathbb{R}^{n_d \times n_r}$. The local projection basis Ψ_n is the left singular vectors in the compact SVD of $\mathcal{U}$. We hereafter use boldface to represent matrices and vectors after local projection such as

$$\mathbf{A}(u_n) := \Psi_n^T A(u_n)\Psi_n \in \mathbb{R}^{n_r \times n_r}, \tag{5.3}$$

$$\mathbf{b}(u_{n-1}, u_n) := \Psi_n^T b(u_{n-1}, u_n) \in \mathbb{R}^{n_r}. \tag{5.4}$$

The discretised heat equation after local projection is rewritten as

$$\mathbf{A}(u_n) r_n = \mathbf{b}(u_{n-1}, u_n). \tag{5.5}$$

One advantage of local projection is that it relieves the curse of dimensionality as the number of inputs increases and/or each input has a larger range of interest. Both will cause a larger training data set (larger N) yielding more resource cost in training and storage, but once we get the relative distance predictor the dimension after projection n_r remains small so the online execution time does not increase much. However, the construction of local bases needs to be implemented online since in subsampling we use the values of test inputs which are not known offline, while the construction of global basis with POD method [100] can be finished offline since it constructs the global basis as the left singular vectors of the compact SVD of all temperature snapshots available thus independent from test inputs. In other words, local projection sacrifices the online time cost required to generate local project bases to further reduce the projected dimension from N to n_r with $n_r \ll N$.

Nonetheless, the projected model remains time-consuming due to its high nonlinearity. After local projection, the computation of nonlinear temperature-dependent functions in both $\mathbf{A}$ and $\mathbf{b}$ are not reduced. The number of nonlinear computations still depends on the original fine spatial discretization, the order of nonlinearity, and Gaussian quadrature rules. The difficulty of nonlinearity and its solution with randomized sketching are further elaborated in the following section.

5.2 Randomized sketching

The nonlinearity of the heat equation comes from temperature-dependent thermal properties and radiation heat loss, all of which are modeled as nonlinear functions. As stated in section 3.3, the temperature-dependent parts of $A(u_n)$ and $b(u_{n-1}, u_n)$, such as the stiffness matrix $K(u_n) \in \mathbb{R}^{n_d \times n_d}$, the mass matrix $M(u_n) \in \mathbb{R}^{n_d \times n_d}$, and the nonlinear part of radiation heat loss $Q_r(u_n) \in \mathbb{R}^{n_d \times n_d}$, need to integrate the corresponding nonlinear functions over the domain discretized into n_d DoF and n_e tetrahedral elements. For example, the mass matrix

$M(u_n)$ is defined as

$$M(u_n)_{ij} = \int_\Omega \rho(u_n)c(u_n)\phi_i\phi_j \mathrm{d}x, \tag{5.6}$$

where ϕ_i and ϕ_j are respectively the basis function of the ith and jth node in FEM. The integration in equation (5.6) is approximated by taking n_m integration points in each element based on Gaussian quadrature rules [114, 115], which is

$$M(u_n) \approx \sum_{\tau=1}^{n_m} W_m^{(\tau)^T} D_m^{(\tau)} W_m^{(\tau)} \tag{5.7}$$

where $W_m^{(\tau)} \in \mathbb{R}^{n_e \times n_d}$ is the matrix of basis functions in FEM evaluated at the τth integration point of each element, and $D_m^{(\tau)} \in \mathbb{R}^{n_e \times n_e}$ is a diagonal matrix containing the product $\rho(u_n)c(u_n)$ evaluated by the temperature at the τth integration point of each element. After local projection, the projected mass matrix $\mathbf{M}(u_n) \in \mathbb{R}^{n_r \times n_r}$ becomes

$$\begin{aligned} \mathbf{M}(u_n) &= \sum_{\tau=1}^{n_m} \Psi_n^T W_m^{(\tau)^T} D_m^{(\tau)} W_m^{(\tau)} \Psi_n \\ &= \sum_{\tau=1}^{n_m} \mathbf{W}_m^{(\tau)^T} D_m^{(\tau)} \mathbf{W}_m^{(\tau)}, \end{aligned} \tag{5.8}$$

where $\mathbf{W}_m^{(\tau)} = W_m^{(\tau)}\Psi_n \in \mathbb{R}^{n_e \times n_r}$ for $\tau = 1, \cdots, n_m$ do not need to be repeatedly computed in Picard iterations since they are not temperature-dependent. The diagonals of $D_m^{(\tau)}$ for $\tau = 1, \cdots, n_m$, however, have a total of $n_e \times n_m$ values awaiting for nonlinear computation of density and specific heat capacity at integration points in each Picard iteration. It is time-consuming as finer spatial discretization will increase n_e and higher order of nonlinearity will increase n_m. Accordingly, the expensive time cost due to nonlinearity is not reduced by a local projection which is also true for other temperature-dependent matrices like $\mathbf{K}(u_n)$ and $\mathbf{Q_r}(u_n)$. It is a major obstacle in implementing the projected model rapidly. To address this issue, we introduce randomised sketching based on Bernoulli sampling. Take the projected mass matrix $\mathbf{M}(u_n)$ as an example, the sum in equation (5.8) can be rewritten as the sum of $n_m n_e$ rank-one matrices

which is

$$\begin{aligned}\mathbf{M}(u_n) &= \mathcal{W}_m^T \mathcal{D}_m \mathcal{W}_m \\ &= \sum_{i=1}^{n_m n_e} \mathcal{D}_{m_{ii}} \mathcal{W}_{m_{i*}}^T \mathcal{W}_{m_{i*}},\end{aligned} \tag{5.9}$$

where the thin matrix $\mathcal{W}_m \in \mathbb{R}^{n_m n_e \times n_r}$ and the diagonal matrix $\mathcal{D}_m \in \mathbb{R}^{n_m n_e \times n_m n_e}$ are respectively

$$\mathcal{W}_m = \begin{bmatrix} \mathbf{W}_m^{(1)} \\ \vdots \\ \mathbf{W}_m^{(n_m)} \end{bmatrix}, \quad \text{and} \quad \mathcal{D}_m = \begin{bmatrix} D_m^{(1)} & & \\ & \ddots & \\ & & D_m^{(n_m)} \end{bmatrix}. \tag{5.10}$$

We then sketch $\mathbf{M}(u_n)$ with only some of these rank-one matrices weighted by a factor. Namely,

$$\hat{\mathbf{M}}(u_n) = \sum_{i=1}^{n_m n_e} \zeta_{m_i} \frac{\mathcal{D}_{m_{ii}}}{\omega_{m_i}} \mathcal{W}_{m_{i*}}^T \mathcal{W}_{m_{i*}}, \tag{5.11}$$

where ζ_{m_i} is a Bernoulli random variable with probability of success $0 \leqslant \omega_{m_i} \leqslant 1$. We only manage to dramatically reduce evaluating $\mathcal{D}_{m_{ii}}$ without much accuracy compromise when many $\zeta_{m_i} = 0$ and $\|\hat{\mathbf{M}}(u_n) - \mathbf{M}(u_n)\|$ is small with a high probability. To fulfill this, we set the probability ω_{m_i} according to the thin matrix $\mathcal{W}_m$ as algorithm 4 [103]. The row selection scheme is accordingly generated as the row selection vector ζ_{m_i} and the corresponding weight $1/\omega_{m_i}$ for $i = 1, \cdots, n_m n_e$.

Algorithm 4 The row selection scheme in randomised sketching

Input: a thin matrix $\mathcal{W}_m \in \mathbb{R}^{n_m n_e \times n_r}$, a constant ι.

Output: a row selection vector $\zeta_m \in \mathbb{R}^{n_m n_e}$, a probability vector $\omega_m \in \mathbb{R}^{n_m n_e}$.

1: $\Phi \in \mathbb{R}^{n_m n_e \times n_r}$ is the left singular vectors in the compact SVD of $\mathcal{W}_m$.
2: **for** $i = 1$ to $n_m n_e$ **do**
3: Compute the leverage score $\ell_i = \|\Phi_{i*}\|^2$.
4: $\omega_{m_i} = \min\{1, \iota / n_r \ell_i\}$.
5: ζ_{m_i} is generated as a Bernoulli random number with probability ω_{m_i}.
6: **end for**

5.3 High-dimensional temperature prediction

Similar to the randomised sketching of the projected mass matrix $\mathbf{M}(u_n)$, we can sketch other temperature-dependent matrices in the locally projected model. The sketched emulator with data-driven local projection then becomes

$$\hat{\mathbf{A}}(\hat{u}_n)\hat{r}_n = \hat{\mathbf{b}}(\hat{u}_{n-1}, \hat{u}_n). \tag{5.12}$$

After solving it with Picard iteration, we obtain an accurate result of $\hat{r}_n$. The prediction of high-dimensional temperature $\hat{u}_n$ is then reconstructed by the data-driven local projection basis Ψ_n following

$$\hat{u}_n = \Psi_n \hat{r}_n. \tag{5.13}$$

The pseudo code of the sketched surrogate with data-driven local projection is as algorithm 5.

Algorithm 5 The sketched surrogate with data-driven local projection

Input: the training input $X \in \mathbb{R}^{n_x \times N}$, the training output $U \in \mathbb{R}^{n_d \times N}$, a test input $x_n = [P, v, vt_n]^T \in \mathbb{R}^{n_x}$, a reduced dimension n_r, the basis function matrices in the full-order model with FEM.

Output: the high-dimensional temperature prediction $\hat{u}(x_n) \in \mathbb{R}^{n_d}$.

1: Subsample the n_r temperatures $\mathcal{U} \in \mathbb{R}^{n_d \times n_r}$ deemed to be the closest ones to $\hat{u}(x_n)$ as in section 4.1.

2: Compute the orthonormal basis $\Psi_n \in \mathbb{R}^{n_d \times n_r}$ as the left singular vectors in the compact SVD of $\mathcal{U}$.

3: Respectively project the basis function matrices as $\mathbf{W}_k$, $\bar{\mathbf{W}}_k$, $\mathbf{W}_a$, $\mathbf{W}_m^{(\tau)}$ for $\tau = 1, \cdots, n_m$, $\mathbf{W}_r^{(\tau)}$ and $\bar{\mathbf{W}}_r^{(\tau)}$ for $\tau = 1, \cdots, n_r$, $\mathbf{W}_c^{(\tau)}$ and $\bar{\mathbf{W}}_c^{(\tau)}$ for $\tau = 1, \cdots, n_c$, and $\mathbf{W}_l^{(\tau)}$ for $\tau = 1, \cdots, n_l$.

4: Respectively form the tall matrices $\mathcal{W}_m$, $\mathcal{W}_r$, $\bar{\mathcal{W}}_r$, $\mathcal{W}_c$, $\bar{\mathcal{W}}_c$, and $\mathcal{W}_l$.

5: In randomized sketching, respectively generate the row selection scheme $\{\zeta_m, \omega_m\}$ from $\mathcal{W}_m$, $\{\zeta_k, \omega_k\}$ from $\mathbf{W}_k$, and $\{\zeta_r, \omega_r\}$ from $\mathcal{W}_r$.

6: Compute the temperature-dependent nonlinear functions for the selected diagonal entries in D_k, $\mathcal{D}_m$, and $\mathcal{D}_r$.

7: Form the sketched and projected matrix $\hat{\mathbf{A}}(\hat{u}_n)$ and vector $\hat{\mathbf{b}}(\hat{u}_{n-1}, \hat{u}_n)$.

8: Solve the sketched and projected model $\hat{\mathbf{A}}(\hat{u}_n)\hat{r}_n = \hat{\mathbf{b}}(\hat{u}_{n-1}, \hat{u}_n)$ with Picard iterations, and get $\hat{r}_n \in \mathbb{R}^{n_r}$.

9: Reconstruct the high-dimensional temperature prediction $\hat{u}(x_n) = \Psi_n \hat{r}_n$.

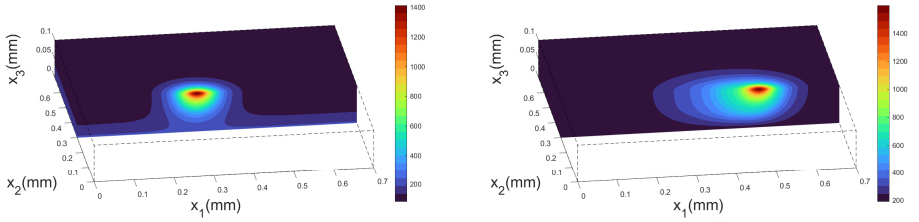
Chapter 6

Sketched surrogate with online local projection

Similar to the sketched surrogate with data-driven local projection aforementioned in chapter 5, the proposed surrogate in this chapter also uses randomized sketching and local projection. The difference, however, is the way to generate the online local projection bases and the approximation of sampling probability in randomized sketching. The meaning of online here is twofold. First, the generation of projection bases is swift and almost in real-time. Second, the entire process of basis generation including the adjustment and calibration is finished in the process of simulation without any offline preparation in advance, namely, the data generation and training required for the data-driven local projection in chapter 5. Compared with the data-driven local projection, this online local projection needs more dimension (normally a few hundred) and therefore takes more time in projection and randomized sketching. To further avoid the computationally expensive full projection and randomized sketching, we develop an additional design to approximate the sampling probability used in randomized sketching.

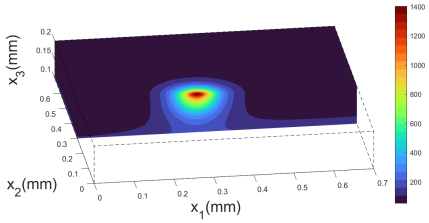
At the start of a new line of scanning in LPBF, the temperature profiles are not regular for the first few time steps. This is partly due to the temperature-controlled building platform and the existing temperature from previous printing. More importantly, with a small time interval, there is not enough heat at the be-

ginning to form steady melt pools. For example, a thermal simulation of scanning a straight line is shown in figure 6.1, from which we can tell that the steady temperature distributions are gradually formed with obvious anisotropy. Therefore,

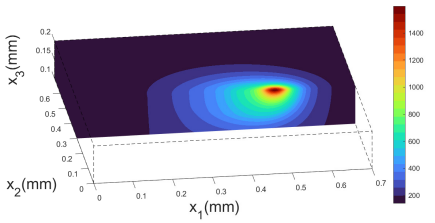


(a) Temperature profile at 0.05ms on layer 1.

(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.

Figure 6.1: 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) can not be described as (anisotropic) Gaussian functions.

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 time-efficient surrogate. The surrogate is also proposed to reduce the expensive time cost due to high dimensionality and nonlinearity. To begin with, we also 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 $u_n \approx \Psi_n r_n$. As mentioned in chapter 5, the equation (3.14) is projected as

$$\mathbf{A}(u_n) r_n = \mathbf{b}(u_{n-1}, u_n), \tag{6.1}$$

where $\mathbf{A}(u_n) = \Psi_n^T A(u_n) \Psi_n$ and $\mathbf{b}(u_{n-1}, u_n) = \Psi_n^T b(u_{n-1}, u_n)$. Same as the surrogate in chapter 5, 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 equation (6.1) with randomized sketching

to yield a swift surrogate as

$$\hat{\mathbf{A}}(\hat{u}_n)\hat{r}_n = \hat{\mathbf{b}}(\hat{u}_n, \hat{u}_{n-1}), \tag{6.2}$$

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 framework of the sketched surrogate with online local projection is as figure 6.2. The procedure of

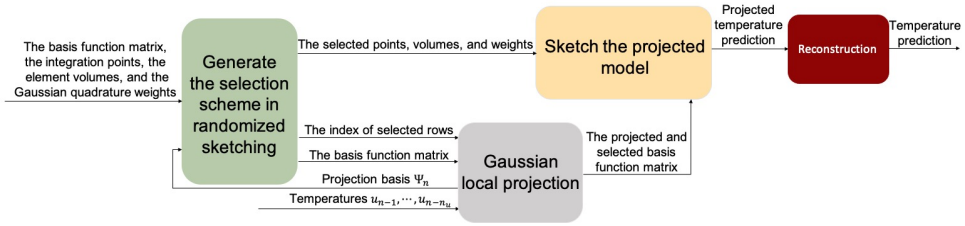


Figure 6.2: The framework of the sketched surrogate with online local projection where Gaussian local projection bases are generated online and randomized sketching is achieved by approximate optimal sampling probability.

implementing the surrogate is outlined in algorithm 8 where both sketching and projection procedures have several parameters to control the trade-off between model accuracy and time cost. From algorithm 8, 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 6.1. Then, we explain online basis generation and projection in section 6.2.

6.1 Randomized sketching with approximate sampling probability

We now explain the design by focusing on the mass matrix $M(u_n)$ defined in equation (3.17) 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 figure 6.3 where the modules related to randomized sketching are illustrated in detail and the module of Gaussian local projection is enriched later in section 6.2. In the reduced model with a projection basis Ψ_n, the projected

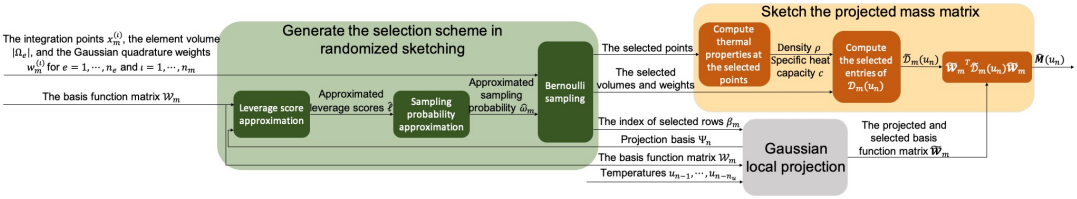


Figure 6.3: The sketching procedure to compute the projected and sketched mass matrix.

mass matrix $\mathbf{M}(u_n) \in \mathbb{R}^{n_r \times n_r}$ is

$$\begin{aligned} \mathbf{M}(u_n) &= \mathcal{W}_m^T \mathcal{D}_m \mathcal{W}_m, \\ &= \sum_{i=1}^{n_m n_e} \mathcal{D}_{m_{ii}} \mathcal{W}_{m_{i*}}^T \mathcal{W}_{m_{i*}}, \end{aligned} \tag{6.3}$$

where the tall matrix $\mathcal{W}_m \in \mathbb{R}^{n_m n_e \times n_r}$ and $\mathcal{D}_m \in \mathbb{R}^{n_m n_e \times n_m n_e}$ are respectively the projected basis function matrix and the temperature-dependent diagonal matrix defined as equation (5.10). We also tell that $\mathbf{M}(u_n)$ can be represented as the linear combination of $n_m n_e$ rank-one matrices $\mathcal{W}_{m_{i*}}^T \mathcal{W}_{m_{i*}}$ weighted by $\mathcal{D}_{m_{ii}}$ for $i = 1, \cdots, n_m n_e$. The time cost due to nonlinearity, however, is reflected by the diagonal matrix $\mathcal{D}_m$. It is shown in equation (6.3) 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 $\mathcal{D}_m$ 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 $\mathcal{W}_m$ and $\mathcal{D}_m$ 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

$$\begin{aligned} \hat{\mathbf{M}}(u_n) &= \tilde{\mathcal{W}}_m^T \tilde{\mathcal{D}}_m \tilde{\mathcal{W}}_m, \\ &= \sum_{i=1}^{n_g} \tilde{\mathcal{D}}_{m_{ii}} \tilde{\mathcal{W}}_{m_{i*}}^T \tilde{\mathcal{W}}_{m_{i*}}, \end{aligned} \tag{6.4}$$

where $\tilde{\mathcal{W}}_m \in \mathbb{R}^{n_g \times n_r}$ only contains the n_g selected rows of $\mathcal{W}_m$. 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 $\tilde{\mathcal{W}}_m$ weighted by $\tilde{\mathcal{D}}_{m_{ii}}$ is selected from the jth row of $\mathcal{W}_m$ weighted by $\mathcal{D}_{m_{jj}}$, we have the diagonal entries of $\tilde{\mathcal{D}}_m \in \mathbb{R}^{n_g \times n_g}$ as

$$\tilde{\mathcal{D}}_{m_{ii}} = \frac{1}{\omega_{m_j}} \mathcal{D}_{m_{jj}}, \quad \text{for} \quad i = 1, \cdots, n_g, \tag{6.5}$$

where $0 < \omega_{m_j} \leqslant 1$ is the probability of selecting the jth row of $\mathcal{W}_m$. All n_g indexes j in equation (6.5) are recorded as a row index vector $\beta_m \in \mathbb{R}^{n_g}$ which is useful in the online projection explained in section 6.2.2. In randomized sketching stated above, it is essential to set all success probabilities $\omega_m \in \mathbb{R}^{n_m n_e}$ properly to ensure low sketching error while cutting down sketching time cost. Accordingly, we specify ω_m based on the fast-computed approximation of leverage scores which is detailed in the subsequent section 6.1.1.

6.1.1 Approximate sampling probability

To significantly reduce nonlinear computations, it is critical to properly set the success probabilities $\omega_m \in \mathbb{R}^{n_m n_e}$ so that we manage to retain $n_g \ll n_m n_e$ and a small sketching error $\|\mathbf{M}(u_n) - \hat{\mathbf{M}}(u_n)\|$. The vector of probabilities ω_m depends on the leverage scores of the rows of $\mathcal{W}_m \in \mathbb{R}^{n_m n_e \times n_r}$ [103, 116]. Namely, for $i = 1, \cdots, n_m n_e$ we have

$$\omega_{m_i} = \min\left(1, \iota / n_r \ell_i\right), \tag{6.6}$$

$$\ell_i = \|\Phi_{i*}\|^2, \tag{6.7}$$

where the statistical leverage score $\ell \in \mathbb{R}^{n_m n_e}$ is the squared row norms of Φ, and $\Phi \in \mathbb{R}^{n_m n_e \times n_r}$ is the left singular vectors in the compact SVD of $\mathcal{W}_m$. The parameter $\iota > 0$ 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 $\mathcal{W}_m = \mathsf{W}_\mathsf{m} \Psi_n$ with the computational complexity $O(n_m n_e n_d n_r)$ and the compact SVD of $\mathcal{W}_m$. Therefore, we propose a way to approximate ℓ with less time in SVD and without implementing the

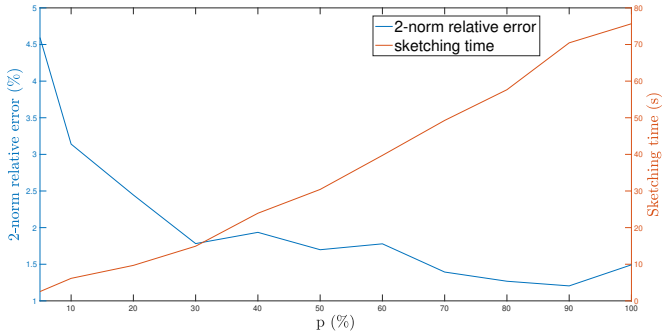


Figure 6.4: The relative errors with different percentages p.

exact product $\mathsf{W_m}\Psi_n$. As in equation (6.7), the exact leverage scores are computed as the squared row norms of the n_r left singular vectors in Φ. 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 $\mathcal{W}_m$, and then use the squared row norms of the $\tilde{n}_r$ columns to approximate ℓ. In specific,

$$\hat{\ell}_i = \|\tilde{\Phi}_{i*}\|^2, \tag{6.8}$$

where $\tilde{\Phi} \in \mathbb{R}^{n_m n_e \times \tilde{n}_r}$ containing the $\tilde{n}_r$ left singular vectors is obtained from the $\tilde{n}_r$ rank approximation of $\mathcal{W}_m$. We can choose $\tilde{n}_r$ as a small proportion of n_r without a high sacrifice in accuracy. As shown in figure 6.4, 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 figure 6.4, 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, $\tilde{\Phi}$ can be fast-computed via randomized SVD. While the standard procedure of randomized SVD in [117] is detailed by algorithm 7 in section 6.2.1, it is modified as algorithm 6 to avoid computing $\mathcal{W}_m = \mathsf{W_m}\Psi_n$ explicitly.

