Naval Research Laboratory
Washington, DC 20375-5320

NRL/6390/MR—2022/1

Multiscale/Multifunctional Tailoring of Parts Produced by Powder-Based Additive Manufacturing via Design Optimization Incorporating Hyper-Reduced-Order Modeling

STEVEN N. RODRIGUEZ
ATHANASIOS ILIOPOULOS
JOHN C. STEUBEN
JOHN G. MICHOPOULOS

Center for Materials Physics and Technology
Material Science and Technology Division

January 31, 2022

REPORT DOCUMENTATION PAGE		*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 31-01-2022	**2. REPORT TYPE** NRL Memorandum Report	**3. DATES COVERED** *(From - To)* June 9, 2019 – June 9, 2021

4. TITLE AND SUBTITLE Multiscale/Multifunctional Tailoring of Parts Produced by Powder-Based Additive Manufacturing via Design Optimization Incorporating Hyper-Reduced-Order Modeling	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER** NISE
6. AUTHOR(S) Steven N. Rodriguez, Athanasios Iliopoulos, John C. Steuben, and John G. Michopoulos	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER** N2U1
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320	**8. PERFORMING ORGANIZATION REPORT NUMBER** NRL/6390/MR--2022/1
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320	**10. SPONSOR / MONITOR'S ACRONYM(S)** NRL-NISE **11. SPONSOR / MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Karles Fellowship

14. ABSTRACT

Physics insight gained from computational modeling and simulation (CMS) of additive manufacturing (AM) processes can help facilitate effective multiscale/multifunctional tailoring of AM parts. However, accurately capturing the physics of AM with traditional CMS can be costly with respect to resources and time. The main objective of the work described in this document is to explore projection-based reduced-order modeling (PROM) as a means to reduce CMS resources and time of the heat-deposition additive manufacturing physics. The presented PROM framework can be deployed to accelerate multi-query settings, as seen in design optimization frameworks for the multiscale/multifunctional tailoring of AM-based parts.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Steven Rodriguez
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	U	33	**19b. TELEPHONE NUMBER** *(include area code)* (202) 404-8630

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank.

CONTENTS

FIGURES

TABLES

This page intentionally left blank

EXECUTIVE SUMMARY

Physics insight gained from computational modeling and simulation (CMS) of additive manufacturing (AM) processes can help facilitate effective multiscale/multifunctional tailoring of AM parts. However, accurately capturing the physics of AM with traditional CMS can be costly with respect to resources and time. The main objective of the work described in this document is to explore projection-based reduced-order modeling (PROM) as a means to reduce CMS resources and time of the heat-deposition additive manufacturing physics. The presented PROM framework can be deployed to accelerate multi-query settings, as seen in design optimization frameworks for the multiscale/multifunctional tailoring of AM-based parts.

This page intentionally left blank

MULTISCALE/MULTIFUNCTIONAL TAILORING OF PARTS PRODUCED BY POWDER-BASED ADDITIVE MANUFACTURING VIA DESIGN OPTIMIZATION INCORPORATING HYPER-REDUCED-ORDER MODELING

1. RESEARCH SUMMARY

The research effort objective of this Karles Fellowship program was to leverage projection-based reduced-order modeling (PROM) to reduce computational costs and associated compute time to simulate additive manufacturing heat-deposition processes. The efforts under this fellowship resulted in two major contributions: 1) the development of a novel mathematical framework, the *projection-tree reduce-order modeling* (PTROM) methodology, that drastically reduces cost and compute-time of computational modeling and simulations (CMS) in nonlocal meshless frameworks; and 2) the first reported application of PROM with smoothed particle hydrodynamics for additive manufacturing CMS. This report is structured as follows. Section 2 introduces the smoothed particle hydrodynamics method from an overhead perspective and within the physics context of additive manufacturing. Next, Section 3 introduces the PROM mathematical framework employed under the current fellowship, namely the least-squares Petrov–Galerkin method equipped with hyper-reduction. Section 4 showcases the deployment of the PROM on four separate heat-deposition numerical experiments under a reproducible setting. Section 5 provides conclusions and outlook of the research developments under the current Karles Fellowship program. Finally, it is important to highlight that the PTROM was an auxiliary development and its details are out of the scope of this report. However, the PTROM is briefly discussed in Appendix A.