Algorithm 6 The approximation of success probability.

Input: the basis function matrix $\mathsf{W_m} \in \mathbb{R}^{n_m n_e \times n_d}$, the Gaussian projection basis $\Psi_n \in \mathbb{R}^{n_d \times n_r}$, a parameter $\iota > 0$, a proportion $1/n_r \leqslant p < 1$, an embedding dimension k.

Output: an approximation of success probability $\hat{\omega}_m$.

1: $\tilde{n}_r = \lfloor pn_r \rfloor$.
2: Generate two random sub-Gaussian matrices $\Upsilon_1 \in \mathbb{R}^{n_r \times \tilde{n}_r}$ and $\Upsilon_2 \in \mathbb{R}^{k \times n_m n_e}$.
3: Compute $\tilde{\mathcal{W}}_m \in \mathbb{R}^{n_m n_e \times \tilde{n}_r}$ by $\tilde{\mathcal{W}}_m = \mathsf{W_m}(\Psi_n \Upsilon_1)$.
4: Compute the orthonormal matrix $\Lambda \in \mathbb{R}^{n_m n_e \times \tilde{n}_r}$ in the QR-decomposition of $\tilde{\mathcal{W}}_m$.
5: Compute the left singular vectors of $(\Upsilon_2 \Lambda)^\dagger \Upsilon_2 \mathsf{W_m} \Psi_n$ as $\bar{\Phi} \in \mathbb{R}^{\tilde{n}_r \times \tilde{n}_r}$.
6: $\tilde{\Phi} = \Lambda \bar{\Phi}$.
7: **for** $i = 1$ to $n_m n_e$ **do**
8: $\quad \hat{\ell}_i = \|\tilde{\Phi}_{i*}\|^2$.
9: $\quad \hat{\omega}_{m_i} = \min(1, \iota / n_r \hat{\ell}_i)$.
10: **end for**

In specific, as stated in the third step of algorithm 6 we compute $\mathsf{W_m}(\Psi_n \Upsilon_1)$ instead of $\mathsf{W_m} \Psi_n \Upsilon_1$ to avoid the expensive multiplication $\mathsf{W_m} \Psi_n$. The computational complexity of $\mathsf{W_m}(\Psi_n \Upsilon_1)$ and $\mathsf{W_m} \Psi_n \Upsilon_1$ are respectively $O(n_d n_r \tilde{n}_r + n_m n_e n_d \tilde{n}_r)$ and $O(n_m n_e n_d n_r + n_m n_e n_r \tilde{n}_r)$. Since we have $\tilde{n}_r = \lfloor pn_r \rfloor$ and $1/n_r \leqslant p < 1$, it is validated that $O(n_d n_r \tilde{n}_r + n_m n_e n_d \tilde{n}_r) < O(n_m n_e n_d n_r + n_m n_e n_r \tilde{n}_r)$. 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 $\hat{\omega}_m$ is accordingly generated via equation (6.6).

6.2 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 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 centre $\mu(t_n)$ 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 equation (3.3) 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 figure 6.5 in general, while the details of online projection basis generation and online projection are respectively explained in section 6.2.1 and section 6.2.2.

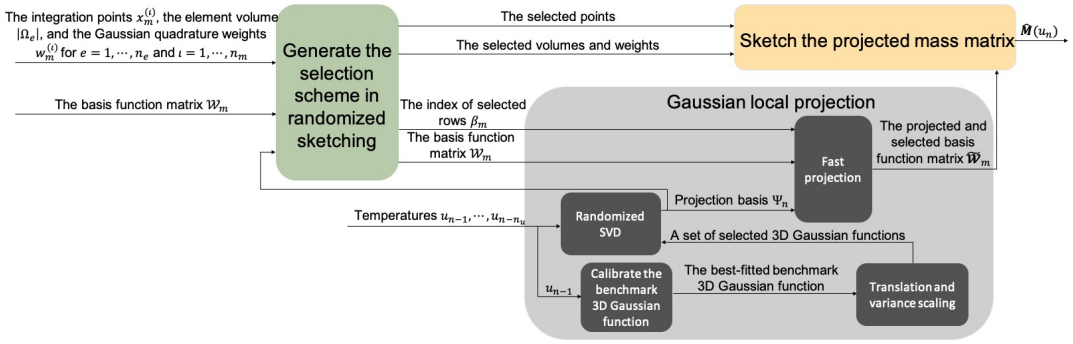


Figure 6.5: The projection procedure to compute the projected and sketched mass matrix.

6.2.1 Online generation of projection basis

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 $\mathcal{G} \in \mathbb{R}^{n_d \times n_G}$ 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 $\mathcal{Q} \in \mathbb{R}^{n_d \times (n_G + n_u + 1)}$ which is

$$\mathcal{Q} = [\mathcal{G}, u_{n-n_u}, \cdots, u_{n-1}, 1], \tag{6.9}$$

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 centre 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 $\mu^n \in \mathbb{R}^{3\times n_\mu}$ where $\mu^n_{*i} = \mu(t_n) + i[\Delta d, 0, 0]^T$ for $i = 0, \cdots, n_\mu - 1$. 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 $f_{n-1}(x) = \exp(-\sum_{i=1}^{3}(x_i - \mu(t_{n-1})_i)^2/2\bar{\sigma}_i^2)$. The optimal $\bar{\sigma}$ satisfies

$$\underset{\bar{\sigma}}{\operatorname{argmin}} \|u_{n-1}(\bar{X})/\max(u_{n-1}) - f_{n-1}(\bar{X})\|, \tag{6.10}$$

where $u_{n-1}(\bar{X}) \in \mathbb{R}^{n_d^{'}}$ 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 $n_d^{'}$ nodes are specified as the rows in the matrix $\bar{X} \in \mathbb{R}^{n_d^{'}\times 3}$. With linear least square regression, $\hat{\bar{\sigma}} \in \mathbb{R}^3$ is approximated by

$$1/\hat{\bar{\sigma}}^2 = (B(\bar{X})^T B(\bar{X}))^{-1} B(\bar{X})^T \ln(u_{n-1}(\bar{X})/\max(u_{n-1})), \tag{6.11}$$

where $B(\bar{X}) \in \mathbb{R}^{n_d^{'}\times 3}$ 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

$$\sigma(\eta)_{*i} = [\exp(\eta_j \ln \bar{\sigma}_1), \exp(\eta_k \ln \bar{\sigma}_2), \exp(\eta_q \ln \bar{\sigma}_3)]^T, \quad \text{for} \quad i = 1, \cdots, n_\sigma, \tag{6.12}$$

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 $\mathcal{G}$ in equation (6.9) as

$$\mathcal{G}_{di} = \exp\left(-\sum_{\iota=1}^{3} \frac{(X_{d\iota} - \mu_{\iota j}^{n})^2}{2\sigma(\eta)_{\iota k}^{\ 2}}\right), \quad \text{for} \quad d = 1, \cdots, n_d, \quad \text{and} \quad i = 1, \cdots, n_G, \tag{6.13}$$

where $X \in \mathbb{R}^{n_d \times 3}$ contains the spatial coordinate of the n_d DoF. The ith column of $\mathcal{G}$ 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 $\mathcal{Q}$. 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 $\mathcal{Q}$ can be computationally expensive. As a result, we use randomized SVD as algorithm 7 to reduce the generation time of the orthonormal basis Ψ_n [117].

Algorithm 7 Randomized SVD with sub-Gaussian random matrices

Input: a matrix $\mathcal{Q} \in \mathbb{R}^{n_d \times n_Q}$, an approximation rank n_r, an embedding dimension k .

Output: the orthonormal basis $\Psi_n \in \mathbb{R}^{n_d \times n_r}$.

1: Generate two random sub-Gaussian matrices $\Upsilon_1 \in \mathbb{R}^{n_Q \times n_r}$ and $\Upsilon_2 \in \mathbb{R}^{k \times n_d}$.

2: $\Lambda_1 = \mathcal{Q} \times \Upsilon_1$.

3: Compute the orthogonal matrix Λ_2 in the QR-decomposition of Λ_1.

4: Compute the left singular vectors of $(\Upsilon_2 \Lambda_2)^\dagger \Upsilon_2 \mathcal{Q}$ as $\bar{\Psi}$.

5: $\Psi_n = \Lambda_2 \bar{\Psi}$.

6.2.2 Online projection

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

$$\mathcal{W}_m = \mathsf{W}_\mathsf{m} \Psi_n, \tag{6.14}$$

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 6.1.1 and algorithm 6. Consequently, we can obtain the row indexes of n_g

successful selections as β_m before fulfilling the projection. We can thereby skip $\mathcal{W}_m$ and directly get $\tilde{\mathcal{W}}_m$ as

$$\tilde{\mathcal{W}}_m = \mathsf{W}_{\mathsf{m}_{\beta_m *}} \Psi_n. \tag{6.15}$$

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 equation (6.14) to $O(n_g n_d n_r)$ for equation (6.15).

Algorithm 8 The sketched surrogate with online local projection.

Input: a total amount of time steps n_t, the full-order model as equation (3.14), the number of reduced dimension n_r, the integers n_t, n_u, n_μ, and n_σ, a distance Δd, a scale vector η.

Output: the temperature profiles u_n for $n = 1, \cdots, N$.

1: **for** $n = 1$ to N **do**

2: **if** $n \leqslant n_t$ **then**

3: Get u_n with the full-order model $A(u_n)u_n = b(u_{n-1}, u_n)$.

4: **else**

5: Get n_μ mean coordinates as $\mu^n_{*i} = \mu(t_n) + i[\Delta d, 0, 0]^T$ for $i = 0, \cdots, n_\mu - 1$ where $\mu(t_n)$ is the center of laser beam at t_n.

6: Calibrate the benchmark standard deviation $\bar{\sigma}$ according to the previous temperature u_{n-1} via linear regression.

7: Scale the benchmark standard deviation $\bar{\sigma}$ with η to form n_σ different sets of standard deviation.

8: Evaluate $n_G = n_\mu n_\sigma$ different Gaussian functions as $\mathcal{G} \in \mathbb{R}^{n_d \times n_G}$.

9: Gather the Gaussian functions and previous temperatures as $\mathcal{Q} = [\mathcal{G}, u_{n-n_u}, \cdots, u_{n-1}, 1]$.

10: Generate the Gaussian local projection basis Ψ_n as the left sigular vectors of $\mathcal{Q}$.

11: Respectively approximate success probabilities $\hat{\omega}_m$, $\hat{\omega}_k$, and $\hat{\omega}_r$ for W_m, W_k, and W_r following algorithm 6 where the proportion p is used to balance the trade-off between sketching time and model accuracy.

12: Respectively generate the selected row indexes β_m, β_k, and β_r accord-

ing to $\hat{\omega}_m$, $\hat{\omega}_k$, and $\hat{\omega}_r$.

Implement online projection to obtain $\tilde{\mathcal{W}}_m$, $\tilde{\mathcal{W}}_k$, and $\tilde{\mathcal{W}}_r$.

Get the approximated matrices $\hat{\mathbf{M}}(\hat{u}_n)$, $\hat{\mathbf{K}}(\hat{u}_n)$, and $\hat{\mathbf{Q}}_{\mathbf{r}}(\hat{u}_n)$, and accordingly form $\hat{\mathbf{A}}(\hat{u}_n)$ and $\hat{\mathbf{b}}(\hat{u}_n, \hat{u}_{n-1})$.

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

Chapter 7

Results

The thermal model of LPBF is numerically solved with FEM, which manages to describe thermal characteristics with different printing parameters. It serves as a reference to evaluate the performance of the three surrogates in model accuracy and time cost reduction. We hereafter denote the reduced GP surrogate as $\mathcal{F}_1$, the sketched surrogate with data-driven local projection as $\mathcal{F}_2$, and the sketched surrogate with online local projection as $\mathcal{F}_3$. The results of the full-order thermal simulator with FEM, the performance of the three surrogates, and their comparison are detailed in this chapter.

7.1 The thermal simulator with FEM

7.1.1 Parameter setup and validation

The full-order model with FEM is used as a reference model, which simulates the process of scanning two straight lines back and forth in a two-layer AlSi10Mg powder bed surrounded by an argon atmosphere. AlSi10Mg is a favoured choice for applications requiring lightweight metallic parts with high castability and relatively high mechanical properties [118]. The parameters of this simulator include the mesh size, thermal properties of AlSi10Mg and argon, adjustable parameters in numerical experiments, and other model parameters. The powder bed domain numerically expressed as a cuboid is discretized with a mesh scheme with a conforming refinement in the printing area, in which fine meshes distribute

around the scanning line on the top layer (tetrahedrons with a side length of 0.005mm) and the rest area has coarser mesh (tetrahedrons with a side length of 0.025mm). 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 DoF, 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 being set to be around and moving with the melt pool. The thermal properties of AlSi10Mg and argon are fitted by piecewise polynomials based on experimental data in [13, 119–122]. In specific, the fitted polynomials

Table 7.1: The fitted polynomial functions of thermal properties [13, 119–122].

Definition (Unit)	Fitted polynomials
Thermal conductivity of AlSi10Mg (W/mK)	$\kappa = \begin{cases} 8.36 \times 10^{-8}u^3 - 1.67 \times 10^{-4}u^2 + 0.12u + 144.76, & u \leqslant u_s \\ -1.91u + 1257.74, & u_s < u < u_l \\ 0.0088u + 78.46, & u \geqslant u_l \end{cases}$
Density of AlSi10Mg (kg/m^3)	$\rho = \begin{cases} -8.63 \times 10^{-4}u^2 + 0.17u + 2634.20, & u \leqslant u_s \\ -0.85u + 2922.55, & u_s < u < u_l \\ -0.33u + 2613.19, & u \geqslant u_l \end{cases}$
Specific heat capacity of AlSi10Mg (J/kgK)	$\bar{c} = \begin{cases} 0.0011u^2 + 0.0168u + 741.16, & u \leqslant u_s \\ 0.32u + 924.04, & u_s < u < u_l \\ 0.0068u + 1134.32, & u \geqslant u_l \end{cases}$
Density of argon (kg/m^3)	$\rho_a = 3.54 \times 10^{-13}u^4 - 1.96 \times 10^{-9}u^3 + 3.93 \times 10^{-6}u^2 - 0.0036u + 1.65$
Specific heat capacity of argon (J/kgK)	$c_a = 520$

are listed in table 7.1 where $u_s = 567°C$ and $u_l = 614°C$ are respectively the lower and upper bound of the mushy temperature range. AlSi10Mg is solid when its temperature is lower than u_s, and is liquid when its temperature exceeds u_l. Apart from these temperature-dependent polynomials, the parameters of porosity

and latent heat that do not depend on temperature are specified in table 7.2. Laser power P and scan speed v are the printing parameters adjustable in the

Table 7.2: Parameters in thermal properties.

Symbol	Definition	Value	Unit
ϕ_p	the powder bed porosity	0.4	-
L	the latent heat of AlSi10Mg	423	kJ/kg
ϖ	the logistic growth rate of f_p	0.05	-

simulations. For the numerical experiments in this section, the admissible range of laser power P and scan speed v are respectively 200-300W and 200-1000mm/s. The static scale factor of anisotropy $\lambda = [2.5, 1, 3.2]^T$ is found by the trial-and-error calibration to fit the experiment results in [13] which includes a selective laser melting process using the apparatus with a YLR-500-SM ytterbium fibre 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 in [13] is visualized as figure 7.1 where the positions at the centre of each layer are used to validate our thermal simulation. We further detail the validation of our thermal simulation in table 7.3.

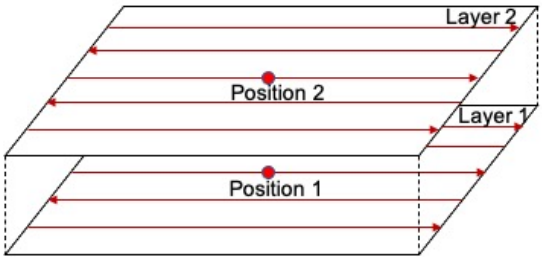


Figure 7.1: The laser scanning pattern in [13] where position 1 and 2 used in validation are respectively the centre of the first and second layer.

The scale factor λ, however, can also be a random scale factor. In this section, 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. Other model parameters are set as the same for all numerical experiments. Specifically, on the top surface

Table 7.3: The validation of the printing process with laser power 250W and scan speed 200mm/s.

Layer	Type	Highest temperature (°C)	Melt pool size (μm)
1	[13]	1482	$129.1 \times 94.2 \times 61.7$
	FEM	1501	$137.5 \times 105.6 \times 65.6$
		1501	$133.7 \times 105.3 \times 66.1$
		1500	$133.5 \times 102.2 \times 65.7$
		1498	$133.8 \times 105.0 \times 65.1$
		1495	$135.0 \times 102.7 \times 66.0$
		1498	$135.0 \times 104.7 \times 65.8$
		1498	$133.6 \times 103.4 \times 65.3$
		1497	$133.1 \times 104.2 \times 66.0$
		1495	$133.4 \times 105.1 \times 65.7$
		1496	$133.2 \times 104.4 \times 65.9$
		1464	$120.0 \times 96.8 \times 59.7$
		1460	$117.5 \times 95.1 \times 61.7$
		1467	$118.1 \times 96.2 \times 60.4$
		1460	$118.1 \times 96.8 \times 60.7$
		1462	$117.5 \times 95.1 \times 60.5$
		1463	$118.0 \times 96.9 \times 60.7$
		1459	$117.5 \times 95.2 \times 60.9$
		1467	$118.8 \times 95.1 \times 60.1$
		1463	$118.1 \times 95.5 \times 61.5$
		1490	$130.9 \times 98.3 \times 63.4$
2	[13]	1548	$148.3 \times 111.4 \times 67.5$
	FEM	1549	$158.1 \times 116.0 \times 81.2$
		1553	$163.0 \times 120.1 \times 84.1$
		1554	$162.8 \times 119.6 \times 83.5$
		1555	$162.9 \times 119.9 \times 82.9$
		1548	$162.5 \times 121.6 \times 83.8$
		1552	$162.5 \times 121.9 \times 84.5$
		1552	$162.5 \times 121.8 \times 83.8$
		1555	$163.1 \times 121.0 \times 83.2$
		1557	$162.5 \times 121.2 \times 83.2$
		1554	$165.0 \times 121.5 \times 84.2$
		1530	$147.8 \times 113.6 \times 80.1$
		1514	$142.5 \times 112.3 \times 77.0$
		1515	$142.5 \times 111.2 \times 76.7$
		1514	$140.4 \times 110.8 \times 76.7$
		1522	$139.8 \times 111.5 \times 76.8$
		1517	$141.4 \times 110.7 \times 75.9$
		1518	$140.7 \times 112.2 \times 76.5$
		1513	$141.1 \times 110.5 \times 77.1$
		1520	$145.0 \times 110.6 \times 77.6$
		1546	$156.9 \times 117.8 \times 81.8$

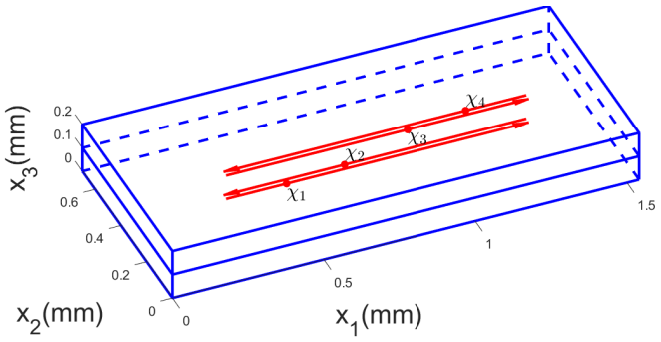


Figure 7.2: The scanning pattern and the selected positions of the two-layer domain.

of the i-layer domain a laser beam first scans from $x_l^{(1)}$ to $x_r^{(1)}$ then from $x_r^{(2)}$ to $x_l^{(2)}$, and the hatch distance between the two lines is Δh. The scanning pattern of all numerical experiments is shown in figure 7.2. Following the same scanning trajectory, the simulations are set with the same spatial and temporal discretization, building platform temperature, heat loss and heat source parameters, the values of which are listed in table 7.4.

Table 7.4: Model parameters in the thermal model of LPBF [13, 14, 108, 123].

Symbol	Definition (Unit)	Value
Δt	A time step in temporal discretization (ms)	0.05
-	The size of the i-layer domain (mm)	$1.54 \times 0.70 \times 0.10i$
$x_l^{(1)}$	The start of the first line on layer i (mm)	$[0.30, 0.29, 0.10i]^T$
$x_r^{(1)}$	The end of the first line on layer i (mm)	$[1.30, 0.29, 0.10i]^T$
$x_l^{(2)}$	The end of the second line on layer i (mm)	$[0.30, 0.31, 0.10i]^T$
$x_r^{(2)}$	The start of the second line on layer i (mm)	$[1.30, 0.31, 0.10i]^T$
Δh	Hatch distance (mm)	0.02
u_a	Ambient temperature (°C)	20
a	Absorptivity	0.09
ς	The effective laser beam radius (μm)	35
u_b	The constant temperature on Γ_b (°C)	200
h	Heat convection coefficient (W/(m^{2}K))	10
ε	Emissivity	0.04

7.1.2 Thermal behaviours of LPBF

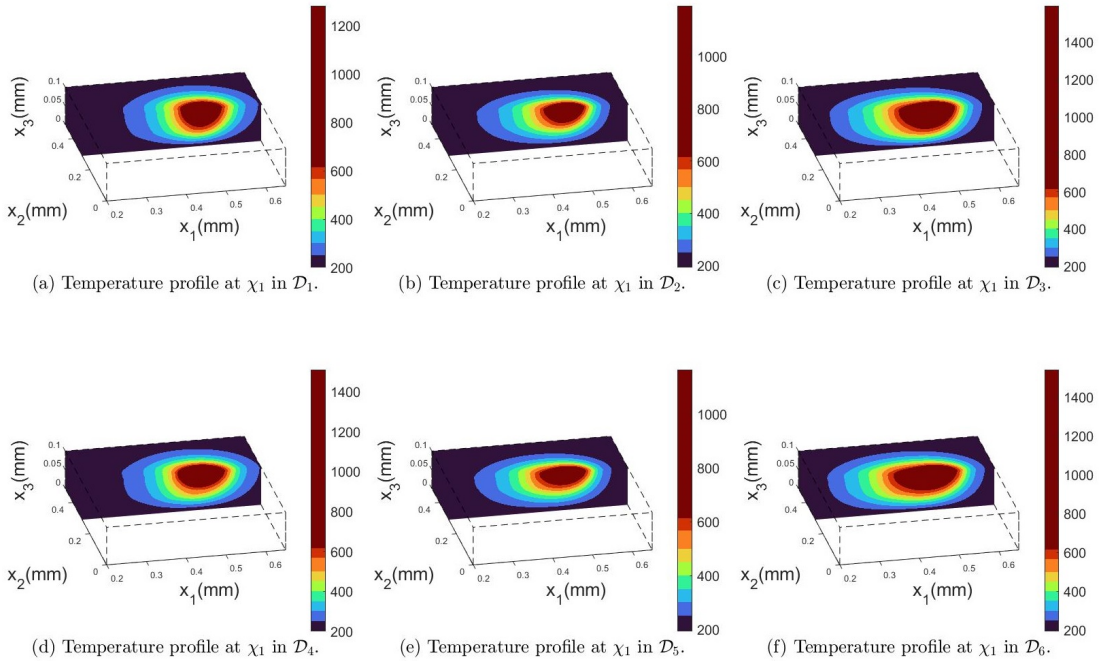
The thermal simulator with FEM manages to characterize temperature distributions, thermal histories, and melt pool sizes during the printing process with

different laser power and scan speed. We hereby present 6 examples with different laser power P, scan speed v, and static/random scale factor of anisotropy λ. These 6 examples are denoted as $\mathcal{D}_i$ for $i = 1, \cdots, 6$, and their parameters are detailed in table 7.5. Four positions are selected to present the simulation results, each of which belongs to different scanning line. They are marked in figure 7.2 and respectively locate at $\chi_1 = [0.5, 0.29, 0.1]^T$, $\chi_2 = [0.7, 0.31, 0.1]^T$, $\chi_3 = [0.9, 0.29, 0.2]^T$, and $\chi_4 = [1.1, 0.31, 0.2]^T$. The temperature distributions of the position χ_i for $i = 1, \cdots, 4$ in all six tests are respectively shown in figure 7.3, 7.4, 7.5, and 7.6. The first and second interval in the colormap respectively includes the temperature range $u \geqslant u_l$ and $u_s \leqslant u < u_l$ and represents the liquid and musy state of the material, while other temperature intervals are $500°\text{C} \leqslant u < u_s$, $460°\text{C} \leqslant u < 500°\text{C}$, $420°\text{C} \leqslant u < 460°\text{C}$, $380°\text{C} \leqslant u < 420°\text{C}$, $340°\text{C} \leqslant u < 380°\text{C}$, $300°\text{C} \leqslant u < 340°\text{C}$, $260°\text{C} \leqslant u < 300°\text{C}$, and $u < 260°\text{C}$. Comparing the subplots (a) ($P = 200\text{W}$, $v = 200\text{mm/s}$) and (b) ($P = 200\text{W}$, $v = 1000\text{mm/s}$), we can tell that with the same laser power a lower scan speed results in higher maximum temperature and larger melt pool size. By comparing the subplots (b) ($P = 200\text{W}$, $v = 1000\text{mm/s}$) and (c) ($P = 300\text{W}$, $v = 1000\text{mm/s}$), it is shown that with the same scan speed a higher laser power results in higher maximum temperature and larger melt pool size. For the subplots in the same column, they have the same laser power and scan speed but the upper row has the static anisotropy scale factor λ while the lower row has random λ. It is observed that the randomness of λ may lead to different highest temperature and anisotropy in an irregular way. In specific, the highest temperatures and melt pool sizes at the four positions χ_i for $i = 1, \cdots, 4$ in the six tests $\mathcal{D}_i$ for $i = 1, \cdots, 6$ are summarized in table 7.6.

Another thermal behaviour is the thermal history of a specific position throughout the printing process. The thermal histories of the four positions χ_i for $i = 1, \cdots, 4$ in the six tests $\mathcal{D}_i$ for $i = 1, \cdots, 6$ are shown and compared in figure 7.7. As it is a process of scanning two lines in two layers, there are four peaks in the thermal histories of the positions on the first layer (χ_1 and χ_2) while the thermal histories of the positions on the second layer (χ_3 and χ_4) have two peaks. We can tell that with a lower scan speed ($v = 200\text{mm/s}$) the heating-up

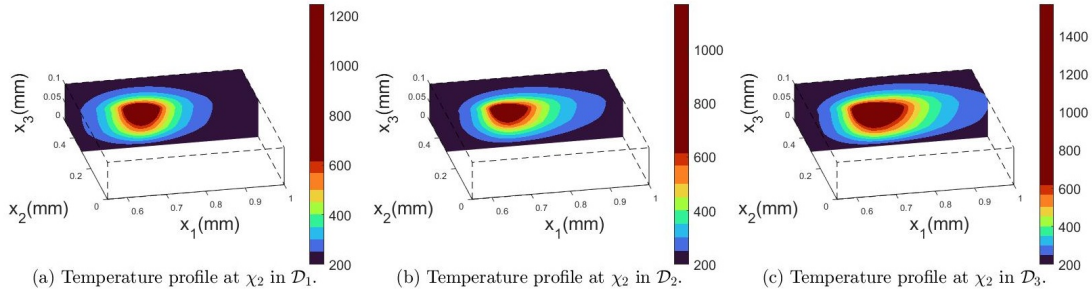
Table 7.5: Parameters of the 6 examples.

Test	Laser power (W)	Scan speed (mm/s)	λ
$\mathcal{D}_1$	200	200	$[2.5, 1, 3.2]^T$
$\mathcal{D}_2$	200	1000	$[2.5, 1, 3.2]^T$
$\mathcal{D}_3$	300	1000	$[2.5, 1, 3.2]^T$
$\mathcal{D}_4$	200	200	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{D}_5$	200	1000	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$
$\mathcal{D}_6$	300	1000	$\lambda_1 \sim U(1,4),\ \lambda_2 \sim U(0.5,1.5),\ \lambda_3 \sim U(1.5,4.5)$

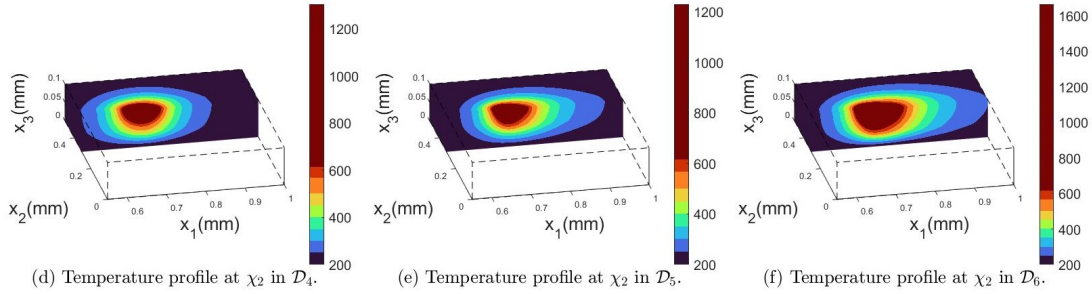


(a) Temperature profile at χ_1 in $\mathcal{D}_1$.
(b) Temperature profile at χ_1 in $\mathcal{D}_2$.
(c) Temperature profile at χ_1 in $\mathcal{D}_3$.
(d) Temperature profile at χ_1 in $\mathcal{D}_4$.
(e) Temperature profile at χ_1 in $\mathcal{D}_5$.
(f) Temperature profile at χ_1 in $\mathcal{D}_6$.

Figure 7.3: Temperature profiles at χ_1 in all six tests.

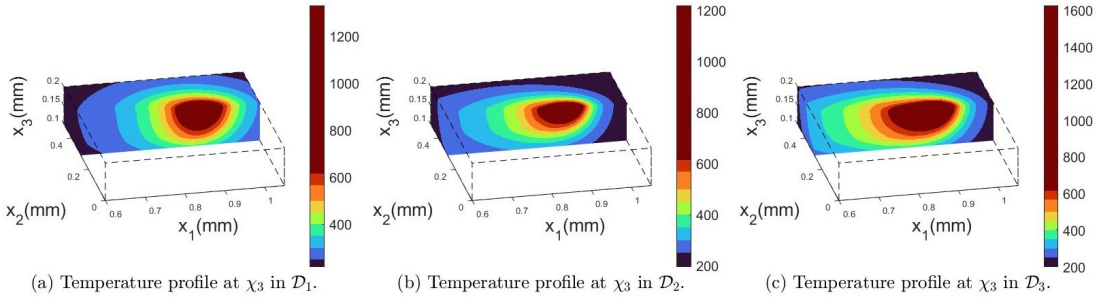


(a) Temperature profile at χ_2 in $\mathcal{D}_1$. (b) Temperature profile at χ_2 in $\mathcal{D}_2$. (c) Temperature profile at χ_2 in $\mathcal{D}_3$.

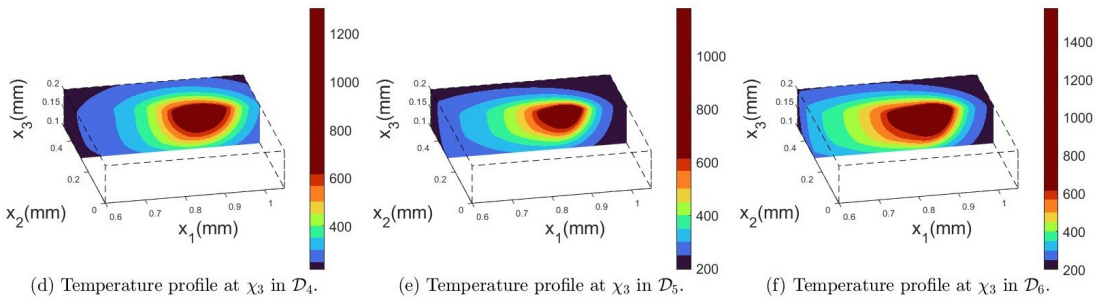


(d) Temperature profile at χ_2 in $\mathcal{D}_4$. (e) Temperature profile at χ_2 in $\mathcal{D}_5$. (f) Temperature profile at χ_2 in $\mathcal{D}_6$.

Figure 7.4: Temperature profiles at χ_2 in all six tests.

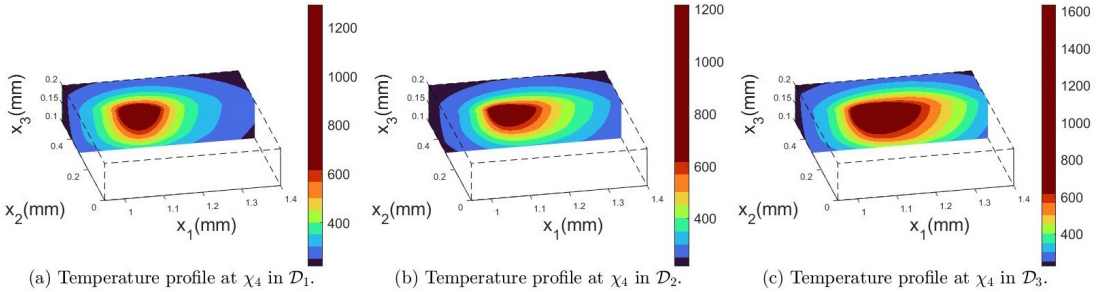


(a) Temperature profile at χ_3 in $\mathcal{D}_1$. (b) Temperature profile at χ_3 in $\mathcal{D}_2$. (c) Temperature profile at χ_3 in $\mathcal{D}_3$.

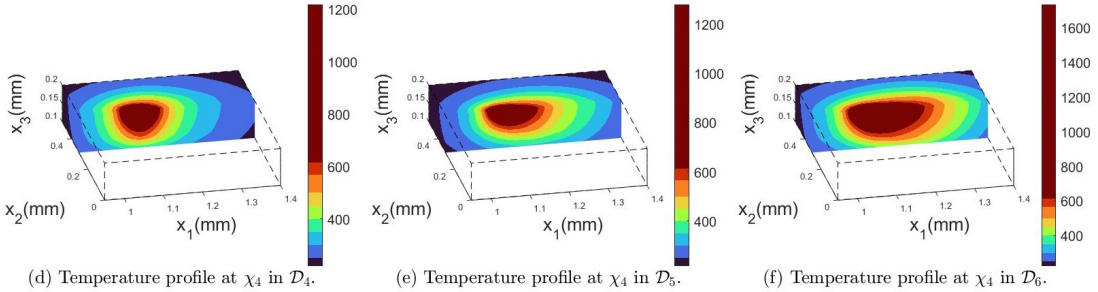


(d) Temperature profile at χ_3 in $\mathcal{D}_4$. (e) Temperature profile at χ_3 in $\mathcal{D}_5$. (f) Temperature profile at χ_3 in $\mathcal{D}_6$.

Figure 7.5: Temperature profiles at χ_3 in all six tests.

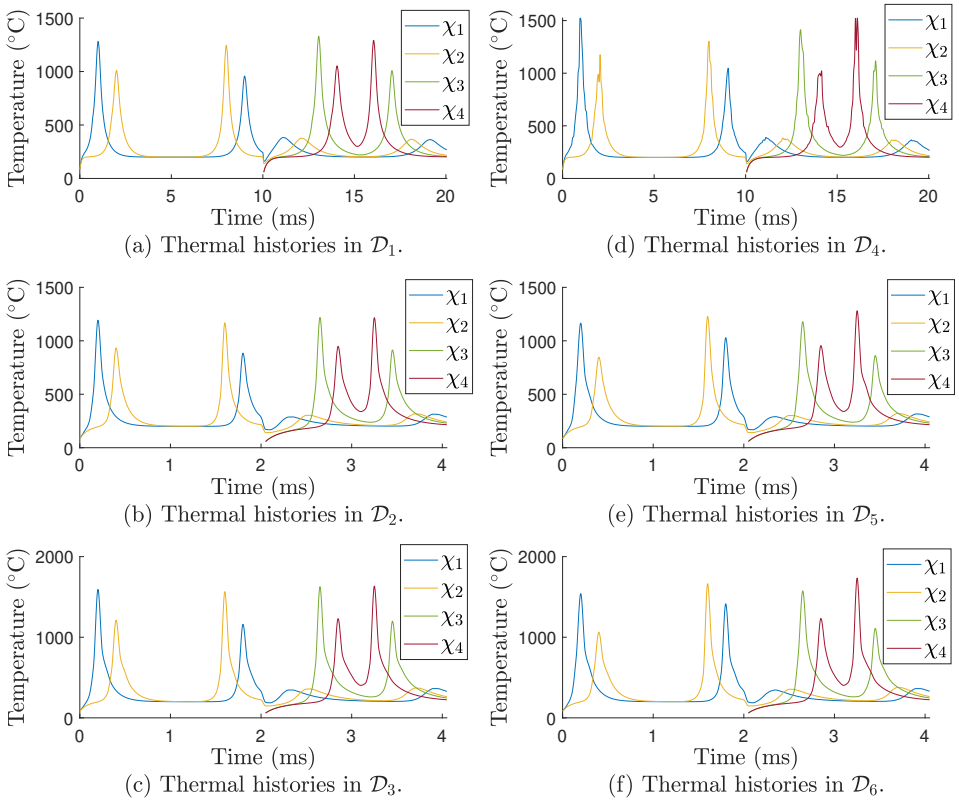


(a) Temperature profile at χ_4 in $\mathcal{D}_1$. (b) Temperature profile at χ_4 in $\mathcal{D}_2$. (c) Temperature profile at χ_4 in $\mathcal{D}_3$.



(d) Temperature profile at χ_4 in $\mathcal{D}_4$. (e) Temperature profile at χ_4 in $\mathcal{D}_5$. (f) Temperature profile at χ_4 in $\mathcal{D}_6$.

Figure 7.6: Temperature profiles at χ_4 in all six tests.



(a) Thermal histories in $\mathcal{D}_1$. (d) Thermal histories in $\mathcal{D}_4$.

(b) Thermal histories in $\mathcal{D}_2$. (e) Thermal histories in $\mathcal{D}_5$.

(c) Thermal histories in $\mathcal{D}_3$. (f) Thermal histories in $\mathcal{D}_6$.

Figure 7.7: Thermal histories at the four selected positions in all tests.

and cooling-down speeds of a position are closer while with a higher scan speed ($v = 1000$mm/s) the cooling-down speed is obviously slower than its heating-up speed. With a high scan speed the laser beam arrives and leaves fast, fast arrival

Table 7.6: Highest temperatures and melt pool sizes at the selection positions in all tests.

Position	Test	Highest temperature (°C)	Melt pool size (μm)		
			Length	Width	Depth
χ_1	$\mathcal{D}_1$	1285.82	115.89	94.11	56.57
	$\mathcal{D}_2$	1198.69	115.00	80.89	44.91
	$\mathcal{D}_3$	1601.10	156.23	95.81	60.27
	$\mathcal{D}_4$	1518.07	130.05	104.57	45.80
	$\mathcal{D}_5$	1171.93	135.62	85.45	40.41
	$\mathcal{D}_6$	1549.71	182.50	100.27	53.75
χ_2	$\mathcal{D}_1$	1249.28	102.50	86.98	51.08
	$\mathcal{D}_2$	1174.10	105.00	77.46	40.75
	$\mathcal{D}_3$	1573.55	147.50	92.16	58.38
	$\mathcal{D}_4$	1308.97	110.00	84.00	44.88
	$\mathcal{D}_5$	1235.93	102.84	79.81	46.29
	$\mathcal{D}_6$	1673.56	135.98	94.82	63.83
χ_3	$\mathcal{D}_1$	1333.99	140.00	107.23	73.00
	$\mathcal{D}_2$	1223.63	135.00	87.38	51.25
	$\mathcal{D}_3$	1633.00	190.85	103.23	71.55
	$\mathcal{D}_4$	1308.84	160.00	97.93	69.32
	$\mathcal{D}_5$	1184.22	130.42	81.73	57.25
	$\mathcal{D}_6$	1580.86	185.95	98.92	77.86
χ_4	$\mathcal{D}_1$	1296.33	121.34	98.19	64.68
	$\mathcal{D}_2$	1222.35	142.50	85.83	54.52
	$\mathcal{D}_3$	1642.65	206.03	109.65	77.44
	$\mathcal{D}_4$	1222.16	117.88	95.20	70.66
	$\mathcal{D}_5$	1287.44	150.00	83.67	52.49
	$\mathcal{D}_6$	1742.34	224.19	103.94	76.08

makes the position quickly reach its maximum temperature while the cooling is less affected by the laser beam that has already quickly moved away. With a low scan speed, however, as the effect of the laser beam dominates the temperature changes, the heating and cooling that happen when the laser beam slowly approaches and leaves are realized at a similar speed. In figure 7.7, the subplots in the same row have the same laser power and scan speed but the left ones have static anisotropy scale factors λ while the right ones have random λ. It is shown

that the randomness of λ causes more significant effects on the thermal histories with lower scan speed as it has more time steps enabling it to bring in more randomness.

7.2 Reduced Gaussian process surrogate

7.2.1 Data generation and parameter setup

A data-driven relative distance predictor is used in both $\mathcal{F}_1$ and $\mathcal{F}_2$, the data generator of which is the thermal simulator with FEM. As discussed before the training input is of the form $X_{*j} = [P_j, v_j, v_j t_j]^T$ for $j = 1, \cdots, N$ containing laser power P, scan speed v, and time t. In the absence of the information on printing parameter distributions, we apply a uniform grid search in the admissible range of laser power P: 200-300W with an interval of 50W, scan speed v: 200-1000mm/s with an interval of 200mm/s, and time t: the whole printing process with a fixed time interval $\Delta t = 0.05$ms. With this sampling scheme, we construct the training data set of the i-layer domain $X^{(i)} \in \mathbb{R}^{3\times 1383}$ and $U^{(i)} \in \mathbb{R}^{n_d \times 1383}$ for $i = 1, 2$. The dimensionality of training temperatures n_d, however, is the number of DoF in the i-layer domain. Additionally, there are a total amount of 16 tests including different laser power (P=225, 275W), scan speed (v=300, 500, 700, 900mm/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 [13], 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. All training data are generated with the static anisotropy scale factor λ, which makes the data-driven relative distance predictor lacking of the knowledge of anisotropy randomness. Therefore, with the 16 tests denoted as $\mathcal{S}_i$ for $i = 1, \cdots, 16$ we validate the feasibility of our surrogate to cope with different printing parameters and test its ability in enduring a certain level of anisotropy randomness. The parameter settings of each test are specified in table 7.7, and the test inputs and the corresponding reference temperature are respectively $X_t^{(i)} \in \mathbb{R}^{3\times 1272}$ and

$U_t^{(i)} \in \mathbb{R}^{n_d \times 1272}$. Other model parameters used in thermal modeling have been listed in table 7.4. The parameters needed in $\mathcal{F}_1$ are specified in table 7.8. For the convenience of comparison, the performance of $\mathcal{F}_3$ is also evaluated with the inputs X_t. All tests are implemented in Matlab R2020b on a computer with 2.6GHz 6-Core Intel Core i7 processor and 16GB RAM.

Table 7.7: Parameters of the 16 tests.

Test	Laser power (W)	Scan speed (mm/s)	λ
$\mathcal{S}_1$	225	300	$[2.5, 1, 3.2]^T$
$\mathcal{S}_2$	225	500	$[2.5, 1, 3.2]^T$
$\mathcal{S}_3$	225	700	$[2.5, 1, 3.2]^T$
$\mathcal{S}_4$	225	900	$[2.5, 1, 3.2]^T$
$\mathcal{S}_5$	275	300	$[2.5, 1, 3.2]^T$
$\mathcal{S}_6$	275	500	$[2.5, 1, 3.2]^T$
$\mathcal{S}_7$	275	700	$[2.5, 1, 3.2]^T$
$\mathcal{S}_8$	275	900	$[2.5, 1, 3.2]^T$
$\mathcal{S}_9$	225	300	$\lambda_1 \sim U(1,4)$, $\lambda_2 \sim U(0.5,1.5)$, $\lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{10}$	225	500	$\lambda_1 \sim U(1,4)$, $\lambda_2 \sim U(0.5,1.5)$, $\lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{11}$	225	700	$\lambda_1 \sim U(1,4)$, $\lambda_2 \sim U(0.5,1.5)$, $\lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{12}$	225	900	$\lambda_1 \sim U(1,4)$, $\lambda_2 \sim U(0.5,1.5)$, $\lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{13}$	275	300	$\lambda_1 \sim U(1,4)$, $\lambda_2 \sim U(0.5,1.5)$, $\lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{14}$	275	500	$\lambda_1 \sim U(1,4)$, $\lambda_2 \sim U(0.5,1.5)$, $\lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{15}$	275	700	$\lambda_1 \sim U(1,4)$, $\lambda_2 \sim U(0.5,1.5)$, $\lambda_3 \sim U(1.5,4.5)$
$\mathcal{S}_{16}$	275	900	$\lambda_1 \sim U(1,4)$, $\lambda_2 \sim U(0.5,1.5)$, $\lambda_3 \sim U(1.5,4.5)$

Table 7.8: Parameters in $\mathcal{F}_1$.

Symbol	Definition	Value
g	The size of neighborhood	20
$n_r^{'}$	The number of training data selection	5
ϱ	The constant to adjust the decline of weights	0.001

7.2.2 Dimension reduction

In the pre-processing of $\mathcal{F}_1$, the number of GP is reduced by a nonlinear dimension reduction technique. There are two important values in this method: one is the neighbourhood size g, the value of which should keep the reachability between

vertices in the weighted graph. In our case, as stated in table 7.8 we set $g = 20$. Another one is the target of dimension reduction $\bar{n}_r$. It takes the minimum value that satisfies $(\sum_{i=1}^{\bar{n}_r} \Lambda_{ii})/(\sum_{i=1}^{N} \Lambda_{ii}) > 99\%$ where Λ is the singular value diagonal matrix in equation (4.3). With the training temperature snapshots $U^{(i)}$ for $i = 1, 2$, the reduced dimension, as in table 7.9 , are respectively 276 and 242. In comparison with the number of training data $N = 1383$, it is a dramatic reduction respectively cutting down 80.04% and 82.50% of GPs required thus less effort in both training and prediction.