2. SMOOTHED PARTICLE HYDRODYNAMICS

Smoothed particle hydrodynamics (SPH) has recently been adopted by several disciplines in the sciences and engineering. Recent works have employed SPH to model dark-matter halos, free-surface fluid-structure interactions, and additive manufacturing, among many other applications [1]. The SPH framework is based on a nonlocal meshless formulation used to discretize a continuum with a set of discrete particles [2, 3]. A brief overview of the method is now introduced.

2.1 Smoothed particle hydrodynamics formulation

The SPH formulation relies on the following representation of a function in integral form,

$$f(\boldsymbol{r}) = \int_{\Omega} f(\boldsymbol{r}')\delta\left(\boldsymbol{r} - \boldsymbol{r}'\right) dV' \tag{1}$$

such that $\boldsymbol{r},\ \boldsymbol{r}' \in \mathbb{R}^3$, $dV' \in \mathbb{R}_+$ is a differential volume element, Ω is the volume of the integral in Eq. 1 containing $\boldsymbol{r}$, a position vector in Cartesian space with coordinates $x, y, z \in \mathbb{R}$, and where

$$\delta(\boldsymbol{r} - \boldsymbol{r}') = \begin{cases} 1, & \boldsymbol{r} = \boldsymbol{r}' \\ 0, & \boldsymbol{r} \neq \boldsymbol{r}' \end{cases} \tag{2}$$

Manuscript approved January 23, 2022.

is the Dirac delta function. The SPH framework approximates Eq. 1 by the following expression,

$$f(\boldsymbol{r}) \approx \langle f(\boldsymbol{r}) \rangle := \int_{\Omega_S} f(\boldsymbol{r}')W(\boldsymbol{r} - \boldsymbol{r}', h)\, d\boldsymbol{V}', \tag{3}$$

with a smoothing kernel $W(\boldsymbol{r} - \boldsymbol{r}', h) \in \mathbb{R}$, that exists over a finite domain Ω_S, where $h \in \mathbb{R}_+$ is defined as the smoothing length of the kernel (i.e., the radius of the finite domain Ω_S), and where the brackers $\langle\rangle$ denote an approximation of a function by SPH convention. A specific kernel will be introduced later in Section 2.2. Next, the SPH framework approximates Eq. 3 by a summation of *Lagrangian particles*, such that the differential volume element can be represented over a collection of infinitesimal particle volumes $V_j = \frac{m_j}{\rho_j}$, where $m_j, \rho_j \in \mathbb{R}_+$, and the resulting approximation is expressed by

$$\langle f(\boldsymbol{r}) \rangle \approx \sum_{j=1}^{N_i} f(\boldsymbol{r}_j) W\left(\boldsymbol{r} - \boldsymbol{r}_j, h\right) V_j. \tag{4}$$

Here, $N_i \in \mathbb{N}$ is the number of neighboring particles in the finite domain Ω_S of the i^{th} particle of interest. Neighboring particles are denoted with the subscript j. Also, the set of N_i neighboring particles is defined by $\mathcal{N}_i \equiv \mathbb{N}(N_i)$ where $\mathbb{N}(N_i) := \{1 \ldots N_i\}$, such that $j \in \mathcal{N}_i$. Note that $\mathcal{N}_i \subseteq \mathcal{N}$, where $\mathcal{N} \equiv \mathbb{N}(N)$ and $N \in \mathbb{N}$ is the total number of particles, and $i, j \in \mathcal{N}$. Finally, differentiation of a function within the SPH formulation is represented as follows [2, 3]:

$$\langle \nabla f(\boldsymbol{r}) \rangle \approx -\sum_{j=1}^{N_i} f(\boldsymbol{r}_j) \nabla W\left(\boldsymbol{r} - \boldsymbol{r}_j, h\right) V_j, \tag{5}$$

where $\nabla := \{\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\}$ is the three-dimensional Cartesian gradient operator.

2.2 Modeling and simulating additive manufacturing

The work performed under the Karles Fellowship program concerned a heat-deposition problem emulating AM processes, and modeled the heat equation with temperature-dependent density $\rho(u)$, temperature-dependent conductivity $k(u)$, and temperature-dependent specific heat $c_v(u)$

$$\rho(u)c_v(u)\frac{du}{dt} = \nabla \cdot (k(u)\nabla u) + Q, \tag{6}$$

where $u \in \mathbb{R}$ denotes temperature. Temperature dependence of material properties is implied for notational simplicity throughout the remainder of this paper, and so the temperature argument is suppressed, such that $\rho \equiv \rho(u)$, $c_v \equiv c_v(u)$, $k \equiv k(u)$, and heat diffusivity $\alpha := k/\rho c_v$. Here, $t \in [0, T_f]$ corresponds to time, where $T_f \in \mathbb{R}_+$ is the final time considered and Q is internally generated heat density source term.

Equation 6 can be represented in the SPH formulation by employing the approximations introduced in Section 2. The resulting temperature-dependent heat equation in SPH form, as presented in [4], is expressed as follows:

$$\rho_i c_{v,i} \frac{du_i}{dt} = \sum_{j=1}^{N_i} \frac{m_j}{\rho_j} \frac{4k_i k_j}{\left(k_i + k_j\right)} (u_i - u_j) \frac{\boldsymbol{r}_{ij}}{r_{ij}^2} \cdot \nabla W_{ij} + \rho_i \frac{q_i A_i}{m_i}. \tag{7}$$

In this work, the Wendland smoothing kernel [2] is employed and is expressed as

$$W_{ij}(s,h) := \frac{1}{\pi h^3} \begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3, & 0 \le s \le 1 \\ \frac{1}{4}\left(2-s\right)^3, & 1 \le s \le 2 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Here, $\boldsymbol{r}_{ij} := \boldsymbol{r}_i - \boldsymbol{r}_j$, $r_{ij} := \|\boldsymbol{r}_{i,j}\|_2$, where $\boldsymbol{r}_i$ and $\boldsymbol{r}_j \in \mathbb{R}^3$ are defined as the position vectors of a particle i and j, respectively, in Cartesian space; and $s := r_{ij}/h$. The mass of particle i is defined by $m_i := (\rho V)/N_p$, where $V \in \mathbb{R}_+$ is the total volume of the numerical domain under consideration. Next, A_i represents the fixed square area that is encompassed by the particle, i. In the current work, the heat flux is applied on the $x - y$ plane and so the particle area is also in the $x - y$ plane. The resulting particle area definition is $A_i := \Delta x \Delta y$, where $\Delta x, \Delta y \in \mathbb{R}_+$ are predefined distances between the center of particles in the x and y directions for the current work [4]. Note that for boundary particles $A_i := \frac{1}{2}\Delta x \Delta y$ and for corner particles $A_i := \frac{1}{4}\Delta x \Delta y$. The definition of the area encompassed by a particle is simplistic but sufficient for the current work, as the particle positions remain fixed and result in simple geometries of AM components. However, a more general approach for complex geometries with deforming domains could be adopted to define the areas encompassed by particles as presented in [4]. Finally, the heat-flux, $q_i \in \mathbb{R}_+$, applied on particle i is defined by the following spatial Gaussian distribution,

$$q_i := \frac{q_0}{2\pi\sigma^2} e^{\frac{-(\bar{x}_i^2 + \bar{y}_i^2)}{2\pi\sigma^2}}, \tag{9}$$

where $q_0 \in \mathbb{R}_+$ is the heat deposition power, and $\sigma \in \mathbb{R}_+$ is a width distribution parameter. Here, $\bar{x}_i := x_i + tv_x$ and $\bar{y}_i := y_i + tv_y$ correspond to the surface location of the heat-deposition, where $v_x, v_y \in \mathbb{R}$ are the velocity components of the heat-deposition path. An illustration of the heat-deposition problem discretized with SPH is presented in Fig. 1.