Table 7.9: The dimension reduction in $\mathcal{F}_1$.

Layer	Reduced dimension	Reduction percentage (%)
1	276	80.04
2	242	82.50

7.2.3 Model accuracy and robustness

Given test inputs $X_t^{(i)}$ in the i-layer domain for $i = 1, 2$, we predict the corresponding temperatures as $\hat{U}_t^{(i)} = \mathcal{F}_1(X_t^{(i)})$. The model accuracy is compared

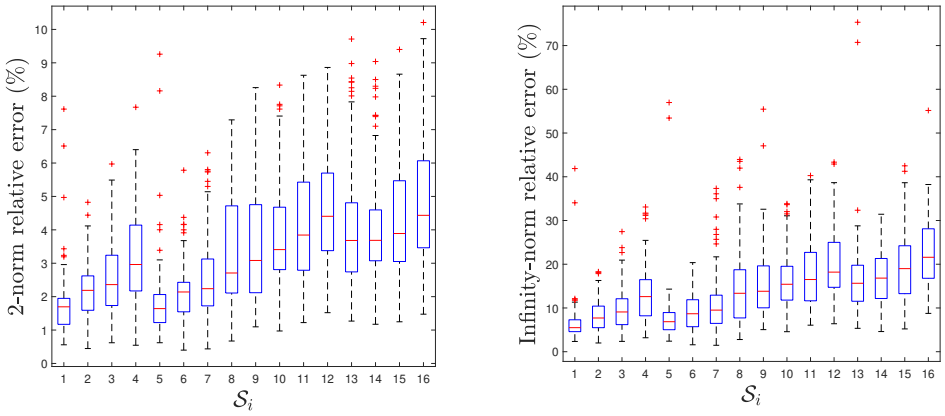


Figure 7.8: The boxplot of relative errors between FEM and $\mathcal{F}_1$.

in three aspects: relative errors, melt pools, and thermal histories. The relative errors are computed in 2-norm and infinity-norm, both of which 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 a surrogate, we have the 2-norm relative error as

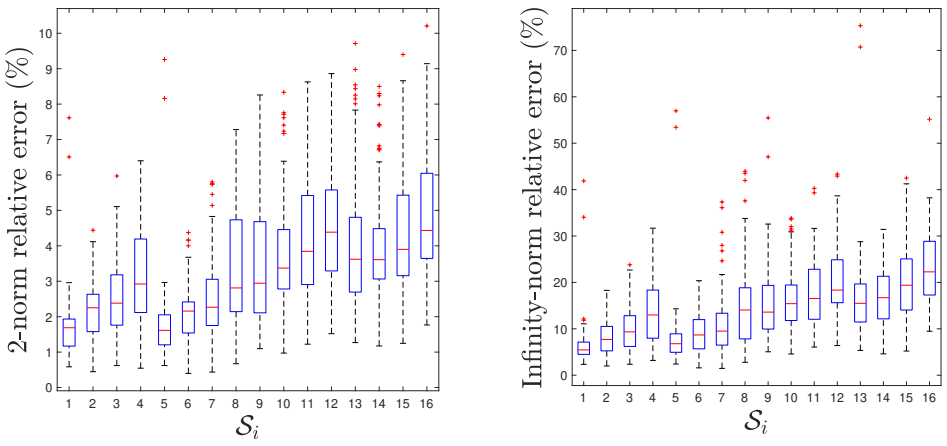


Figure 7.9: The boxplot of relative errors between FEM and $\mathcal{F}_1$ without the first five time steps of each scanning line.

$e = \|u - \hat{u}\|_2 / \|u\|_2$ and the infinity-norm relative error as $e_\infty = \|e^c\|_\infty$ where $e_i^c = |u_i - \hat{u}_i| / u_i$ for $i = 1, \cdots n_d$. The spread of relative errors is shown in the boxplot as figure 7.8 where the ith box expresses the relative errors of the test $\mathcal{S}_i$ for $i = 1, \cdots, 16$. In each box, the red line represents the median (Q_2) while the bottom and top lines 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 $1.5 \times (Q_3 - Q_1)$ where $Q_3 - Q_1$ is also known as the interquartile range (IQR). All values beyond the two whiskers are marked by red crosses representing outliers. At the beginning of each scanning line, there is not enough heat to melt enough materials and form regular temperature distributions. Therefore, the selected training data may not be competent to predict the temperature distribution we want. In figure 7.9, the boxplots of relative errors without the first five time steps of each scanning line are shown. Though some outliers are removed, there are still some predictions observed with comparably large relative errors. While for $\mathcal{F}_2$ the effect of picking out the first five time steps of each scanning line is more obvious, the larger errors in figure 7.14 are removed in figure 7.15. It indicates that some larger errors in $\mathcal{F}_1$ are caused by inaccurate predictions of the order and exact value of relative distances instead of the selection of training data.

We can tell from figure 7.8 that though the majority of temperatures can be accurately predicted some predictions have very high errors. Specifically, there

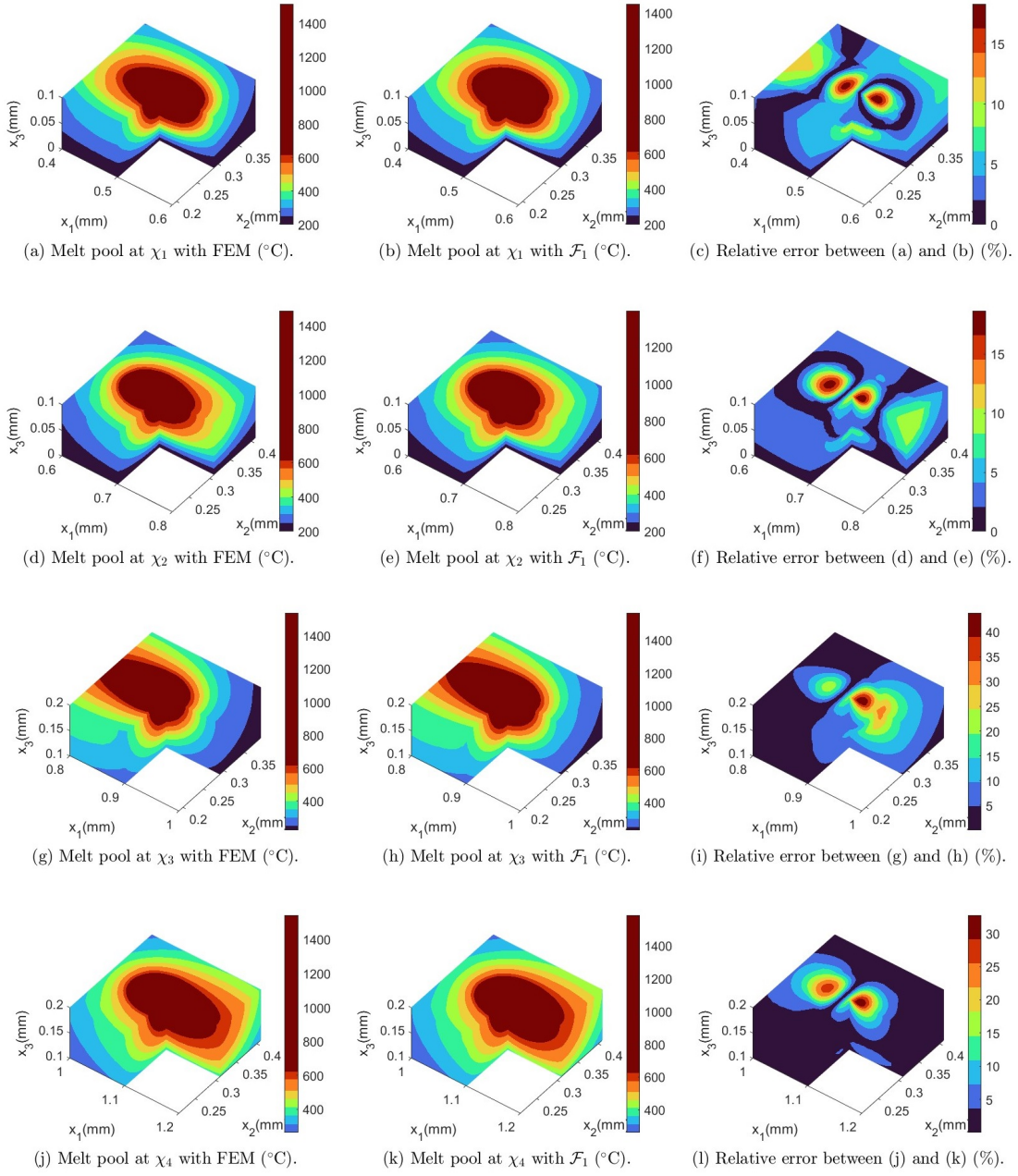
are respectively 4.52%, 30.07%, and 85.30% of tests with 2-norm relative errors less than 1%, 2%, and 5%, while the maximum relative error is 10.21%. 41.94%, 66.51%, and 97.01% of infinity-norm relative errors are respectively less than 10%, 15%, and 30%, and the maximum infinity-norm relative error is 75.33%. The first 8 tests $\mathcal{S}_i$ for $i = 1, \cdots, 8$ having the static anisotropy scale factor λ are observed to be more accurate than the last 8 tests $\mathcal{S}_i$ for $i = 9, \cdots, 16$ having the random λ. The comparison of the 2-norm and infinity-norm relative errors between using a static/random λ is shown in table 7.10 where $n_t = 0$ and 5 respectively indicates the relative errors with and without the first five time steps of each scanning line. It tells that the exclusion of the first five time steps of each

Table 7.10: The accuracy comparison of $\mathcal{F}_1$ between using a static or random λ.

n_t	λ	2-norm relative error (%)				Infinity-norm relative error (%)			
		Max	$< 1\%$	$< 2\%$	$< 5\%$	Max	$< 10\%$	$< 15\%$	$< 30\%$
0	Static	9.26	8.96	51.34	96.46	56.97	70.44	90.33	98.35
	Random	10.21	0.08	8.81	74.14	75.33	13.44	42.69	95.68
5	Static	9.26	9.40	52.30	96.90	56.97	71.72	90.16	98.40
	Random	10.21	0.09	8.87	75.53	75.33	14.01	42.55	95.57

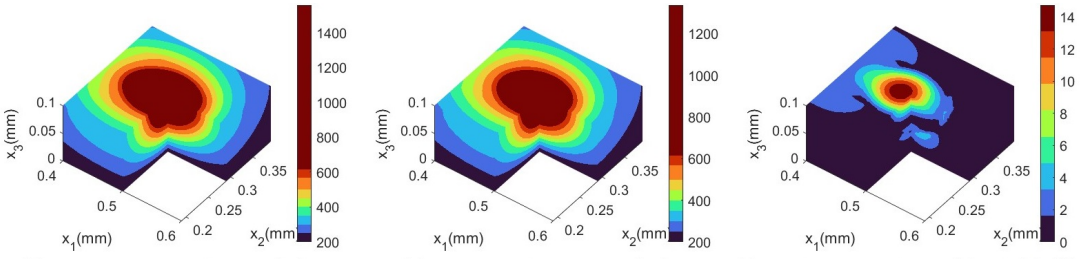
scanning line does not have a significant effect on the model accuracy and $\mathcal{F}_1$ is not competent in dealing with anisotropy randomness. While the percentage of tests having 2-norm relative errors less than 2% significantly drops from 51.34% to 8.81% when λ is random, there are only 42.69% of tests with a random λ satisfying $e_\infty < 15\%$ but the corresponding percentage for a static λ is 90.33%. Also, when λ is random instead of static, the maximum of e and e_∞ respectively increases from 9.26% to 10.21% and from 56.97% to 75.33%.

Despite the relative error of temperature profiles, we also evaluate the prediction accuracy of melt pools and thermal histories. Take $\mathcal{S}_8$ and $\mathcal{S}_9$ as examples, the temperatures around the melt pools at χ_i for $i = 1, \cdots, 4$ are respectively displayed as figure 7.10 and 7.11. The maximum of relative errors around χ_i for $i = 1, \cdots, 4$ are respectively 18.30%, 18.65%, 43.87%, and 32.74% in $\mathcal{S}_8$ while in $\mathcal{S}_9$ they are correspondingly 14.75%, 12.13%, 13.25%, and 8.86%. In $\mathcal{S}_8$, the highest temperatures around the position χ_i for $i = 1, \cdots, 4$ are respectively 1517.59°C, 1490.11°C, 1540.79°C, and 1548.21°C with FEM while the corre-

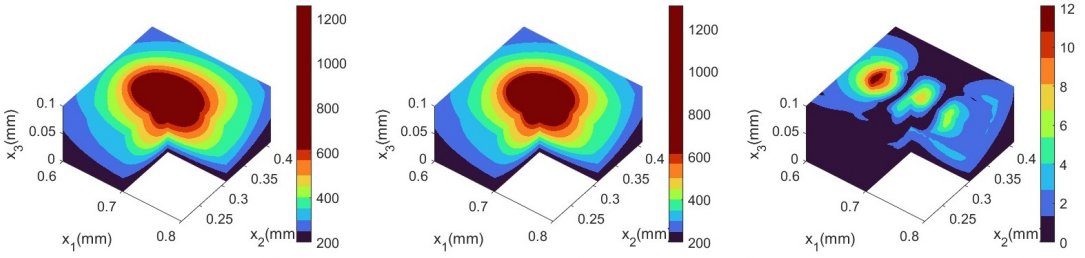


(a) Melt pool at χ_1 with FEM (°C).
(b) Melt pool at χ_1 with $\mathcal{F}_1$ (°C).
(c) Relative error between (a) and (b) (%).
(d) Melt pool at χ_2 with FEM (°C).
(e) Melt pool at χ_2 with $\mathcal{F}_1$ (°C).
(f) Relative error between (d) and (e) (%).
(g) Melt pool at χ_3 with FEM (°C).
(h) Melt pool at χ_3 with $\mathcal{F}_1$ (°C).
(i) Relative error between (g) and (h) (%).
(j) Melt pool at χ_4 with FEM (°C).
(k) Melt pool at χ_4 with $\mathcal{F}_1$ (°C).
(l) Relative error between (j) and (k) (%).

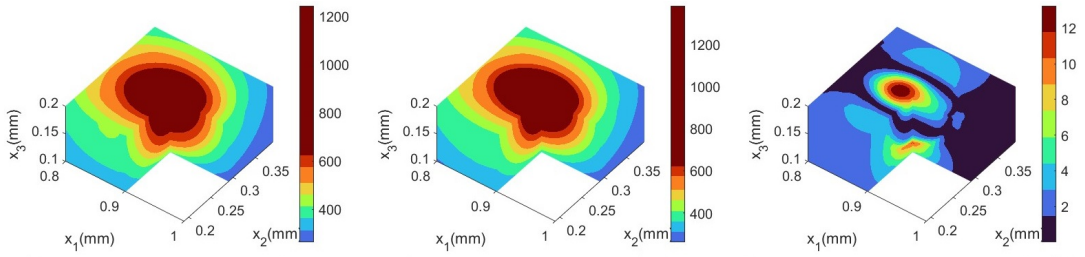
Figure 7.10: The melt pool comparison between FEM and $\mathcal{F}_1$ in the test $\mathcal{S}_8$.



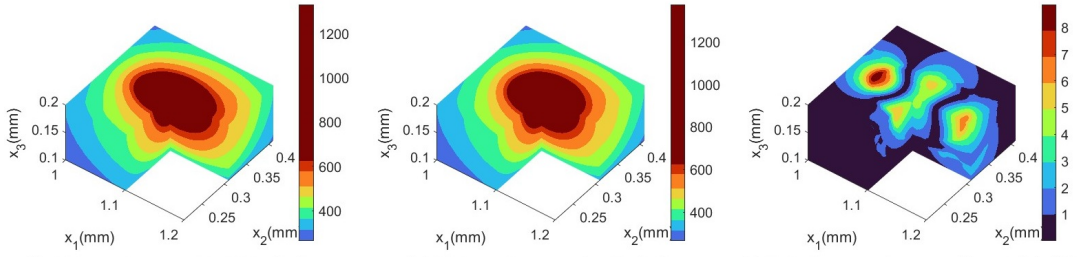
(a) Melt pool at χ_1 with FEM (°C). (b) Melt pool at χ_1 with $\mathcal{F}_1$ (°C). (c) Relative error between (a) and (b) (%).



(d) Melt pool at χ_2 with FEM (°C). (e) Melt pool at χ_2 with $\mathcal{F}_1$ (°C). (f) Relative error between (d) and (e) (%).



(g) Melt pool at χ_3 with FEM (°C). (h) Melt pool at χ_3 with $\mathcal{F}_1$ (°C). (i) Relative error between (g) and (h) (%).



(j) Melt pool at χ_4 with FEM (°C). (k) Melt pool at χ_4 with $\mathcal{F}_1$ (°C). (l) Relative error between (j) and (k) (%).

Figure 7.11: The melt pool comparison between FEM and $\mathcal{F}_1$ in the test $\mathcal{S}_9$.

sponding highest temperatures with $\mathcal{F}_1$ are 1451.79°C, 1399.11°C, 1574.90°C, and 1587.89°C with relative errors 4.34%, 6.11%, 2.21%, and 2.56%. In $\mathcal{S}_9$, with FEM the highest temperatures are respectively 1559.23°C, 1259.34°C, 1244.87°C, and 1331.57°C while with $\mathcal{F}_1$ they are estimated as 1338.92°C, 1312.15°C, 1385.95°C, and 1378.00°C with relative errors 14.13%, 4.19%, 11.33%, and 3.49%. It is also observed that the position of the highest temperature may not be the exact position where the laser beam centre locates especially with the surrogate model. The melt pool sizes at these four positions are listed and compared in table 7.11. No melt pool sizes are precisely approximated (like with a relative error less than 0.01%). The relative errors of melt pool size approximation in $\mathcal{S}_8$ ranges from 0.11% to 9.99%, while in $\mathcal{S}_9$ the range is from 1.10% to 12.27%. The thermal histories at χ_i for $i = 1, \cdots, 4$ are compared in figure 7.12 for $\mathcal{S}_8$ and figure 7.13 for $\mathcal{S}_9$. Their relative errors are respectively 6.64%, 8.37%, 7.88%, and 9.14% in $\mathcal{S}_8$, and in $\mathcal{S}_9$ the corresponding relative errors are 9.20%, 6.41%, 9.50%, and 6.53%. The effect of anisotropy randomness on thermal histories is obvious

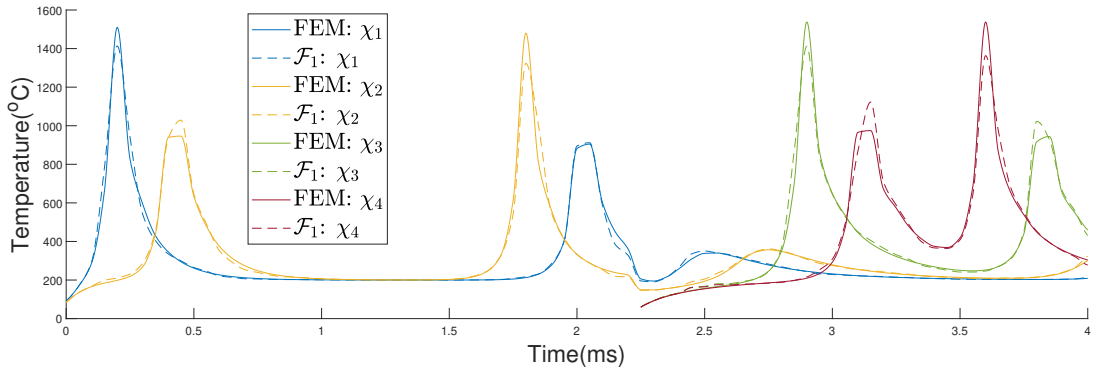


Figure 7.12: The thermal history comparison between FEM and $\mathcal{F}_1$ in $\mathcal{S}_8$.

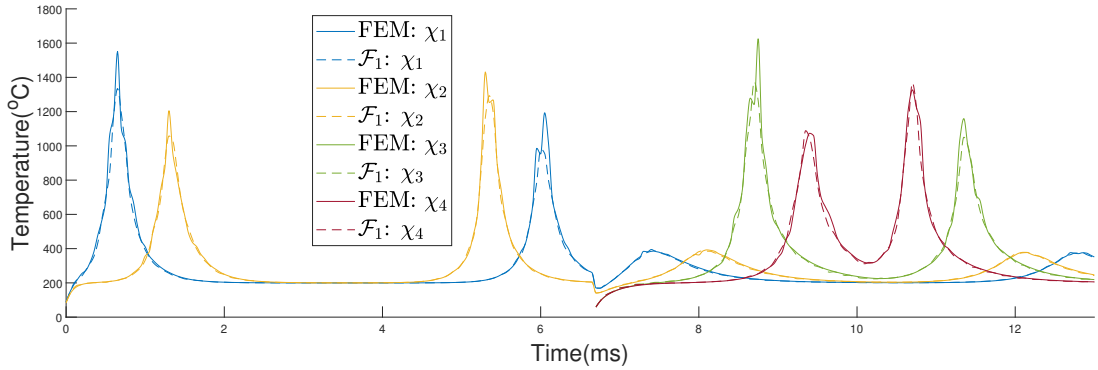


Figure 7.13: The thermal history comparison between FEM and $\mathcal{F}_1$ in $\mathcal{S}_9$.

in $\mathcal{S}_9$ but the results from $\mathcal{F}_1$ are not capable of predicting these randomnesses since the training data of data-driven relative distance predictor only contains the temperatures with the static anisotropy scale factor λ.

Table 7.11: The comparison of melt pool sizes between FEM and $\mathcal{F}_1$ in the test $\mathcal{S}_8$ and $\mathcal{S}_9$.

Test	Position	Melt pool size	FEM (μm)	$\mathcal{F}_1$ (μm)	Relative error (%)
$\mathcal{S}_8$	χ_1	Length	148.20	133.39	9.99
		Width	94.19	103.28	9.65
		Depth	59.59	63.07	5.84
	χ_2	Length	135.00	123.56	8.48
		Width	91.18	93.95	3.04
		Depth	56.43	58.42	3.53
	χ_3	Length	179.51	180.00	0.27
		Width	103.63	103.23	0.39
		Depth	69.45	69.52	0.11
	χ_4	Length	188.27	172.55	8.35
		Width	105.09	110.25	4.91
		Depth	73.62	76.08	3.34
$\mathcal{S}_9$	χ_1	Length	130.00	122.50	5.77
		Width	96.69	95.29	1.45
		Depth	55.89	58.21	4.16
	χ_2	Length	122.87	107.79	12.27
		Width	86.98	90.26	3.77
		Depth	55.80	53.26	4.55
	χ_3	Length	143.24	145.39	1.50
		Width	118.77	109.69	7.65
		Depth	82.16	73.70	10.29
	χ_4	Length	147.50	131.34	10.96
		Width	94.81	104.22	9.93
		Depth	66.59	67.32	1.10

7.2.4 Offline preparation

The offline preparation required includes data generation, nonlinear dimension reduction, and training. The time cost of data generation is the total time of running 15 simulations with laser power $P = 200$, 250, 300W, scan speed $v = 200$, 400, 600, 800, and 1000mm/s, and $\Delta t = 0.05$ms. The time cost of nonlinear di-

mension reduction includes establishing the weighted graph with a neighbourhood size of 20, computing the shortest distance with Floyd–Warshall algorithm [113], and getting the reduced representation of the pair-wise relative distance. The training time contains the implementation of the maximum likelihood estimation of the strictly positive hyperparameters $\hat{\Theta} \in \mathbb{R}^{5\times\bar{n}_r}$ and the computation of the matrix S as in equation (4.13). Though these are offline efforts, they are also the necessary resource cost to fulfil $\mathcal{F}_1$. The offline time cost of $\mathcal{F}_1$ is summarized in table 7.12.

Table 7.12: The offline time cost of $\mathcal{F}_1$.

Layer	Data generation (day)	Nonlinear dimension reduction (s)	Training (hr)
1	7.30	55.10	1.10
2	9.28	53.49	0.85

7.2.5 Time cost reduction

At each time step of the high-fidelity thermal model with FEM, the online execution 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. With the level of accuracy compromise described in section 7.2.3, the predictions of high-dimensional temperatures are accomplished almost in real-time since only swift computations are left for online execution. In specific, the comparison of time cost is listed as table 7.13. Though the preparation described in section 7.2.4 is required offline, the online execution time is respectively reduced from 459.90s to 0.12s and from 597.72s to 0.11s on average in the one- to two-layer domain. The real-time implementation benefits from three characteristics of $\mathcal{F}_1$. First, the purely data-driven approach skips the burdensome computation caused by nonlinearity like Picard iteration and Gaussian quadratures. Second, only a comparably small number of GPs are required after nonlinear dimension reduction, which cuts the time cost of offline training and online prediction. Third, the prediction as equation

(4.11) manages to simultaneously predict all GPs by executing the dot product between matrices instead of computing the univariate GP individually. While the performance in reducing online execution time is satisfactory, the compromise of accuracy is evident. Therefore, $\mathcal{F}_1$ is suitable for applications that ask for real-time implementations but are endurable to some accuracy compromise.

Table 7.13: The average execution time of FEM and $\mathcal{F}_1$.

Layer	FEM (s)	$\mathcal{F}_1$ (s)
1	459.90	0.12
2	597.72	0.11

7.3 Sketched surrogate with data-driven local projection

7.3.1 Parameter setup

For the convenience of comparison, the same test inputs $X_t^{(i)}$ are used to develop temperature predictions and evaluate the performances. For each pair of test input $X_{t_{*j}}^{(i)}$ for $j = 1, \cdots, 1272$, a local projection basis is formed by selecting n_r temperature snapshots from $U^{(i)}$ that are predicted as the closest ones to $U_{t_{*j}}^{(i)}$. While the neighbourhood size $g = 20$ and the constant to adjust the weight decline $\varrho = 0.001$ are the same as $\mathcal{F}_1$ listed in table 7.8, other parameters necessary for $\mathcal{F}_2$ are specified in table 7.14. It is noticed that in randomized sketching with a smaller projection dimension n_r the multiplier of leverage score ι that manages to accurately approximate the model can be smaller, which makes $\mathcal{F}_2$'s ι in table 7.14 become smaller than $\mathcal{F}_3$'s ι in table 7.21.

Table 7.14: Parameters of $\mathcal{F}_2$.

Symbol	Definition	Value
n_r	the dimension of projection	30
ι	the multiplier of leverage score	50000

7.3.2 Model accuracy and robustness

The test temperatures $U_t^{(i)}$ are estimated with $\mathcal{F}_2$ as $\hat{U}_t^{(i)} = \mathcal{F}_2(X_t^{(i)})$. The spread of 2-norm and infinity-norm relative errors are shown as the boxplots in figure 7.14. Among all tests, the maximum of 2-norm and infinity-norm relative errors are respectively 7.99% and 65.69%. There are respectively 46.31%, 69.73%, and 97.76% of tests with 2-norm relative errors less than 1%, 2%, and 5%, while 74.45%, 89.31%, and 99.41% of tests respectively have the infinity-norm relative errors less than 10%, 15%, and 30%. Though the projection bases in $\mathcal{F}_2$ do not rely on previous temperatures in the simulation like the inclusion of previous temperatures and the Gaussian function calibration in $\mathcal{F}_3$, these data-driven local projection bases are not very accurate for the first several time steps. With

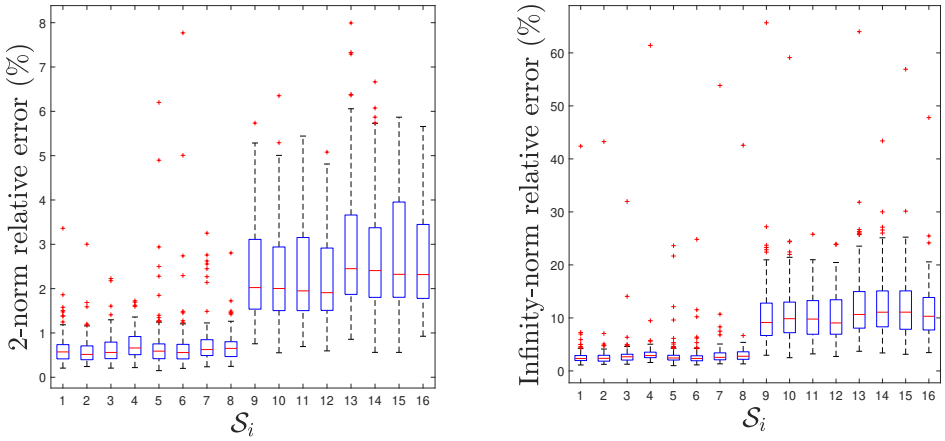


Figure 7.14: The relative errors between FEM and $\mathcal{F}_2$.

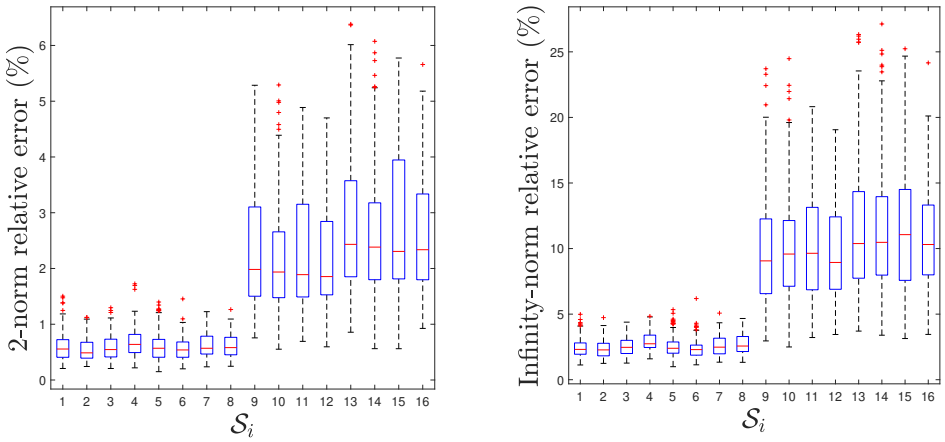


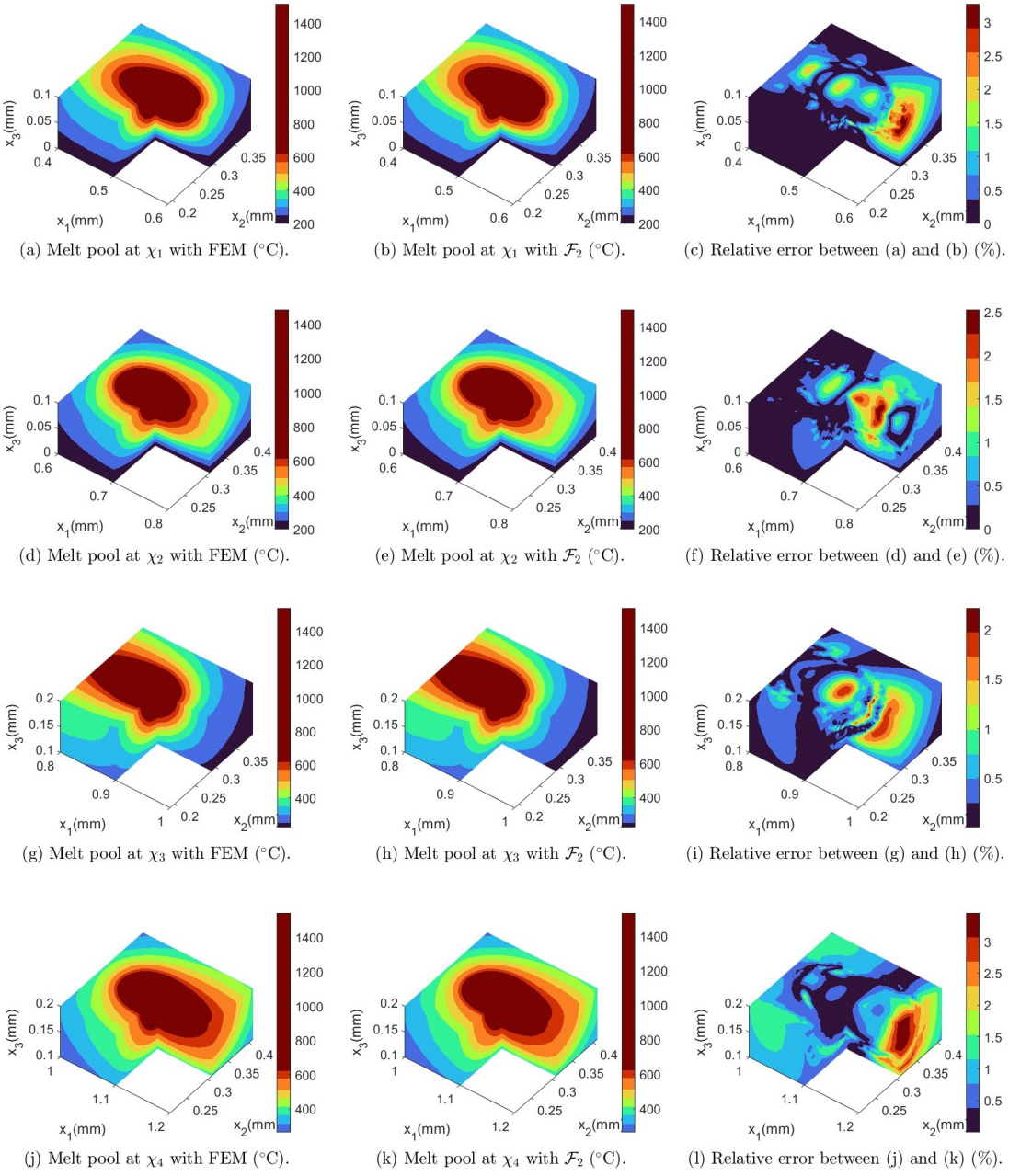
Figure 7.15: The relative errors between FEM and $\mathcal{F}_2$ without the first 5 time steps of each scanning line.

a small time interval $\Delta t = 0.05$ms there is not enough heat to melt many materials at the beginning, which results in irregular melt pools and temperature distributions. The selected training temperatures though deemed to be similar to the final temperature are not competent to fulfill accurate approximation. The relative errors excluding the first 5 time steps of each scanning line are shown as the boxplots in figure 7.15, from which we can tell that $\mathcal{F}_2$ is quite accurate for the tests with static anisotropy scale factors. Among all tests with static λ ($\mathcal{S}_i$ for $i = 1, \cdots, 8$), the maximum 2-norm and infinity-norm relative errors are respectively 1.72% and 6.19%. Though the tests with random λ ($\mathcal{S}_i$ for $i = 9, \cdots, 16$) are less accurate, it is still a significant improvement where the maximum 2-norm and infinity-norm relative errors are respectively 6.38% and 27.12%. To compare the model accuracy in the situation with static/random λ and with ($n_t = 0$) /without ($n_t = 5$) the first 5 time steps, the relative errors of all tests are summarized in table 7.15. It is obvious that the performance of $\mathcal{F}_2$ drops significantly

Table 7.15: The accuracy comparison of $\mathcal{F}_2$ between using a static or random λ.

n_t	λ	2-norm relative error (%)				Infinity-norm relative error (%)			
		Max	$< 1\%$	$< 2\%$	$< 5\%$	Max	$< 10\%$	$< 15\%$	$< 30\%$
0	Static	7.77	89.62	98.35	99.76	61.41	98.90	99.29	99.53
	Random	7.99	2.99	41.12	95.75	65.69	50.00	79.32	99.29
5	Static	1.72	93.53	100.00	100.00	6.19	100.00	100.00	100.00
	Random	6.38	3.10	42.73	96.63	27.12	51.60	82.18	100.00

when λ is random. The percentage of tests with $e < 1\%$ falls from 89.62% to 2.99%, and there are only 41.12% of tests with random λ satisfying $e < 2\%$ in comparison with 98.35% in the case with static λ. Similarly, for the infinity-norm relative error the randomness in λ makes the percentage of tests with $e_\infty < 10\%$ decreasing from 98.90% to 50.00% while with $e_\infty < 15\%$ the drops is from 99.29% to 79.32%. Since $\mathcal{F}_1$ is not agile to handle anisotropy randomness and $\mathcal{F}_2$ also uses the same data-driven relative distance predictor in $\mathcal{F}_1$ to form projection bases, it is expected that $\mathcal{F}_2$ is also not robust enough to random λ and the adding of randomness in λ causes an evident drop of the model accuracy performance of $\mathcal{F}_2$. The melt pools around the four selected positions in the test $\mathcal{S}_8$ and $\mathcal{S}_9$ are respectively delineated in figure 7.16 and 7.17, and the approximation of melt



(a) Melt pool at χ_1 with FEM (°C). (b) Melt pool at χ_1 with $\mathcal{F}_2$ (°C). (c) Relative error between (a) and (b) (%).

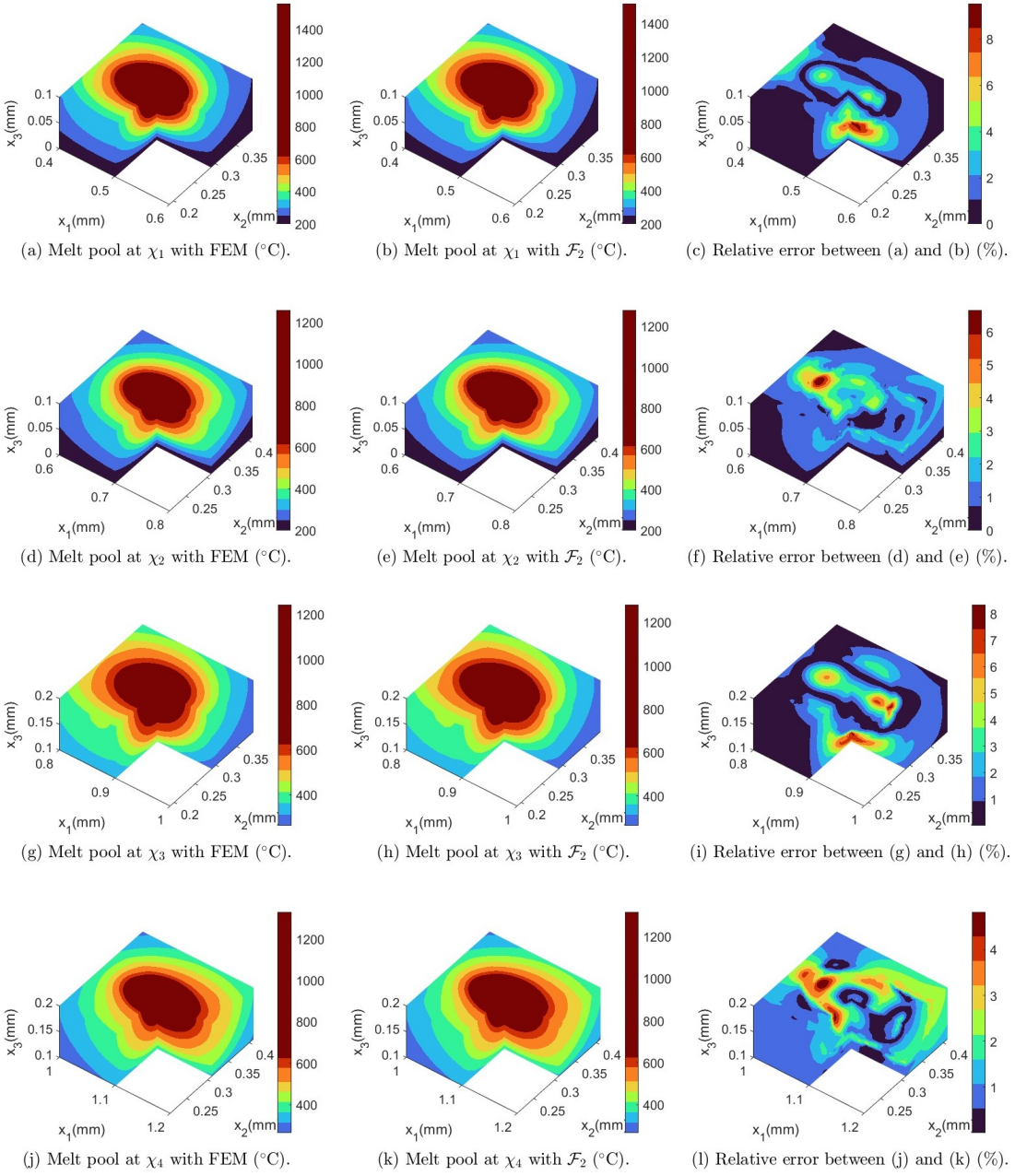
(d) Melt pool at χ_2 with FEM (°C). (e) Melt pool at χ_2 with $\mathcal{F}_2$ (°C). (f) Relative error between (d) and (e) (%).

(g) Melt pool at χ_3 with FEM (°C). (h) Melt pool at χ_3 with $\mathcal{F}_2$ (°C). (i) Relative error between (g) and (h) (%).

(j) Melt pool at χ_4 with FEM (°C). (k) Melt pool at χ_4 with $\mathcal{F}_2$ (°C). (l) Relative error between (j) and (k) (%).

Figure 7.16: The melt pool comparison between FEM and $\mathcal{F}_2$ in the test $\mathcal{S}_8$.

pool sizes are listed in table 7.16. In $\mathcal{S}_8$, the maximum temperatures at χ_i for $i = 1, \cdots, 4$ are estimated as 1504.46°C, 1507.15°C, 1520.17°C, and 1537.94°C with relative errors 0.86%, 1.14%, 1.34%, and 0.66%, and the maximum relative errors of temperatures around melt pools are respectively 3.27%, 2.53%, 2.22%, and 3.45%. While the width of χ_1 and χ_3 are precisely approximated,



(a) Melt pool at χ_1 with FEM (°C). (b) Melt pool at χ_1 with $\mathcal{F}_2$ (°C). (c) Relative error between (a) and (b) (%).

(d) Melt pool at χ_2 with FEM (°C). (e) Melt pool at χ_2 with $\mathcal{F}_2$ (°C). (f) Relative error between (d) and (e) (%).

(g) Melt pool at χ_3 with FEM (°C). (h) Melt pool at χ_3 with $\mathcal{F}_2$ (°C). (i) Relative error between (g) and (h) (%).

(j) Melt pool at χ_4 with FEM (°C). (k) Melt pool at χ_4 with $\mathcal{F}_2$ (°C). (l) Relative error between (j) and (k) (%).

Figure 7.17: The melt pool comparison between FEM and $\mathcal{F}_2$ in the test $\mathcal{S}_9$.

the maximum of relative errors in melt pool size approximation is 5.70%. In $\mathcal{S}_9$, the maximum temperatures at χ_i for $i = 1, \cdots, 4$ are estimated as 1522.54°C,

1281.55°C, 1283.88°C, and 1318.68°C with relative errors 2.35%, 1.76%, 3.13%, and 0.97%. The maximum of relative errors around melt pools are respectively 9.54%, 6.65%, 8.34%, and 4.82%. The length of χ_4 is precisely approximated, and the maximum of relative errors in melt pool size approximation is 8.92%. As

Table 7.16: The comparison of melt pool sizes between FEM and $\mathcal{F}_2$ in the test $\mathcal{S}_8$ and $\mathcal{S}_9$.

Test	Position	Melt pool size	FEM (μm)	$\mathcal{F}_2$ (μm)	Relative error (%)
$\mathcal{S}_8$	χ_1	Length	148.20	146.20	1.35
		Width	94.19	94.19	< 0.01
		Depth	59.59	58.21	2.31
	χ_2	Length	135.00	132.50	1.85
		Width	91.18	91.31	0.15
		Depth	56.43	56.16	0.47
	χ_3	Length	179.51	176.02	1.95
		Width	103.63	103.63	< 0.01
		Depth	69.45	70.47	1.48
	χ_4	Length	188.27	177.53	5.70
		Width	105.09	104.26	0.79
		Depth	73.62	73.87	0.34
$\mathcal{S}_9$	χ_1	Length	130.00	127.50	1.92
		Width	96.69	99.99	3.42
		Depth	55.89	60.00	7.35
	χ_2	Length	122.87	122.96	0.08
		Width	86.98	91.18	4.83
		Depth	55.80	56.43	1.12
	χ_3	Length	143.24	148.74	3.84
		Width	118.77	111.21	6.36
		Depth	82.16	77.60	5.55
	χ_4	Length	147.50	147.50	< 0.01
		Width	94.81	103.26	8.92
		Depth	66.59	68.51	2.88

shown in figure 7.18, the four thermal histories in $\mathcal{S}_8$ are finely recovered, which merely have relative errors 0.65%, 0.98%, 0.98%, and 1.01%. In the test $\mathcal{S}_9$ with random λ, however, the corresponding relative errors are 3.07%, 3.05%, 3.83%, and 3.34%. It is less accurate than the test $\mathcal{S}_8$, but the approximated lines from $\mathcal{F}_2$ in figure 7.19 show that the randomness in thermal histories can be partly predicted as in $\mathcal{F}_2$ we still run the thermal simulator. Though it is projected and

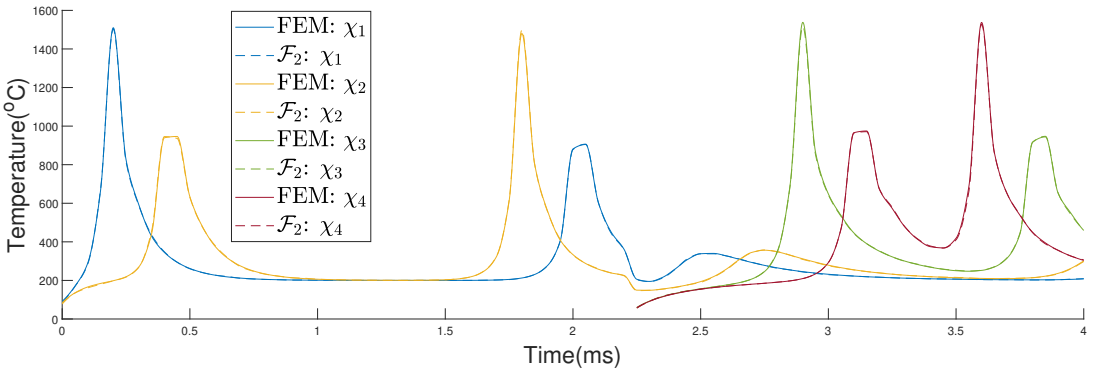


Figure 7.18: The thermal history comparison between FEM and $\mathcal{F}_2$ in $\mathcal{S}_8$.