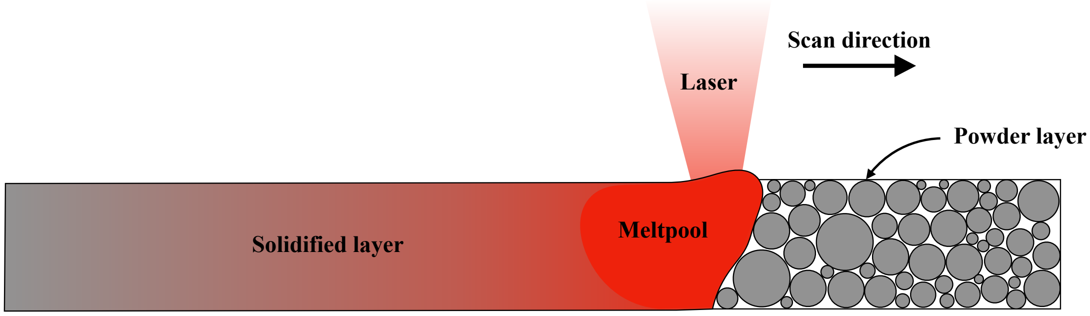


Fig. 1—Two-dimensional illustration of a heat-deposition process over a stainless steel powder-bed

The governing SPH equations are now restructured in semi-discrete vector form, which is an important structure to appropriately define PROM operations in the next section. Here, Eq. 7 is reorganized such that

$$\frac{du_i}{dt} = f_i \tag{10}$$

where

$$f_i := \frac{1}{\rho_i c_{v,i}} \sum_{j=1}^{N} \frac{m_j}{\rho_j} \frac{4k_i k_j}{\left(k_i + k_j\right)} (u_i - u_j) \frac{\boldsymbol{r}_{ij}}{r_{ij}^2} \cdot \nabla W_{ij} + \frac{q_i A_i}{c_{v,i} m_i}. \tag{11}$$

Then, the collection of SPH particles and their temperature dynamics can be expressed by the following ordinary differential equation (ODE)

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{u}, t; \boldsymbol{\mu}), \quad \boldsymbol{u}\,(0, \boldsymbol{\mu}) = \boldsymbol{u}^0(\boldsymbol{\mu}), \tag{12}$$

where $\boldsymbol{u} := \{u_1, u_2, \ldots u_N\}^T$, $\boldsymbol{f} := \{f_1, f_2, \ldots f_{N_p}\}^T$, and where T is the transpose operator. Here, $\boldsymbol{u} : [0, T_F] \times \mathcal{D} \to \mathbb{R}^N$ denotes the time-dependent parameterized state, which is implicitly defined as the solution to the SPH transient heat ODE presented in Eq. 12, with parameters $\boldsymbol{\mu} \in \mathcal{D}$. Here, $\mathcal{D} \subseteq \mathbb{R}^{n_\mu}$ denotes the parameter space of n_μ parameters, and $\boldsymbol{x}^0 : \mathcal{D} \to \mathbb{R}^N$ is the parameterized initial condition. Here, $\boldsymbol{f} : \mathbb{R}^N \times [0, T_f] \times \mathcal{D} \to \mathbb{R}^N$, which denotes the temperature-rate vector that includes heat diffusion and applied heat deposition. Note that the parametric vector, $\boldsymbol{\mu}$, in the current work denotes heat-deposition power (q_0) and path velocity v_x and v_y.

Remark 2.2.1 The current application of the SPH framework depends on a static particle domain, i.e., a Lagrangian domain that does not deform. The fixed particle domain allows the gradient of the kernel and a list of neighbors for each particle within the smoothing length to be precomputed. The precomputed neighbor list is stored in a data structure Ξ, such that $\Xi_i \equiv \mathcal{N}_i$ and $|\Xi_i| = N_i$. Furthermore, the precomputed smoothing kernel gradient is denoted by $\nabla W(\Xi)$, such that $\nabla W(\Xi_i) \equiv \nabla W_{ij}(s, h)$.

The semi-discrete SPH ODE in Eq. 12 can be discretized in time by a linear multistep scheme, where in residual form it could be stated as:

$$\boldsymbol{r}^n\,(\boldsymbol{u}^n; \boldsymbol{\mu}) = \boldsymbol{0}, \;\; n = 1 \ldots N_t \tag{13}$$

where the superscript n designates the value of a variable at time step $n \in \mathbb{N}(N_t)$, $N_t \in \mathbb{N}$ denotes the final number of time steps taken, and $\mathbb{N}(N_t) := \{1 \ldots N_t\}$. The time-discrete residual $\boldsymbol{r} : \mathbb{R}^{N_p} \times \mathcal{D} \to \mathbb{R}^{N_p}$ is defined as