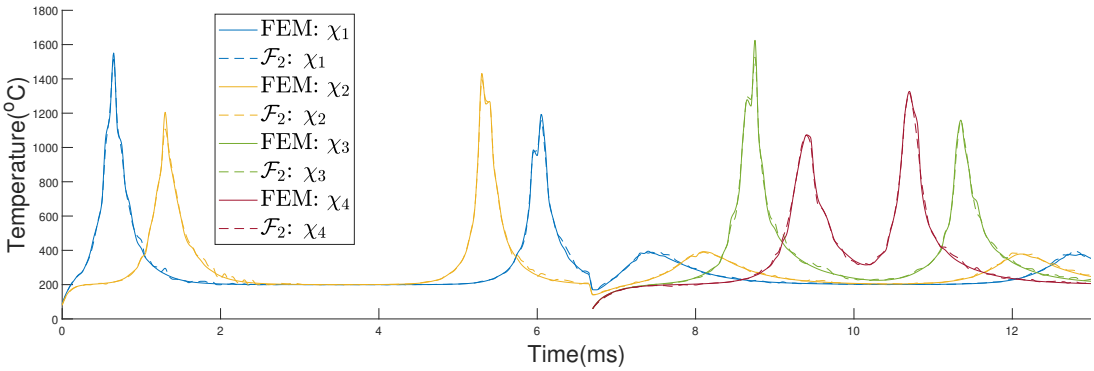


Figure 7.19: The thermal history comparison between FEM and $\mathcal{F}_2$ in $\mathcal{S}_9$.

sketched, the information of anisotropy scale factor λ is used at each time step.

7.3.3 Offline preparation

Since $\mathcal{F}_2$ also needs the data-driven relative distance predictor in $\mathcal{F}_1$, its offline preparation also includes data generation, nonlinear dimension reduction, and training as the description in section 7.2.4. Once the relative distance predictor is established, there is no other preparation required before the simulations start.

7.3.4 Time cost reduction

Since the accuracy compromise of $\mathcal{F}_2$ is comparably small, it is expected that $\mathcal{F}_2$ needs more running time. While the offline preparation time is the same as $\mathcal{F}_1$, the total online execution time, listed as 22.41s and 20.28s on average for the one- to two-layer domain in table 7.17, is composed of four parts: basis generation, projection, sketching, and simulation. It is noted that in the i-layer domain for $i = 1, 2$ the average execution time of basis generation (0.23s and

0.26s) and projection (2.59s and 2.76s) is much less than sketching (8.04s and 7.86s) and simulation (11.55s, and 9.40s). The first three parts: basis generation, projection, and sketching, however, only needs to be computed once with one set of test inputs while the simulation part includes the Picard iteration taking several runs to converge. The online execution time is still observed with a significant reduction respectively saving 95.13% and 96.61% of time cost in the one- to two-layer domain. The promising advantage of $\mathcal{F}_2$ is that it expedites the high-fidelity simulator with a small compromise of accuracy. $\mathcal{F}_2$ manages to make most predictions with relative error less than 1% using no more than 5% of online running time required by FEM, thus its performance is encouraging for applications that require accurate thermal simulations but do not excessively demand real-time implementations.

Table 7.17: The time cost reduction of $\mathcal{F}_2$.

Layer	Average time cost of $\mathcal{F}_2$ (s)					Reduction (%)
	Basis generation	Projection	Sketching	Simulation	Total	
1	0.23	2.59	8.04	11.55	22.41	95.13
2	0.26	2.76	7.86	9.40	20.28	96.61

7.4 Sketched surrogate with online local projection

7.4.1 A simplified example

A simplified example is provided to simulate the heat transfer with a Gaussian heat source moving forward with a speed v, the codes of which are provided in the GitHub repository [21]. The Gaussian function follows equation (3.3) with a fixed amplitude denoted as f_s in this section. The three-dimensional domain is discretized by mesh with less refinement, 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 parameters of the thermal model and the surrogate are respectively listed in table 7.18 and table 7.19.

Table 7.18: Thermal model parameters in the simplified example.

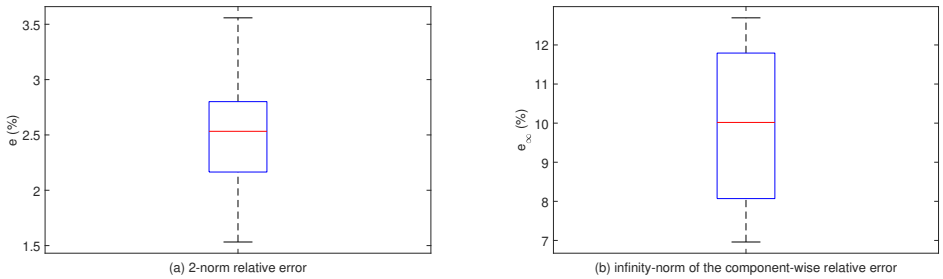
Symbol	Definition (Unit)	Value
n_d	the number of degrees of freedom	2986
$\bar{n}_d$	the number of nodes on Γ_b	561
n_e	the number of elements	17453
-	the size of domain (mm)	$0.50 \times 0.17 \times 0.10$
-	the side length of refined tetrahedron mesh (mm)	0.01
-	the side length of coarse tetrahedron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

Table 7.19: The surrogate model parameters in the simplified example.

Symbol	Definition	Value
η	the scale factor of Gaussian function variance	$[0.5, 1, 1.5]^T$
u_m	the temperature threshold of node selection	30°C
n_r	the dimension of projection	40
ι	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	50μm

7.4.1.1 Model accuracy

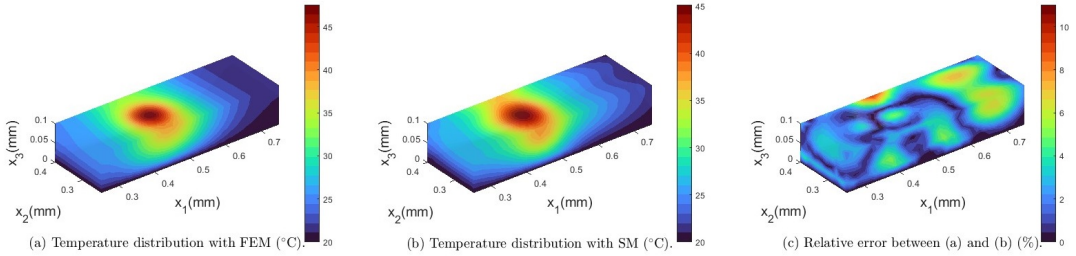
The model accuracy is evaluated by relative errors which are computed in 2-norm and infinity-norm. The boxplots as figure 7.20 show the two types of relative errors for the simplified example in this section. We can tell that 2-norm relative



(a) 2-norm relative error (b) infinity-norm of the component-wise relative error

Figure 7.20: Relative errors between FEM and SM.

errors range from 1.53% to 3.56% while infinity-norm relative errors range from 6.96% to 12.69%. More specifically, we can compare the temperature distributions between FEM and the surrogate $\mathcal{F}_3$. Take one time step ($t = 0.4$ms) as an example, the temperature profiles and the relative errors are shown in figure 7.21.



(a) Temperature distribution with FEM (°C). (b) Temperature distribution with SM (°C). (c) Relative error between (a) and (b) (%).

Figure 7.21: The comparison of temperature distribution between FEM and SM.

7.4.1.2 Time cost reduction

The average time cost of FEM and $\mathcal{F}_3$ are summarized in table 7.20 where the time cost of $\mathcal{F}_3$ 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.

7.4.2 Parameter setup

To compare the performance of $\mathcal{F}_3$ with the previous two surrogates $\mathcal{F}_1$ and $\mathcal{F}_2$, we use the same 16 tests listed in table 7.7. The online local projection basis is

Table 7.20: The comparison of average time cost for the simplified example.

FEM (s)	$\mathcal{F}_3$ (s)					Reduction (%)
	Basis generation	Sketching	Projection	Simulation	Total	
1.3414	0.0066	0.2754	0.0077	0.6004	0.8901	33.64

applied and updated at each time step, which means it is swiftly generated in the process of simulation without the help of offline preparation. The parameters of $\mathcal{F}_3$ are specified in table 7.21, which include the parameters to generate Gaussian local projection bases, approximate the sampling probability, and implement the row selection in randomized sketching.

Table 7.21: Parameters of $\mathcal{F}_3$.

Symbol	Definition	Value
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
n_r	the dimension of projection	200
ι	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	35μm

7.4.3 Model accuracy and robustness

The spread of 2-norm and infinity-norm relative errors of all 16 tests are shown as the boxplots in figure 7.22. We can tell that most of the 2-norm relative errors are smaller than 3%. More specifically, there are respectively 3.99%, 52.22%, and 99.47% of 2-norm relative errors below 1.5%, 2%, and 3%, while the maximum 2-norm relative error among all tests is 3.35%. The infinity norm relative errors, however, ranges from 5.34% to 21.50% where 69.28%, 98.76%, and 99.91% of tests are respectively below 10%, 15%, and 20%. The robustness of $\mathcal{F}_3$ in dealing with some anisotropy randomness is shown in table 7.22. It indicates that though using a random λ results in higher error both in terms of the maximum

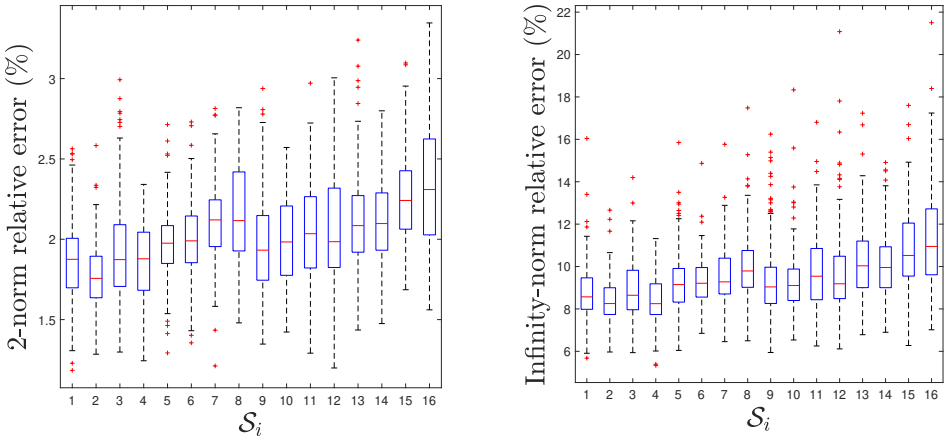
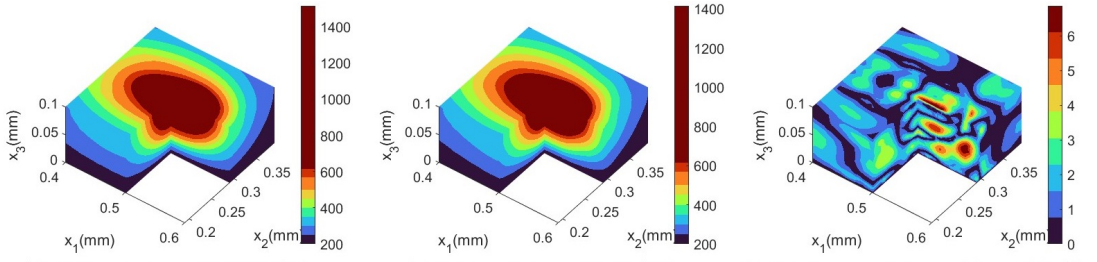


Figure 7.22: The boxplot of relative errors between FEM and $\mathcal{F}_3$.

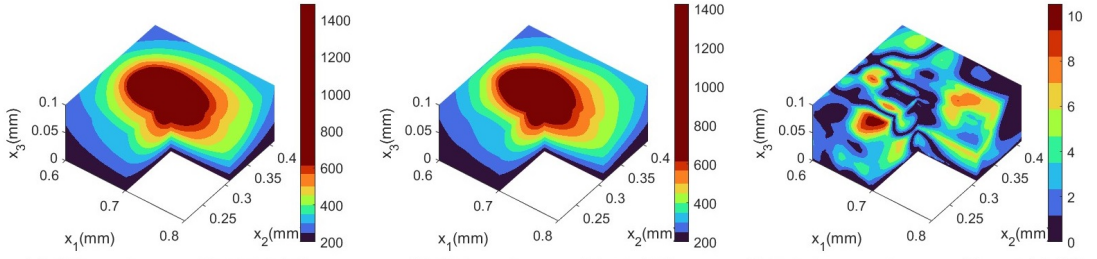
Table 7.22: The accuracy comparison of $\mathcal{F}_3$ between using a static or random λ.

λ	2-norm relative error (%)				Infinity-norm relative error (%)			
	Max	$< 1.5\%$	$< 2\%$	$< 3\%$	Max	$< 10\%$	$< 15\%$	$< 20\%$
Static	2.99	5.32	61.61	100.00	17.48	79.79	99.56	100.00
Random	3.35	2.66	42.82	98.94	21.50	58.78	97.96	99.82

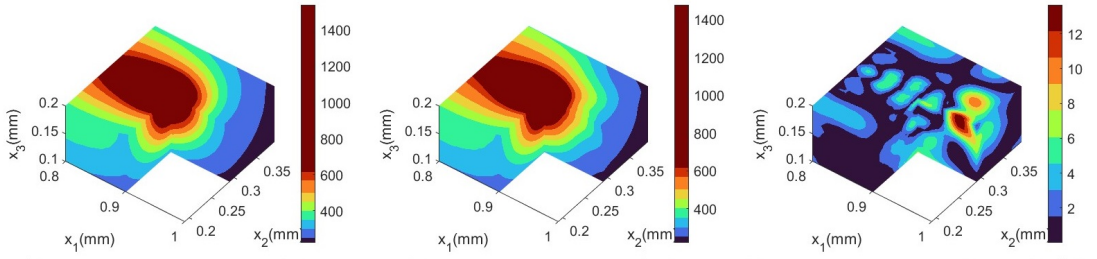
and the spread of relative errors $\mathcal{F}_3$ outperforms in handling the randomness of λ than $\mathcal{F}_1$ and $\mathcal{F}_2$. While the increase of the maximum relative error is comparably small (an increase of 0.36% for 2-norm and 4.02% for infinity-norm relative errors), the percentage of tests with $e < 2\%$ and $e_\infty < 15\%$ respectively decrease from 61.61% to 42.82% and from 99.56% to 97.96%. It is a significantly smaller drop compared with $\mathcal{F}_1$ and $\mathcal{F}_2$. In addition to the relative errors of high-dimensional temperatures, we further focus on the area around melt pools where most temperature gradients concentrate. Around the selected positions χ_i for $i = 1, \cdots, 4$ in $\mathcal{S}_8$, the maximum temperatures with FEM are respectively 1517.59°C, 1490.11°C, 1540.79°C, and 1548.21°C while with $\mathcal{F}_3$ they are correspondingly estimated as 1416.10°C, 1424.75°C, 1480.37°C, and 1386.82°C with relative errors 6.69%, 4.39%, 3.92%, and 10.42%. In $\mathcal{S}_9$, the maximum temperatures at χ_i for $i = 1, \cdots, 4$ are 1559.23°C, 1259.34°C, 1244.87°C, and 1331.57°C with FEM. With $\mathcal{F}_3$, they are 1477.90°C, 1201.75°C, 1175.61°C, and 1266.21°C with relative errors 5.22%, 4.57%, 5.56%, and 4.91%. Figure 7.23 and 7.24 respectively compares the temperatures around the positions χ_i for $i = 1, \cdots, 4$ in the test $\mathcal{S}_8$ and $\mathcal{S}_9$. The maximum relative errors around χ_i for $i = 1, \cdots, 4$ in $\mathcal{S}_8$ are



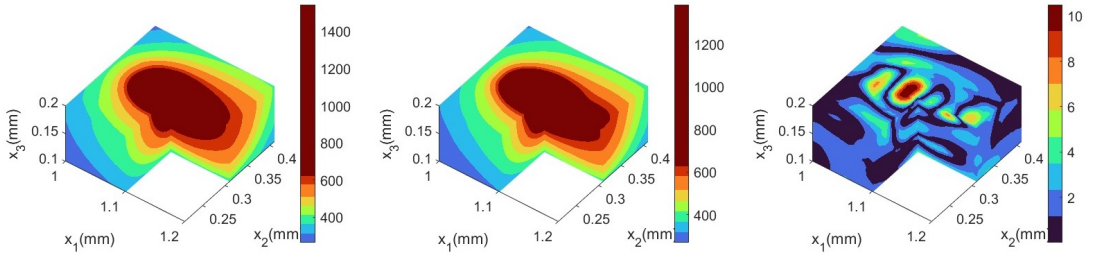
(a) Melt pool at χ_1 with FEM (°C). (b) Melt pool at χ_1 with $\mathcal{F}_3$ (°C). (c) Relative error between (a) and (b) (%).



(d) Melt pool at χ_2 with FEM (°C). (e) Melt pool at χ_2 with $\mathcal{F}_3$ (°C). (f) Relative error between (d) and (e) (%).

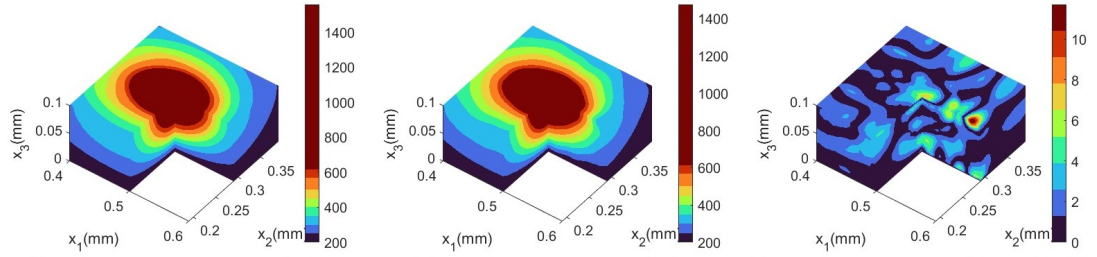


(g) Melt pool at χ_3 with FEM (°C). (h) Melt pool at χ_3 with $\mathcal{F}_3$ (°C). (i) Relative error between (g) and (h) (%).

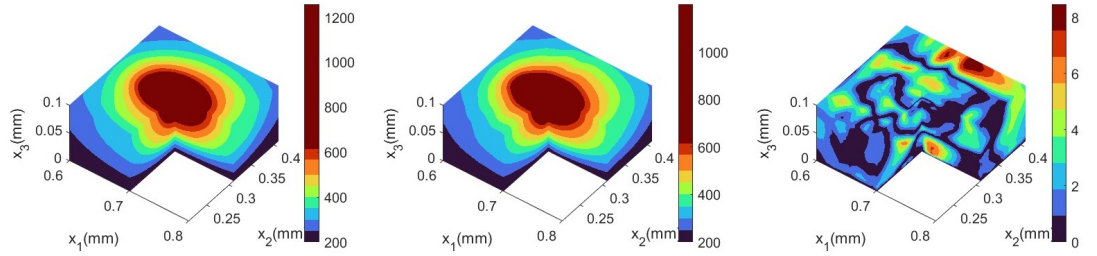


(j) Melt pool at χ_4 with FEM (°C). (k) Melt pool at χ_4 with $\mathcal{F}_3$ (°C). (l) Relative error between (j) and (k) (%).

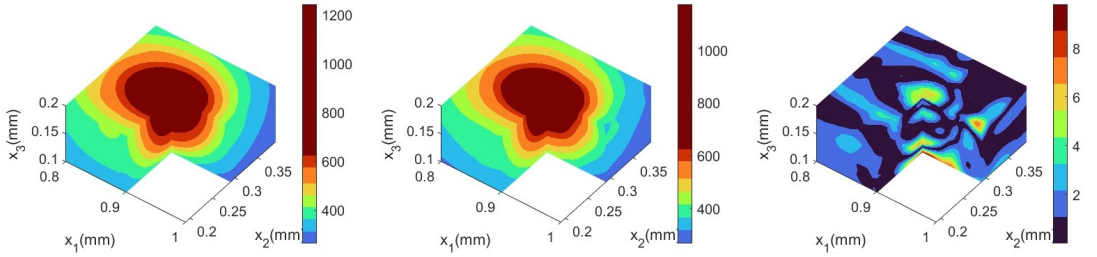
Figure 7.23: The melt pool comparison between FEM and $\mathcal{F}_3$ in the test $\mathcal{S}_8$.



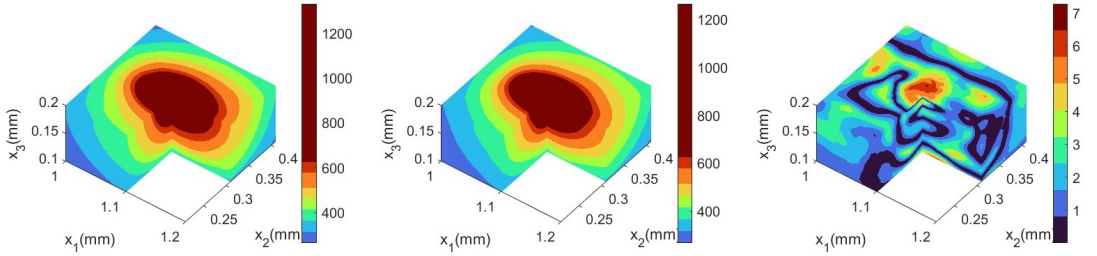
(a) Melt pool at χ_1 with FEM (°C). (b) Melt pool at χ_1 with $\mathcal{F}_3$ (°C). (c) Relative error between (a) and (b) (%).



(d) Melt pool at χ_2 with FEM (°C). (e) Melt pool at χ_2 with $\mathcal{F}_3$ (°C). (f) Relative error between (d) and (e) (%).



(g) Melt pool at χ_3 with FEM (°C). (h) Melt pool at χ_3 with $\mathcal{F}_3$ (°C). (i) Relative error between (g) and (h) (%).



(j) Melt pool at χ_4 with FEM (°C). (k) Melt pool at χ_4 with $\mathcal{F}_3$ (°C). (l) Relative error between (j) and (k) (%).

Figure 7.24: The melt pool comparison between FEM and $\mathcal{F}_3$ in the test $\mathcal{S}_9$.

respectively 6.86%, 10.52%, 13.66%, and 10.50%, while in $\mathcal{S}_9$ the correspondingly maximum relative errors are 11.71%, 8.46%, 9.79%, and 7.27%. The comparison of melt pool size with FEM and $\mathcal{F}_3$ is listed in table 7.23. The maximum relative

Table 7.23: The comparison of melt pool sizes between FEM and $\mathcal{F}_3$ in the test $\mathcal{S}_8$ and $\mathcal{S}_9$.