$$\boldsymbol{r}^n : (\boldsymbol{\xi}^n; \boldsymbol{\nu}) \mapsto \alpha_0 \boldsymbol{\xi}^n - \Delta t \beta_0 \boldsymbol{f}(\boldsymbol{\xi}^n, t^n; \boldsymbol{\nu}) + \sum_{j=1}^{\check{k}} \alpha_j \boldsymbol{u}^{n-j} - \Delta t \sum_{j=1}^{\check{k}} \beta_j \boldsymbol{f}(\boldsymbol{u}^{n-j}, t^{n-j}, \boldsymbol{\nu}), \tag{14}$$

where $\check{k} \in \mathbb{N}$ denotes the number of steps in the multistep scheme. The current work employs backward Euler integration, such that $\check{k} = 1$; $\alpha_0 = \beta_0 = 1$, $\alpha_1 = -1$, $\beta_1 = 0$. Furthermore, the time step is denoted by $\Delta t \in \mathbb{R}_+$ and is considered uniform. Here, $\boldsymbol{u}^{\check{k}}$ denotes the numerical approximation to $\boldsymbol{u}(\check{k}\Delta t; \boldsymbol{\mu})$, and $\boldsymbol{\xi}^n \in \mathbb{R}^{N_p}$ is the unknown state vector that is implicitly solved to explicitly update the state, i.e. $\boldsymbol{u}^n \leftarrow \boldsymbol{\xi}^n$. Finally, the implicit trapezoidal integration employed herein is solved via an inexact Newton method, such that the Jacobian is updated every p_{it} time-steps, where $p_{\text{it}} := cn$ and $c \in \mathbb{N}$. It is also important to note that the current work introduces an inexact temperature rate Jacobian, to compute the residual Jacobian, where only the diagonal block entries are computed, i.e., $\partial \boldsymbol{f} / \partial \boldsymbol{\xi}$. This inexact approach is performed to avoid incurring the additional expense of computing off-diagonal blocks of the temperature-rate Jacobian that provide negligible contributions to the Jacobian of the residual.

2.2.1 Implementing temperature-dependent material properties

The current work considers 316L stainless-steel temperature-dependent density, heat-conduction, specific heat, and corresponding heat diffusivity. Equations 15-17 define the expressions for temperature-dependent behaviors where $u_{\text{melt}} = 1660.5$ K and $u_{\text{boil}} = 3090$ K as presented in [5]. Figure 2 illustrates the corresponding trends of properties as functions of temperature.

$$k(u) = \begin{cases} 9.248 + 1.571 \cdot 10^{-2} u, & u < u_{\text{melt}}, \\ 12.41 + 3.279 \cdot 10^{-3} u, & u_{\text{melt}} \leq u \leq u_{\text{boil}}, \\ 21.6465, & u_{\text{boil}} \leq u. \end{cases} \qquad \text{W/mK} \tag{15}$$

$$c_v(u) = \begin{cases} 458.984 + 0.1328 u, & u \leq u_{\text{melt}}, \\ 769.855, & u_{\text{melt}} \leq u, \end{cases} \qquad \text{J/kgK} \tag{16}$$

$$\rho(u) = \begin{cases} 8084.2 - 0.4209 u - 3.8942 \cdot 10^{-5} u^2, & u \leq u_{\text{melt}}, \\ 7432.7 + 3.9338 \cdot 10^{-2} u - 1.8007 \cdot^{-4} u^2, & u_{\text{melt}} \leq u \leq u_{\text{boil}}, \\ 6124.72, & u_{\text{boil}} \leq u. \end{cases} \qquad \text{kg/m}^3 \tag{17}$$