Test	Position	Melt pool size	FEM (μm)	$\mathcal{F}_3$ (μm)	Relative error (%)
$\mathcal{S}_8$	χ_1	Length	148.20	147.50	0.47
		Width	94.19	94.19	< 0.01
		Depth	59.59	57.57	3.39
	χ_2	Length	135.00	130.70	3.19
		Width	91.18	93.11	2.12
		Depth	56.43	53.26	5.61
	χ_3	Length	179.51	177.92	0.89
		Width	103.63	103.63	< 0.01
		Depth	69.45	63.73	8.24
	χ_4	Length	188.27	197.50	4.90
		Width	105.09	102.99	2.00
		Depth	73.62	69.47	5.63
$\mathcal{S}_9$	χ_1	Length	130.00	130.00	< 0.01
		Width	96.69	96.17	0.54
		Depth	55.89	58.21	4.16
	χ_2	Length	122.87	120.41	2.00
		Width	86.98	87.99	1.16
		Depth	55.80	52.93	5.16
	χ_3	Length	143.24	142.85	0.27
		Width	118.77	111.80	5.87
		Depth	82.16	73.70	10.29
	χ_4	Length	147.50	147.50	< 0.01
		Width	94.81	93.81	1.05
		Depth	66.59	64.70	2.84

errors of melt pool sizes are respectively 8.24% and 10.29% in $\mathcal{S}_8$ and $\mathcal{S}_9$, while the width of χ_1 and χ_3 in $\mathcal{S}_8$, the length of χ_1 and χ_4 in $\mathcal{S}_9$ are precisely approximated. At the same four positions, their thermal histories are compared as figure 7.25 and 7.26. In $\mathcal{S}_8$, the relative errors of thermal histories at χ_i for $i = 1, \cdots, 4$ are respectively 1.88%, 3.30%, 3.09%, and 5.36%. For the thermal histories in $\mathcal{S}_9$, we can see the effect of anisotropy randomness while the relative errors at the four selected positions are respectively 3.01%, 2.88%, 3.37%, and 3.30%.

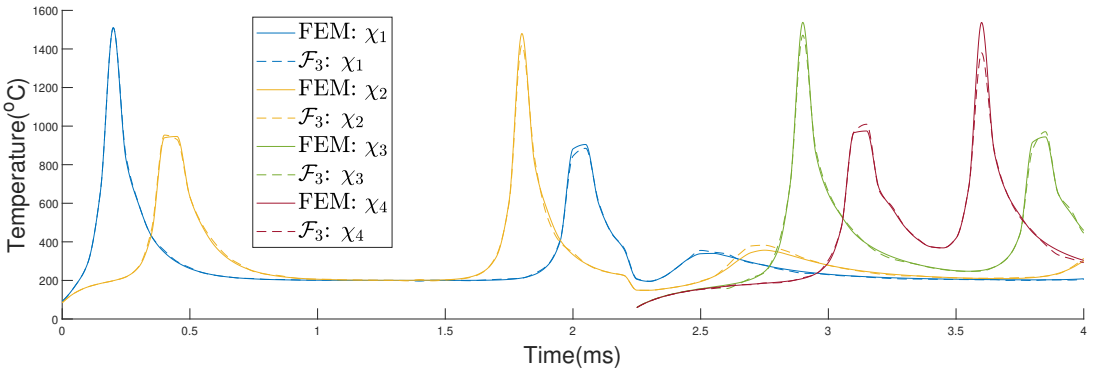


Figure 7.25: The thermal history comparison between FEM and $\mathcal{F}_3$ in $\mathcal{S}_8$.

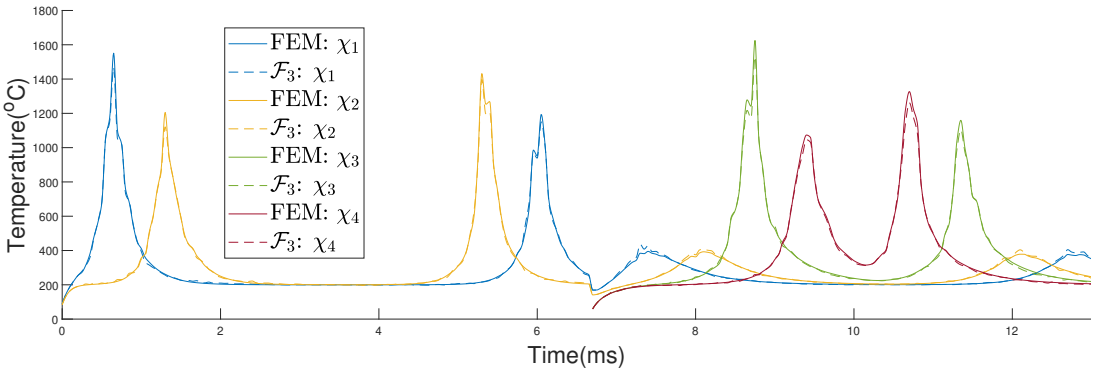


Figure 7.26: The thermal history comparison between FEM and $\mathcal{F}_3$ in $\mathcal{S}_9$.

7.4.4 Offline preparation

Different from $\mathcal{F}_1$ and $\mathcal{F}_2$, there is no offline preparation required for $\mathcal{F}_3$. It is an advantage of $\mathcal{F}_3$ as the data generation and training can be costly in time and storage.

7.4.5 Time cost reduction

While the accuracy of $\mathcal{F}_3$ is validated in section 7.4.3, another important performance is the reduction of execution time. The online execution of $\mathcal{F}_3$, however, consists of 4 parts: basis generation, sketching, projection, and simulation. While the basis generation, sketching, and projection respectively corresponding to section 6.2.1, 6.1, and 6.2.2 are successively executed, 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 7.24, $\mathcal{F}_3$ manages to respectively save 87.85% and 90.39% of

time cost for the one- and two-layer printing. Specifically, while the average execution time of FEM is respectively 459.90s for layer 1 and 597.72s for layer 2, the average execution time of $\mathcal{F}_3$ is correspondingly 55.90s and 57.47s. Most of the execution time in $\mathcal{F}_3$ 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 6 will reduce the sketching time required, and a smaller multiplier ι will cut down the simulation time required.

Table 7.24: The time cost reduction of $\mathcal{F}_3$.

Layer	Average time cost of $\mathcal{F}_3$ (s)					Reduction (%)
	Basis generation	Sketching	Projection	Simulation	Total	
1	1.56	22.95	1.99	29.40	55.90	87.85
2	1.90	24.35	2.34	28.88	57.47	90.39

7.5 Comparison and discussion

Three time-efficient surrogates, namely the reduced GP surrogate $\mathcal{F}_1$, the sketched surrogate with data-driven local projection $\mathcal{F}_2$, and the sketched surrogate with online local projection $\mathcal{F}_3$, have been introduced as rapid computational alternatives to thermal modelling of LPBF with FEM. These three surrogates effectively estimate most tests with high accuracy and offer substantial time savings during online execution. However, it is important to note that each surrogate possesses its own distinct advantages and disadvantages, rendering them suitable for various application scenarios.

7.5.1 Model accuracy

Model accuracy, a crucial performance metric, is assessed using the 2-norm and infinity-norm relative error measurements. We evaluate and compare the three surrogates $\mathcal{F}_i$ $(i = 1, 2, 3)$ in two distinct sets of tests to respectively assess model accuracy and robustness. The first set comprises the initial 8 tests ($\mathcal{S}_i$ for $i =$

$1, \cdots, 8$), where a static anisotropy scale factor is employed. In contrast, the second set consists of the remaining 8 tests ($\mathcal{S}_i$ for $i = 9, \cdots, 16$), which adopt a random anisotropy scale factor. In this section, the first set is utilized to evaluate model accuracy, while the second set will be addressed later in section 7.5.2 to evaluate model robustness. The comparison of the 2-norm and infinity-norm relative errors, denoted as e and e_∞ respectively, is presented in table 7.25. Notably, these relative errors are computed across all time steps, except for the initial five time steps of each scanning line, as the initial stages exhibit irregularities in the melt pools.

Table 7.25: The comparison of model accuracy and robustness.

Type	Quartile	Static			Random		
		$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$
e (%)	Q_1	1.35	0.41	1.75	2.69	1.65	1.85
	Q_2	1.95	0.55	1.93	3.60	2.16	2.06
	Q_3	2.48	0.72	2.10	4.95	3.20	2.28
	Max	9.26	1.72	2.99	10.21	6.38	3.35
e_∞ (%)	Q_1	5.08	1.96	8.11	11.84	7.31	8.67
	Q_2	7.51	2.40	8.91	16.46	9.85	9.65
	Q_3	10.69	2.91	9.81	21.29	13.37	10.93
	Max	56.97	6.19	17.48	75.33	27.12	21.50

In terms of model accuracy, $\mathcal{F}_2$ demonstrates the highest level of accuracy. For 25%, 50%, and 75% of the tests, the 2-norm relative errors are below 0.41%, 0.55%, and 0.72% respectively. The maximum 2-norm relative error among all the tests is only 1.72%, and the infinity-norm relative errors for all tests remain below 6.19%. Following closely, the second most accurate surrogate is $\mathcal{F}_3$. The maximum 2-norm and infinity-norm relative errors for this surrogate are 2.99% and 17.48% respectively, indicating that all tests are estimated with satisfactory accuracy. On the other hand, $\mathcal{F}_1$ is the least accurate among the three surrogates. Although the Q_1, Q_2, and Q_3 values of $\mathcal{F}_1$ are similar to those of $\mathcal{F}_3$, the maximum 2-norm and infinity-norm relative errors for $\mathcal{F}_1$ are 9.26% and 56.97% respectively, suggesting the presence of inaccurately estimated outliers.

7.5.2 Robustness

The three surrogates exhibit varying degrees of robustness when it comes to handling the randomness of the anisotropy scale factor λ. Among them, $\mathcal{F}_3$ is the most robust one. With a random λ, there is a slight increase in Q_1, Q_2, Q_3, and the maximum of 2-norm relative errors respectively by 0.10% (from 1.75% to 1.85%), 0.13% (from 1.93% to 2.06%), 0.18% (from 2.10% to 2.28%), and 0.36% (from 2.99% to 3.35%) for $\mathcal{F}_3$. In terms of the infinity-norm relative errors, the increase in Q_1, Q_2, Q_3, and the maximum are respectively 0.56% (from 8.11% to 8.67%), 0.74% (from 8.91% to 9.65%), 1.12% (from 9.81% to 10.93%), and 4.02% (from 17.48% to 21.50%). Both $\mathcal{F}_1$ and $\mathcal{F}_2$ exhibit less robustness when handling random λ. However, the decrease in model accuracy caused by introducing randomness in λ is slightly more noticeable for $\mathcal{F}_2$ since it performs quite accurately in tests with static λ. Specifically, for $\mathcal{F}_1$, the increase in Q_1, Q_2, Q_3, and the maximum of 2-norm relative errors is 1.34%, 1.65%, 2.47%, and 0.95% respectively, while for $\mathcal{F}_2$, the corresponding increase is 1.24%, 1.61%, 2.48%, and 4.66%. The increase in Q_1, Q_2, Q_3, and the maximum of infinity-norm relative errors is 6.76%, 8.95%, 10.60%, and 18.36% respectively for $\mathcal{F}_1$, whereas for $\mathcal{F}_2$, these values are 5.35%, 7.45%, 10.46%, and 20.93% respectively.

7.5.3 Time cost and offline preparation

The time cost of the three surrogates can be divided into two categories: offline preparation time and online execution time. Offline preparation is necessary for constructing the data-driven relative distance predictor used in both $\mathcal{F}_1$ and $\mathcal{F}_2$, resulting in the same offline preparation time for these two surrogates. However, $\mathcal{F}_3$ does not require any offline preparation time. The details of offline preparation time, including data generation (7.30 days for layer 1 and 9.28 days for layer 2), nonlinear dimension reduction (55.10s for layer 1 and 53.49s for layer 2), and training (1.10hr for layer 1 and 0.85hr for layer 2), are presented in table 7.12.

The reduction in online execution time is compared in table 7.26. Among the three surrogates, $\mathcal{F}_1$ is the fastest, offering near real-time implementations (0.12s for layer 1 and 0.11s for layer 2). $\mathcal{F}_2$ is the second fastest surrogate,

with a reduction in execution time of 95.13% for layer 1 and 96.61% for layer 2. Although $\mathcal{F}_3$ is the slowest among the three, it still manages to significantly reduce the online execution time (87.85% for layer 1 and 90.39% for layer 2).

Table 7.26: The comparison of the online execution time.

Layer	Online execution time (s)				Reduction (%)		
	FEM	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$
1	459.90	0.12	22.41	55.90	99.97	95.13	87.85
2	597.72	0.11	20.28	57.47	99.98	96.61	90.39

7.5.4 Different strengths of each surrogate

The three surrogates exhibit superior performance in various aspects. $\mathcal{F}_1$ excels in reducing time costs, enabling near real-time implementation. However, it is less accurate and robust compared to the other surrogates. While $\mathcal{F}_1$ relies on finding a representative training dataset, limiting its accuracy and robustness, a similar reliance exists in $\mathcal{F}_2$ as they share the same relative distance predictor. However, $\mathcal{F}_2$ overcomes this limitation to some extent. It constructs data-driven local projection bases that do not heavily depend on precise relative distance predictions. Instead, all subsampled temperature snapshots equally contribute to the projection bases. Conversely, $\mathcal{F}_1$ requires accurate and ordered relative distance predictions to compute weights for high-dimensional temperature extrapolation. The strength of $\mathcal{F}_2$ lies in its model accuracy, leveraging information from both precomputed temperature data and the numerical solver with FEM. Although it takes more time than $\mathcal{F}_1$ and lacks robustness to random anisotropy, it offers improved accuracy. $\mathcal{F}_3$ outperforms in terms of robustness to random anisotropy and requires no offline preparation. However, it takes more time compared to the other two surrogates and is less accurate than $\mathcal{F}_2$. Its performance is influenced by the method used to generate online local projection bases. Since the generation of projection bases does not rely on a pre-trained data-driven model, no offline preparation is needed. By employing hundreds of calibrated and adjusted Gaussian functions to delineate temperature distributions, the projection bases of $\mathcal{F}_3$ can cover a wider range of temperature distributions, enhancing its robustness

in handling anisotropy randomness. However, this also increases the dimension of the projected model to hundreds comparing with the tens in $\mathcal{F}_2$, resulting in longer online execution time.

The analysis above reveals that all three surrogates demonstrate strong performance in reducing time costs while maintaining accuracy. Despite being the least accurate and robust, $\mathcal{F}_1$ still achieves a commendable result by suppressing 85.30% of all tests with 2-norm relative errors below 5%. $\mathcal{F}_2$, although slower and less robust, manages to save over 95% of the time cost and reduces the 2-norm relative errors of 95.75% of tests with random λ to below 5%. Similarly, the slower and less accurate $\mathcal{F}_3$ still achieves a remarkable outcome by saving more than 87% of execution time and yielding relative errors lower than 3% for 99.47% of the tests.

The selection of $\mathcal{F}_1$, $\mathcal{F}_2$, or $\mathcal{F}_3$ depends on the specific application requirements. If a general description of temperature profiles suffices, $\mathcal{F}_1$ is a more efficient choice. For situations where precise analysis is essential and an execution time of approximately 20 seconds is acceptable, $\mathcal{F}_2$ is preferable. If considering anisotropy randomness and/or minimizing offline preparation is crucial, $\mathcal{F}_3$ is the more suitable option, even though it entails an execution time of around 50 seconds, which is tolerable.

Chapter 8

Conclusion

Together with a nonlinear thermal model of LPBF numerically solved by FEM, three time-efficient surrogates: the reduced GP surrogate $\mathcal{F}_1$, the sketched surrogate with data-driven local projection $\mathcal{F}_2$, and the sketched surrogate with online local projection $\mathcal{F}_3$ are proposed as swift alternatives with different trade-offs in terms of model accuracy, robustness, offline preparation, and time cost reduction. The methods are published in [19] or [20], while the results of the three surrogates are compared under the same parameter settings and are thereby different from the two aforementioned papers. A simplified example of $\mathcal{F}_3$ published in [20] is presented, the codes of which are provided in the GitHub repository [21].

The thermal model of LPBF is governed by a nonlinear and anisotropic heat equation, the computational domain of which is modelled as a cuboid vertically extended layer after layer. The Dirichlet boundary condition is employed to simulate temperature control on the building platform, imposing a constant temperature on the bottom surface. The Neumann boundary condition is employed to account for Gaussian heat flux, convection, and radiation heat loss. The thermal properties of the metal material are temperature-dependent, additionally considering phase changes, anisotropy, and latent heat. To simulate temperature distributions under various printing parameters (laser power and scan speed), as well as static/random anisotropy, a full-order numerical solver based on FEM is employed. This solver captures melt pool sizes and thermal histories at any desired location. The powder bed domain is discretized using tetrahedrons, with

conforming refined mesh elements around the printing area. Gaussian quadrature approximation and Picard iterations are respectively utilized for integral estimation and solving the nonlinear heat equation. The full-order model exhibits high computational complexity due to the fine spatial discretization and intricate nonlinearity resulting from temperature-dependent thermal properties and boundary conditions. The increased number of elements resulting from high dimensionality, as well as the higher number of integration points required for Gaussian quadrature approximation, contribute to the demands of nonlinear computations in each Picard iteration, significantly impacting computation time. On average, the full-order model with FEM requires 528.81s per time step.

Three time-efficient surrogates with different advantages and disadvantages are proposed. To evaluate their performance, a series of numerical experiments are conducted using a two-layer printing process involving two straight lines scanned back and forth on an AlSi10Mg powder bed under an argon atmosphere. The reduced GP surrogate $\mathcal{F}_1$ is a data-driven model that approximates high-dimensional temperatures through linear combinations of subsampled training temperatures closest to the final prediction. The subsampling process relies on a relative distance predictor comprising nonlinear dimension reduction and Gaussian process regression with three control inputs: laser power, scan speed, and time. Once the relative distance predictor is well-trained, it enables almost real-time high-dimensional temperature prediction. However, $\mathcal{F}_1$ exhibits lower accuracy compared to the other surrogates. In this chapter, the first five time steps of each scanning line are excluded to assess model accuracy as there is not sufficient heat to melt materials and form regular melt pools at the beginning. For static anisotropy, 52.30% and 96.90% of tests for $\mathcal{F}_1$ respectively achieve 2-norm relative errors below 2% and 5%. However, $\mathcal{F}_1$ is weak in handling anisotropy randomness, with only 8.87% and 75.53% of tests with random anisotropy achieving 2-norm relative errors below 2% and 5% respectively. Despite its limitations, $\mathcal{F}_1$ is the only surrogate suitable for real-time implementation, with an average prediction time of 0.12s. Offline preparation is required for $\mathcal{F}_1$, including data generation (16.58 days), nonlinear dimension reduction (108.59s), and training (1.97hr). The sketched surrogate with data-driven local projection $\mathcal{F}_2$ accelerates

the full-order model by subspace projection using data-driven local projection bases. These bases are constructed by subsampling training temperatures based on the relative distance predictor in $\mathcal{F}_1$. However, unlike $\mathcal{F}_1$, $\mathcal{F}_2$ is less reliant on finding representative training data due to its tolerance for prediction errors in relative distances. After projection, the randomized sketching technique is applied to subsample the algebraic operations involved in solving the nonlinear model with Picard iterations. $\mathcal{F}_2$ exhibits the highest accuracy among the three surrogates, with 93.53% of tests achieving 2-norm relative errors below 1% for static anisotropy. However, similar to $\mathcal{F}_1$, $\mathcal{F}_2$ is not robust in handling anisotropy randomness. The percentage of tests with 2-norm relative errors below 1% and 2% respectively decreases from 93.53% to 3.10% and from 100.00% to 42.73% when random anisotropy is introduced. The offline preparation for $\mathcal{F}_2$ is the same as $\mathcal{F}_1$, as both surrogates utilize the same relative distance predictor. In terms of online execution time, $\mathcal{F}_2$ ranks second among the three surrogates, taking an average of 21.35s, which represents a time cost reduction of over 95%. The sketched surrogate with online local projection $\mathcal{F}_3$ also employs local projection and randomized sketching techniques, albeit with different implementations compared to $\mathcal{F}_2$. The local projection bases of $\mathcal{F}_3$ are generated online, eliminating the need for offline preparation. These bases consist of several previous temperature snapshots and three-dimensional Gaussian functions derived from a benchmark Gaussian function calibrated by the most recent temperature distribution. The randomized sketching process in $\mathcal{F}_3$ exploits the approximated sampling probability to further reduce sketching costs and avoid full projection, both are time-consuming steps given the larger reduced dimension compared to $\mathcal{F}_2$. $\mathcal{F}_3$ achieves higher accuracy than $\mathcal{F}_1$ but lower accuracy than $\mathcal{F}_2$. For static anisotropy, the maximum 2-norm relative error is 2.99%, with 61.61% of tests exhibiting relative errors below 2%. However, $\mathcal{F}_3$ demonstrates greater robustness, as for tests with random anisotropy, the maximum relative error is 3.35%, while 42.82% and 98.94% of tests respectively achieve relative errors below 2% and 3%. $\mathcal{F}_3$ has the longest execution time among the three surrogates, taking 56.69s on average. In addition to temperature profiles, the estimation of thermal histories and melt pool sizes at selected positions are also considered. $\mathcal{F}_1$ exhibits

the least accuracy in estimating melt pool size and thermal history, while $\mathcal{F}_2$ is the most accurate, and $\mathcal{F}_3$ demonstrates the highest robustness. It is noteworthy that only $\mathcal{F}_2$ and $\mathcal{F}_3$ are able to capture the effects of random anisotropy in thermal histories.

All three surrogates demonstrate accurate estimation capabilities and significant reductions in online execution time. However, each surrogate strikes a balance between accuracy, robustness, offline preparation, and online execution time in its own way. $\mathcal{F}_1$ prioritizes real-time implementation but compromises offline preparation, accuracy, and robustness. $\mathcal{F}_2$ exhibits impressive performance, achieving relative errors below 1% for 93.53% of static tests while utilizing less than 5% of the full-order model's time cost. However, it requires the same level of offline preparation as $\mathcal{F}_1$ and lacks robustness in handling anisotropy randomness. $\mathcal{F}_3$ stands out as the only surrogate showing robustness without the need for offline preparation. It also achieves considerable accuracy, with 99.47% of all tests displaying 2-norm relative errors below 3%. However, it does take more time compared to the other two surrogates. In summary, selecting the most suitable surrogate depends on the specific requirements of accuracy, robustness, offline preparation, and time cost for different applications. If only rough temperature profiles are needed, with no constraints in robustness and offline preparation, $\mathcal{F}_1$ is recommended for producing real-time predictions. For applications that prioritize high accuracy without excessive demands on robustness, offline preparation, and online execution time, $\mathcal{F}_2$ is suitable. $\mathcal{F}_3$, however, is recommended for scenarios that require sufficiently accurate temperature profiles while considering robustness and/or no offline preparation, with a tolerance for longer online execution time. If the thermal model was utilized in real-time control, the online execution time should be as fast as the real printing process. Namely, the online execution time of each time step should be closed to the time interval set in thermal simulation. When the scale of the thermal model increases such as a FEM model with millions of elements, in each Picard iteration there will be more nonlinear computations required and more time caused by high-dimensional matrix multiplication and inverse. It will cause a significant increase of online execution time in the FEM model as well as $\mathcal{F}_2$ and $\mathcal{F}_3$ since all of them will implement

Picard iterations in each time step. The increase of $\mathcal{F}_1$'s online execution time, however, will be less significant since it is entirely data-driven and the dimensionality increase will mainly affect its offline preparation time. In this case, it is recommended to use a high-performance computer and also apply parallel computing for each Picard iteration in each time step to finish some computations like temperature-dependent thermal properties in parallel.

All three surrogates enable fast computation of temperature profiles, thermal histories, and melt pool sizes, which are vital for models related to microstructure, residual stress, and the prediction of final part mechanical properties, deformation, and fatigue life. Consequently, they advance the quality assurance process for the final parts. In principle, these surrogates can be adapted for other types of thermal-driven additive manufacturing by appropriately adjusting domain evolution and heat source models. Future work can explore their application in solving problems that involve large and/or fast thermal simulations, such as process optimizations and closed-loop controls.

References

[1] M. B. A. Tamez and I. Taha. "A review of additive manufacturing technologies and markets for thermosetting resins and their potential for carbon fiber integration". In: *Additive Manufacturing* (2020), p. 101748.

[2] R. McCann et al. "In-situ sensing, process monitoring and machine control in Laser Powder Bed Fusion: A review". In: *Additive Manufacturing* (2021), p. 102058.

[3] J. C. Najmon, S. Raeisi, and A. Tovar. "Review of additive manufacturing technologies and applications in the aerospace industry". In: *Additive manufacturing for the aerospace industry*. Elsevier, 2019, pp. 7–31.

[4] S. G. Sarvankar and S. N. Yewale. "Additive Manufacturing in Automobile Industry". In: *International Journal of Research in Aeronautical and Mechanical Engineering* 7 (2019), pp. 1–10.

[5] C. Pinna et al. "Additive Manufacturing applications within Food industry: an actual overview and future opportunities". In: *21st Summer School Francesco Turco 2016*. AIDI-Italian Association of Industrial Operations Professors. 2016, pp. 18–24.

[6] C. Culmone, G. Smit, and P. Breedveld. "Additive manufacturing of medical instruments: A state-of-the-art review". In: *Additive Manufacturing* 27 (2019), pp. 461–473.

[7] F. Tao et al. "Digital twin-driven product design, manufacturing and service with big data". In: *The International Journal of Advanced Manufacturing Technology* 94.9 (2018), pp. 3563–3576.

[8] D. Gunasegaram et al. “Towards developing multiscale-multiphysics models and their surrogates for digital twins of metal additive manufacturing”. In: *Additive Manufacturing* 46 (2021), p. 102089.

[9] J. Zhang et al. “Designing against phase and property heterogeneities in additively manufactured titanium alloys”. In: *Nature communications* 13.1 (2022), pp. 1–10.

[10] Z. Hu and S. Mahadevan. “Uncertainty quantification and management in additive manufacturing: current status, needs, and opportunities”. In: *The International Journal of Advanced Manufacturing Technology* 93.5 (2017), pp. 2855–2874.

[11] M. Russell, A. Souto-Iglesias, and T. Zohdi. “Numerical simulation of Laser Fusion Additive Manufacturing processes using the SPH method”. In: *Computer Methods in Applied Mechanics and Engineering* 341 (2018), pp. 163–187.

[12] S. Kollmannsberger et al. “Accurate prediction of melt pool shapes in laser powder bed fusion by the non-linear temperature equation including phase changes”. In: *Integrating Materials and Manufacturing Innovation* 8.2 (2019), pp. 167–177.

[13] Y. Li and D. Gu. “Parametric analysis of thermal behavior during selective laser melting additive manufacturing of aluminum alloy powder”. In: *Materials & Design* 63 (2014), pp. 856–867.

[14] P. Foteinopoulos, A. Papacharalampopoulos, and P. Stavropoulos. “On thermal modeling of Additive Manufacturing processes”. In: *CIRP Journal of Manufacturing Science and Technology* 20 (2018), pp. 66–83.

[15] M. Sheikhi, F. M. Ghaini, and H. Assadi. “Prediction of solidification cracking in pulsed laser welding of 2024 aluminum alloy”. In: *Acta Materialia* 82 (2015), pp. 491–502.

[16] K. Ramanathan and S. Yen. “High-temperature emissivities of copper, aluminum, and silver”. In: *JOSA* 67.1 (1977), pp. 32–38.

[17] S. Zhang, A. L. Gain, and J. A. Norato. "Adaptive mesh refinement for topology optimization with discrete geometric components". In: *Computer Methods in Applied Mechanics and Engineering* 364 (2020), p. 112930.

[18] M. G. Larson and F. Bengzon. "The finite element method: theory, implementation, and practice". In: *Texts in Computational Science and Engineering* 10 (2010).

[19] X. Li and N. Polydorides. "Time-efficient surrogate models of thermal modeling in laser powder bed fusion". In: *Additive Manufacturing* 59 (2022), p. 103122.

[20] X. Li and N. Polydorides. "Fast heat transfer simulation for laser powder bed fusion". In: *Computer Methods in Applied Mechanics and Engineering* 412 (2023), p. 116107.

[21] Xiaohan3795. *Fast_heat_transfer_simulation_for_laser_powder_bed_fusion*. Mar. 2023. URL: `https://github.com/Xiaohan3795/Fast_heat_transfer_simulation_for_laser_powder_bed_fusion`.

[22] S. Sing and W. Yeong. "Laser powder bed fusion for metal additive manufacturing: perspectives on recent developments". In: *Virtual and Physical Prototyping* 15.3 (2020), pp. 359–370.

[23] M. Wang et al. "Effects of annealing on the microstructure and mechanical properties of selective laser melted AlSi7Mg alloy". In: *Materials Science and Engineering: A* 739 (2019), pp. 463–472.

[24] L. Zhuo et al. "Effect of post-process heat treatment on microstructure and properties of selective laser melted AlSi10Mg alloy". In: *Materials Letters* 234 (2019), pp. 196–200.

[25] X. Li et al. "A selective laser melting and solution heat treatment refined Al–12Si alloy with a controllable ultrafine eutectic microstructure and 25% tensile ductility". In: *Acta Materialia* 95 (2015), pp. 74–82.

[26] P. Ma et al. "Influence of annealing on mechanical properties of Al-20Si processed by selective laser melting". In: *Metals* 4.1 (2014), pp. 28–36.

[27] R. Zhao et al. “On the role of volumetric energy density in the microstructure and mechanical properties of laser powder bed fusion Ti-6Al-4V alloy”. In: *Additive Manufacturing* 51 (2022), p. 102605.

[28] A. M. Beese and B. E. Carroll. “Review of mechanical properties of Ti-6Al-4V made by laser-based additive manufacturing using powder feedstock”. In: *Jom* 68.3 (2016), pp. 724–734.

[29] C. Qiu, Y. Liu, and H. Liu. “Influence of addition of TiAl particles on microstructural and mechanical property development in Invar 36 processed by laser powder bed fusion”. In: *Additive Manufacturing* 48 (2021), p. 102457.

[30] E. Krohmer et al. “Revealing dynamic processes in laser powder bed fusion with in situ X-ray diffraction at PETRA III”. In: *Review of Scientific Instruments* 93.6 (2022), p. 065104.

[31] J. Liang et al. “Microstructure evolution of laser powder bed fusion ZK60 Mg alloy after different heat treatment”. In: *Journal of Alloys and Compounds* 898 (2022), p. 163046.

[32] X. Niu et al. “Effective control of microstructure evolution in AZ91D magnesium alloy by SiC nanoparticles in laser powder-bed fusion”. In: *Materials & Design* 206 (2021), p. 109787.

[33] A. B. Kale et al. “Effect of initial microstructure on the deformation heterogeneities of 316L stainless steels fabricated by selective laser melting processing”. In: *Journal of Materials Research and Technology* 9.4 (2020), pp. 8867–8883.

[34] C. Luo and Y. Zhang. “Effect of printing orientation on anisotropic properties in resistance spot welded 316L stainless steels via selective laser melting”. In: *Materials Letters* 254 (2019), pp. 237–241.

[35] T. L. Starr et al. “Controlling phase composition in selective laser melted stainless steels”. In: *2012 International Solid Freeform Fabrication Symposium*. University of Texas at Austin. 2012.

[36] R. Casati et al. “Microstructure and mechanical behavior of hot-work tool steels processed by Selective Laser Melting”. In: *Materials Characterization* 137 (2018), pp. 50–57.

[37] S. Safdar et al. “An anisotropic enhanced thermal conductivity approach for modelling laser melt pools for Ni-base super alloys”. In: *Applied mathematical modelling* 37.3 (2013), pp. 1187–1195.

[38] S. Traore et al. “Influence of gas atmosphere (Ar or He) on the laser powder bed fusion of a Ni-based alloy”. In: *Journal of Materials Processing Technology* 288 (2021), p. 116851.

[39] E. A. Lass et al. “Precipitation and dissolution of δ and γ during heat treatment of a laser powder-bed fusion produced Ni-based superalloy”. In: *Scripta Materialia* 154 (2018), pp. 83–86.

[40] C. Guo et al. “Single-track investigation of IN738LC superalloy fabricated by laser powder bed fusion: Track morphology, bead characteristics and part quality”. In: *Journal of Materials Processing Technology* 290 (2021), p. 117000.

[41] O. Adegoke et al. “Influence of laser powder bed fusion process parameters on the microstructure of solution heat-treated nickel-based superalloy Alloy 247LC”. In: *Materials Characterization* 183 (2022), p. 111612.

[42] C. Pauzon. “The Process Atmosphere as a Parameter in the Laser-Powder Bed Fusion Process”. PhD thesis. Chalmers Tekniska Hogskola (Sweden), 2019.

[43] Q. Zhong et al. “Using feedback control of thermal history to improve quality consistency of parts fabricated via large-scale powder bed fusion”. In: *Additive Manufacturing* 42 (2021), p. 101986.

[44] PWC. *3D printing and the new shape of industrial manufacturing*. PricewaterhouseCoopers LLP, Delaware. 2014.

[45] J. Lee and V. Prabhu. “Simulation modeling for optimal control of additive manufacturing processes”. In: *Additive Manufacturing* 12 (2016), pp. 197–203.

[46] A. Papacharalampopoulos, P. Stavropoulos, and J. Stavridis. “Adaptive control of thermal processes: laser welding and additive manufacturing paradigms”. In: *Procedia CIRP* 67 (2018), pp. 233–237.

[47] W. E. King et al. “Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges”. In: *Applied Physics Reviews* 2.4 (2015), p. 041304.

[48] H. Mindt et al. “Powder bed layer characteristics: the overseen first-order process input”. In: *Metallurgical and Materials Transactions A* 47.8 (2016), pp. 3811–3822.

[49] E. J. Parteli and T. Pöschel. “Particle-based simulation of powder application in additive manufacturing”. In: *Powder Technology* 288 (2016), pp. 96–102.

[50] Z. Xiang et al. “Simulation of forming process of powder bed for additive manufacturing”. In: *Journal of Manufacturing Science and Engineering* 138.8 (2016).

[51] X. Dou, Y. Mao, and Y. Zhang. “Effects of contact force model and size distribution on microsized granular packing”. In: *Journal of Manufacturing Science and Engineering* 136.2 (2014).

[52] C. Boley, S. A. Khairallah, and A. M. Rubenchik. “Calculation of laser absorption by metal powders in additive manufacturing”. In: *Applied optics* 54.9 (2015), pp. 2477–2482.

[53] J. Zhou, Y. Zhang, and J. Chen. “Numerical simulation of random packing of spherical particles for powder-based additive manufacturing”. In: *Journal of manufacturing science and engineering* 131.3 (2009).

[54] I. A. Roberts et al. “A three-dimensional finite element analysis of the temperature field during laser melting of metal powders in additive layer manufacturing”. In: *International Journal of Machine Tools and Manufacture* 49.12-13 (2009), pp. 916–923.

[55] A. Foroozmehr et al. “Finite element simulation of selective laser melting process considering optical penetration depth of laser in powder bed”. In: *Materials & Design* 89 (2016), pp. 255–263.

[56] J. Goldak, A. Chakravarti, and M. Bibby. “A new finite element model for welding heat sources”. In: *Metallurgical transactions B* 15.2 (1984), pp. 299–305.

[57] C. Wu, H. Wang, Y. Zhang, et al. “A new heat source model for keyhole plasma arc welding in FEM analysis of the temperature profile”. In: *WELDING JOURNAL-NEW YORK-* 85.12 (2006), p. 284.

[58] J. C. Steuben et al. “Enriched analytical solutions for additive manufacturing modeling and simulation”. In: *Additive Manufacturing* 25 (2019), pp. 437–447.

[59] B. Schoinochoritis, D. Chantzis, and K. Salonitis. “Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review”. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 231.1 (2017), pp. 96–117.

[60] S. A. Khairallah et al. “Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones”. In: *Acta Materialia* 108 (2016), pp. 36–45.

[61] S. A. Khairallah and A. Anderson. “Mesoscopic simulation model of selective laser melting of stainless steel powder”. In: *Journal of Materials Processing Technology* 214.11 (2014), pp. 2627–2636.

[62] M. Markl and C. Körner. “Multiscale modeling of powder bed–based additive manufacturing”. In: *Annual Review of Materials Research* 46 (2016), pp. 93–123.

[63] J. Zhang et al. “A coupled finite element cellular automaton model to predict thermal history and grain morphology of Ti-6Al-4V during direct metal deposition (DMD)”. In: *Additive Manufacturing* 11 (2016), pp. 32–39.

[64] H. Azizi et al. “Characterizing the microstructural effect of build direction during solidification of laser-powder bed fusion of Al-Si alloys in the dilute limit: A phase-field study”. In: *Acta Materialia* 214 (2021), p. 116983.

[65] P. Mercelis and J.-P. Kruth. “Residual stresses in selective laser sintering and selective laser melting”. In: *Rapid prototyping journal* (2006).

[66] Y. Li et al. “Modeling temperature and residual stress fields in selective laser melting”. In: *International Journal of Mechanical Sciences* 136 (2018), pp. 24–35.

[67] Y. Yang et al. “Prediction of microstructure, residual stress, and deformation in laser powder bed fusion process”. In: *Computational Mechanics* 61.5 (2018), pp. 599–615.

[68] J. Yin et al. “Simulation of temperature distribution in single metallic powder layer for laser micro-sintering”. In: *Computational Materials Science* 53.1 (2012), pp. 333–339.

[69] F. Cernuschi et al. “Modelling of thermal conductivity of porous materials: application to thick thermal barrier coatings”. In: *Journal of the European Ceramic Society* 24.9 (2004), pp. 2657–2667.

[70] C. Butler et al. “Effects of processing parameters and heat treatment on thermal conductivity of additively manufactured AlSi10Mg by selective laser melting”. In: *Materials Characterization* 173 (2021), p. 110945.

[71] J. Ning et al. “Analytical thermal modeling of powder bed metal additive manufacturing considering powder size variation and packing”. In: *Materials* 13.8 (2020), p. 1988.

[72] R. J. Crawford and J. L. Throne. *Rotational molding technology*. William Andrew, 2001.

[73] Y. Shu et al. “A critical look at the prediction of the temperature field around a laser-induced melt pool on metallic substrates”. In: *Scientific reports* 11.1 (2021), pp. 1–11.

[74] K. Karayagiz et al. “Numerical and experimental analysis of heat distribution in the laser powder bed fusion of Ti-6Al-4V”. In: *IISE Transactions* 51.2 (2019), pp. 136–152.

[75] S. Nikam et al. “On the application of the anisotropic enhanced thermal conductivity approach to thermal modelling of laser-based powder bed fusion processes”. In: *Additive Manufacturing* 55 (2022), p. 102870.

[76] F. Corradini and M. Silvestri. “Design and testing of a digital twin for monitoring and quality assessment of material extrusion process”. In: *Additive Manufacturing* 51 (2022), p. 102633.

[77] C. Cimino, E. Negri, and L. Fumagalli. “Review of digital twin applications in manufacturing”. In: *Computers in Industry* 113 (2019), p. 103130.

[78] A. Ladj et al. “A knowledge-based Digital Shadow for machining industry in a Digital Twin perspective”. In: *Journal of Manufacturing Systems* 58 (2021), pp. 168–179.

[79] N. Stojanovic and D. Milenovic. “Data-driven Digital Twin approach for process optimization: An industry use case”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 4202–4211.

[80] G. P. Tancredi, G. Vignali, and E. Bottani. “Integration of digital twin, machine-learning and industry 4.0 tools for anomaly detection: An application to a food plant”. In: *Sensors* 22.11 (2022), p. 4143.

[81] R. Y. Zhong et al. “Intelligent manufacturing in the context of industry 4.0: a review”. In: *Engineering* 3.5 (2017), pp. 616–630.

[82] A. du Plessis. “Effects of process parameters on porosity in laser powder bed fusion revealed by X-ray tomography”. In: *Additive Manufacturing* 30 (2019), p. 100871.

[83] J. P. Oliveira, A. LaLonde, and J. Ma. “Processing parameters in laser powder bed fusion metal additive manufacturing”. In: *Materials & Design* 193 (2020), p. 108762.

[84] N. Levkulich et al. “The effect of process parameters on residual stress evolution and distortion in the laser powder bed fusion of Ti-6Al-4V”. In: *Additive Manufacturing* 28 (2019), pp. 475–484.

[85] T. Debroy et al. “Building digital twins of 3D printing machines”. In: *Scripta Materialia* 135 (2017), pp. 119–124.

[86] M. J. Matthews et al. “Denudation of metal powder layers in laser powder-bed fusion processes”. In: *Additive Manufacturing Handbook*. CRC Press, 2017, pp. 677–692.

[87] C. G. Klingaa et al. “Towards a digital twin of laser powder bed fusion with a focus on gas flow variables”. In: *Journal of Manufacturing Processes* 65 (2021), pp. 312–327.

[88] S. Chakraborty, S. Adhikari, and R. Ganguli. “The role of surrogate models in the development of digital twins of dynamic systems”. In: *Applied Mathematical Modelling* 90 (2021), pp. 662–681.

[89] Á. Bárkányi et al. “Modelling for digital twins—potential role of surrogate models”. In: *Processes* 9.3 (2021), p. 476.

[90] M. J. Asher et al. “A review of surrogate models and their application to groundwater modeling”. In: *Water Resources Research* 51.8 (2015), pp. 5957–5973.

[91] M. Tang, Y. Liu, and L. J. Durlofsky. “A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems”. In: *Journal of Computational Physics* 413 (2020), p. 109456.

[92] M. Mozaffar et al. “Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks”. In: *Manufacturing letters* 18 (2018), pp. 35–39.

[93] M. Roy and O. Wodo. “Data-driven modeling of thermal history in additive manufacturing”. In: *Additive Manufacturing* 32 (2020), p. 101017.

[94] C. K. J. Hou and K. Behdinan. “Dimensionality Reduction in Surrogate Modeling: A Review of Combined Methods”. In: *Data Science and Engineering* (2022), pp. 1–26.

[95] Y. Liu et al. “A global surrogate model technique based on principal component analysis and Kriging for uncertainty propagation of dynamic systems”. In: *Reliability Engineering & System Safety* 207 (2021), p. 107365.

[96] W. Xing, A. A. Shah, and P. B. Nair. “Reduced dimensional Gaussian process emulators of parametrized partial differential equations based on Isomap”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2174 (2015), p. 20140697.

[97] S. Nikolopoulos, I. Kalogeris, and V. Papadopoulos. “Non-intrusive surrogate modeling for parametrized time-dependent PDEs using convolutional autoencoders”. In: *arXiv preprint arXiv:2101.05555* (2021).

[98] A. Carracedo Rodriguez, S. Gugercin, and J. Borggaard. “Interpolatory model reduction of parameterized bilinear dynamical systems”. In: *Advances in Computational Mathematics* 44.6 (2018), pp. 1887–1916.

[99] P. Benner and P. Goyal. “Balanced truncation model order reduction for quadratic-bilinear control systems”. In: *arXiv preprint arXiv:1705.00160* (2017).

[100] B. Kramer and K. E. Willcox. “Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition”. In: *AIAA Journal* 57.6 (2019), pp. 2297–2307.

[101] A. C. Rodriguez, S. Gugercin, and J. Borggaard. “Interpolatory model reduction of parameterized bilinear dynamical systems”. In: *Advances in Computational Mathematics* 44.6 (2018), pp. 1887–1916.

[102] Y. Choi et al. “Space–time reduced order model for large-scale linear dynamical systems with application to boltzmann transport problems”. In: *Journal of Computational Physics* 424 (2021), p. 109845.

[103] R. Lung et al. “A sketched finite element method for elliptic models”. In: *Computer Methods in Applied Mechanics and Engineering* 364 (2020), p. 112933.

[104] S. Chaturantabut and D. C. Sorensen. “Nonlinear model reduction via discrete empirical interpolation”. In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764.

[105] M. Masoomi, S. M. Thompson, and N. Shamsaei. “Laser powder bed fusion of Ti-6Al-4V parts: Thermal modeling and mechanical implications”. In: *International Journal of Machine Tools and Manufacture* 118 (2017), pp. 73–90.

[106] Y. Zhang et al. “Multifidelity surrogate based on single linear regression”. In: *AIAA Journal* 56.12 (2018), pp. 4944–4952.

[107] Z.-H. Han, Zimmermann, and S. Görtz. “Alternative cokriging method for variable-fidelity surrogate modeling”. In: *AIAA journal* 50.5 (2012), pp. 1205–1210.

[108] J. G. S. Macıas et al. “Influence on microstructure, strength and ductility of build platform temperature during laser powder bed fusion of AlSi10Mg”. In: *Acta Materialia* 201 (2020), pp. 231–243.

[109] S. Schmid et al. “A new approach for automated measuring of the melt pool geometry in laser-powder bed fusion”. In: *Progress in Additive Manufacturing* 6.2 (2021), pp. 269–279.

[110] S. Torquato and H. Haslach Jr. “Random heterogeneous materials: microstructure and macroscopic properties”. In: *Appl. Mech. Rev.* 55.4 (2002), B62–B63.

[111] V. Silva et al. “Machine learning acceleration for nonlinear solvers applied to multiphase porous media flow”. In: *Computer Methods in Applied Mechanics and Engineering* 384 (2021), p. 113989.

[112] J. B. Tenenbaum, V. De Silva, and J. C. Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *science* 290.5500 (2000), pp. 2319–2323.

[113] E. Mehrbani and M. H. Kahaei. “Low-Rank Isomap Algorithm”. In: *arXiv preprint arXiv:2103.04060* (2021).

[114] L. M. van den Bos, B. Sanderse, and W. Bierbooms. “Adaptive sampling-based quadrature rules for efficient Bayesian prediction”. In: *Journal of Computational Physics* 417 (2020), p. 109537.

[115] P. Keast. “Moderate-degree tetrahedral quadrature formulas”. In: *Computer methods in applied mechanics and engineering* 55.3 (1986), pp. 339–348.

[116] I. C. Ipsen and T. Wentworth. “The effect of coherence on sampling from matrices with orthonormal columns, and preconditioned least squares problems”. In: *SIAM Journal on Matrix Analysis and Applications* 35.4 (2014), pp. 1490–1520.

[117] Y. Aizenbud and A. Averbuch. “Matrix decompositions using sub-Gaussian random matrices”. In: *Information and Inference: A Journal of the IMA* 8.3 (2019), pp. 445–469.

[118] C. Li et al. “Microstructure and mechanical properties of TiC/AlSi10Mg alloy fabricated by laser additive manufacturing under high-frequency micro-vibration”. In: *Journal of Alloys and Compounds* 794 (2019), pp. 236–246.

[119] C. Liu et al. “Modeling of thermal behavior and microstructure evolution during laser cladding of AlSi10Mg alloys”. In: *Optics & Laser Technology* 123 (2020), p. 105926.

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

[121] S. Lucas. “Measurement of Cp/Cv for argon, nitrogen, carbon dioxide and an argon+ nitrogen mixture”. In: *University College London* (2010).

[122] S. Chen and S. Saxena. “Thermal conductivity of argon in the temperature range 350 to 2500 K”. In: *Molecular Physics* 29.2 (1975), pp. 455–466.

[123] L. Chen et al. “Simulation of surface deformation control during selective laser melting of AlSi10Mg powder using an external magnetic field”. In: *AIP Advances* 9.4 (2019), p. 045012.