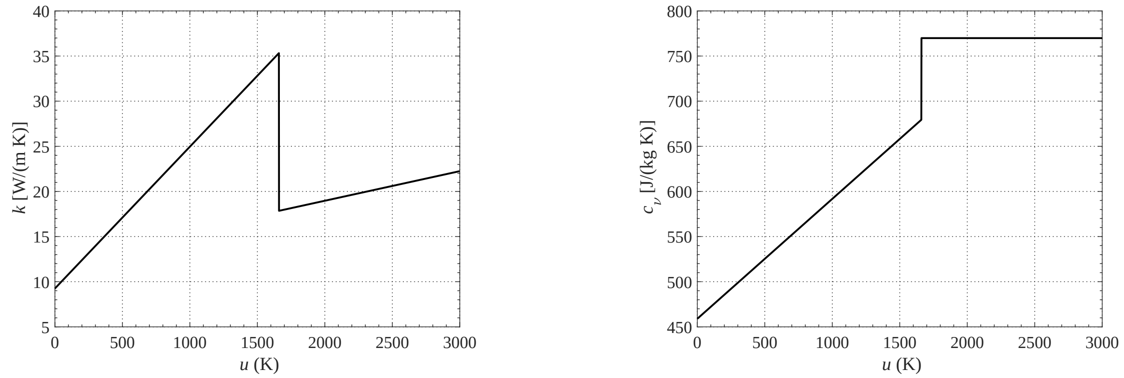
Even though explicit temperature-dependent functions are defined, the current work implements material property approximation via linear interpolation of the data generated from Eqs. 15-17 to enable a simplistic computational architecture to solve the resulting nonlinear SPH temperature-dependent heat equation in Eq. 7. The material property approximation is performed by a table look-up operation to perform interpolation, such that

$$\bar{k}(u) := a_k u + b_k, \tag{18}$$

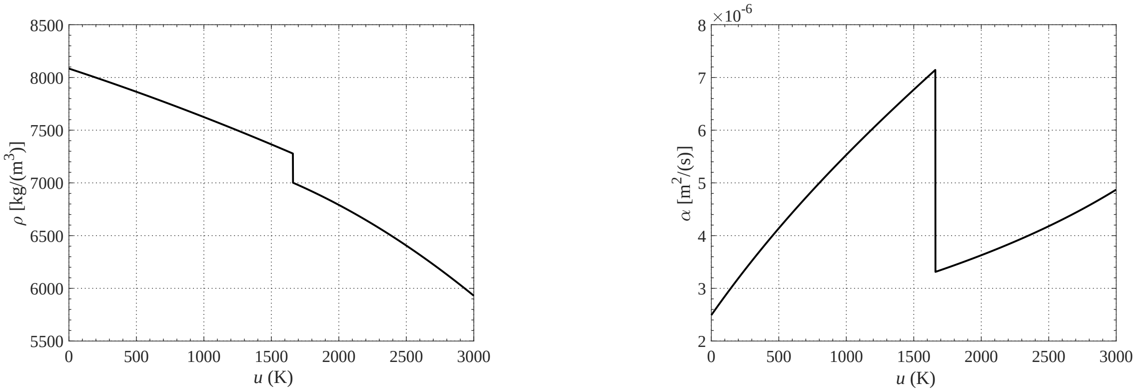
$$\bar{c}_v(u) := a_v u + b_v, \tag{19}$$

$$\bar{\rho}(u) := a_\rho u + b_\rho, \tag{20}$$

are the expressions that define the linear interpolation and are the resulting properties substituted into Eq.7. In other words, $\bar{k}(u) \approx k(u)$, $\bar{c}_v(u) \approx c_v(u)$, $\bar{\rho}(u) \approx \rho(u)$, and the coefficients $a_{(\cdot)}$ and $b_{(\cdot)}$ correspond to coefficients computed via table look-up linear interpolation.



(a) Conductivity (b) Specific heat



(c) Density (d) Thermal diffusivity

Fig. 2—Temperature-dependent material properties for 316L stainless-steel.

3. PROJECTION-BASED REDUCED-ORDER MODELING

A review of the projection-based reduced order modeling approach is now presented, where the approach presented in Refs. [1, 6, 7] is adopted in this work. To enable rapid computations of the nonlinear temperature-dependent SPH equation, i.e., Eq. 7 or equivalently Eq. 12, the current work projects the high-dimensional dynamical systems onto a low-dimensional embedding. Here, the projection seeks for an approximate solution, $\tilde{\boldsymbol{u}} \approx \boldsymbol{u}$, of the form

$$\tilde{\boldsymbol{u}}(t; \boldsymbol{\mu}) = \boldsymbol{u}_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{g}(\hat{\boldsymbol{u}}(t; \boldsymbol{\mu})) \tag{21}$$

where $\tilde{\boldsymbol{u}} : \mathbb{R}_+ \times \mathcal{D} \to \boldsymbol{u}_{\text{ref}} + \mathcal{X}$ and $\mathcal{X} := \{g(\hat{\boldsymbol{\xi}}) \,|\, \hat{\boldsymbol{\xi}} \in \mathbb{R}^M\}$ denotes some trial manifold. Here $\boldsymbol{x}_{\text{ref}} : \mathcal{D} \to \mathbb{R}^{N_p}$ denotes some parameterized reference state and $\boldsymbol{g} : \hat{\boldsymbol{\xi}} \mapsto \boldsymbol{g}(\hat{\boldsymbol{\xi}})$ with $\boldsymbol{g} : \mathbb{R}^M \to \mathbb{R}^{N_p}$ and $M \leq N$ denotes a parameterization function that projects or maps the low-dimensional generalized coordinates $\hat{\boldsymbol{u}} : \mathbb{R}_+ \times \mathcal{D} \to \mathbb{R}^M$ to the high-dimensional approximation, $\tilde{\boldsymbol{u}}$.

The current work focuses only on constructing an affine trial manifold that exist in the Stiefel manifold, i.e., for a full-column-rank matrix, $\mathcal{A} \in \mathbb{R}^{q \times d}$, the Stiefel manifold is defined by $\mathcal{V}_d(\mathbb{R}^q) \equiv \{\boldsymbol{\mathcal{A}} \in \mathbb{R}^{q \times d} \,|\, \boldsymbol{\mathcal{A}}^T \boldsymbol{\mathcal{A}} = \mathbf{I}\}$. This affine manifold is expressed as $\boldsymbol{g} : \hat{\boldsymbol{\xi}} \mapsto \boldsymbol{\Phi}\hat{\boldsymbol{\xi}}$, where $\boldsymbol{\Phi} \in \mathcal{V}_M(\mathbb{R}^{N_p})$, where the current work constructs a POD basis matrix as the mapping operator, $\boldsymbol{\Phi}$. Finally, for the current POD basis, the reference state can be set as $\boldsymbol{u}_{\text{ref}} = \boldsymbol{u}^0(\boldsymbol{\mu})$, and so the full-order model state vector can be approximated as

$$\tilde{\boldsymbol{u}} = \boldsymbol{u}^0 + \boldsymbol{\Phi}\hat{\boldsymbol{u}}. \tag{22}$$

The work presented herein only considers affine trial manifolds. However, future work will look into implementing nonlinear trial manifolds due to their potential benefits in improving robustness within the current AM PROM context. In fact, work by Lee and Carlberg [8] suggests that nonlinear manifold LSPG projection can improve the efficiency and robustness of advection-dominated flows, or transport phenomena, to overcome Kolmogorov $N-$width limitations posed by affine projections. Therefore, given that AM is driven by an advecting heat-source (via heat deposition), it is fitting to incorporate these nonlinear trial manifolds within a PROM AM setting.

3.1 Dimensionality reduction via proper-orthogonal decomposition

The projection operator, $\boldsymbol{\Phi}$, considered in the current investigation is based on constructing a low-dimensional basis via the proper orthogonal decomposition (POD). To build $\boldsymbol{\Phi}$, the method of snapshots is employed, where the singular-value decomposition (SVD) is used to factor the snapshot data matrix $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{N_p \times N_t}$, where

$$\boldsymbol{\mathcal{S}} = \left[\boldsymbol{u}^1,\ \boldsymbol{u}^2, \ldots,\ \boldsymbol{u}^{N_t-1},\ \boldsymbol{u}^{N_t}\right], \tag{23}$$

and where columns of $\boldsymbol{\mathcal{S}}$ represent the time history of the state vector, $\boldsymbol{u}$. By factoring the snapshot matrix via the SVD, we obtain

$$\boldsymbol{\mathcal{S}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{24}$$

where the left-singular matrix $\boldsymbol{U} \in \mathcal{V}_{N_p}(\mathbb{R}^{N_p})$, the singular-value matrix $\boldsymbol{\Sigma} \equiv \operatorname{diag}(\sigma_i) \in \mathbb{R}^{N_p \times N_t}$ has diagonal entries that follow a monotonic decrease such that, $\sigma_1 \geq \ldots \geq \sigma_{N_p} \geq 0$, and the right-singular matrix $\boldsymbol{V} \in \mathcal{V}_{N_t}(\mathbb{R}^{N_t})$. The POD basis used to build the low-dimensional subspace is constructed by taking the M left singular vectors of $\boldsymbol{U}$, such that $M \ll \min(N_d, N_t)$ where $\boldsymbol{\Phi} \equiv \left[U^1, \ldots, U^M\right]$. Constructing this POD basis is performed as a training step *a priori*, before any online simulations are performed.

3.2 Least-Squares Petrov–Galerkin projection

The current work employs a least-squares Petrov–Galerkin (LSPG) projection procedure to evolve the SPH system of equations in a low-dimensional embedding [6]. The LSPG approach is a time-discrete residual minimization framework, where the projected state approximation of Eq. 22 is substituted into the time-discrete residual, Eq. 14, and cast into a nonlinear least-squares formulation. The LSPG method provides discrete optimality of the residual, $\boldsymbol{r}^n(\tilde{\boldsymbol{u}}, \boldsymbol{\mu})$, at every time-step [7], such that

$$\hat{\boldsymbol{u}}^n = \underset{\boldsymbol{z} \in \mathbb{R}^M}{\arg\min} \|\boldsymbol{r}(\boldsymbol{u}_0 + \boldsymbol{\Phi}\boldsymbol{z})\|_2^2. \tag{25}$$

The solution to Eq. 25 yields the following iterative linear least-squares formulation via the Gauss-Newton method:

$$\hat{\boldsymbol{u}}^{n(k)} = \underset{\boldsymbol{z} \in \mathbb{R}^M}{\arg\min} \left\|\boldsymbol{J}^n(\boldsymbol{u}_0 + \boldsymbol{\Phi}\boldsymbol{z}^{n(k)}; \boldsymbol{\mu})\boldsymbol{\Phi}\boldsymbol{z} + \boldsymbol{r}^{n(k)}(\boldsymbol{u}_0 + \boldsymbol{\Phi}\boldsymbol{z}^{n(k)}; \boldsymbol{\mu})\right\|_2^2 \tag{26}$$

and updates to the iterative solution are given by

$$\tilde{\boldsymbol{u}}^{n(k+1)} = \tilde{\boldsymbol{u}}^{n(k)} + \alpha^{n(k)}\boldsymbol{\Phi}\Delta\hat{\boldsymbol{u}}^{n(k)} \tag{27}$$

for $k = 0, \ldots, K$, where $K \in \mathbb{N}$ is the final iteration count, and where $\alpha^{n(k)} \in \mathbb{R}$ denotes a step length in the search direction, $\Delta\hat{\boldsymbol{u}}^{n(k)} \equiv \boldsymbol{u}^{n(k)} - \boldsymbol{u}^{n(k-1)}$, that can be computed to ensure global convergence (e.g., satisfy the strong Wolfe conditions [9]). Here, the initial guesses $\hat{\boldsymbol{u}}^{n(0)}$ for the iterative problem are taken as $\hat{\boldsymbol{u}}^{n-1}$.

3.3 Hyper-reduction via Gauss–Newton with approximated tensors

Despite the evolution of the SPH system of equations in a low-dimensional sub-space, for all time steps, the LSPG method requires the evaluation of $O(kN_p)$ residual minimization operations. Therefore, a layer of reduction is required to reduce the OCC of the residual minimization counts in Eq. 26 and 27. The current work employs hyper-reduction to perform sparse residual minimization by way of the Gauss–Newton with approximated tensors approach [7]. The GNAT hyper-reduction approach enables sparse LSPG residual minimization via a weighted l^2-norm,

$$\hat{\boldsymbol{u}}^{n} = \underset{\tilde{\mathbf{x}} \in \mathbb{R}^{M}}{\arg\min} ||\boldsymbol{\Theta}\boldsymbol{r}(\tilde{\boldsymbol{u}})||_2^2. \tag{28}$$

Here, the weighting matrix, $\boldsymbol{\Theta}$, is constructed by a gappy POD approach [10], where the time-discrete residual is approximated and minimized over a sparse set of entries. The residual approximation, $\tilde{\boldsymbol{r}} \approx \boldsymbol{r}$, is constructed by way of a time-discrete residual POD basis employing the offline training procedure discussed in Section 3.1, where $\tilde{\boldsymbol{r}} = \boldsymbol{\Phi}_{\boldsymbol{r}}\hat{\boldsymbol{r}}$, such that $\boldsymbol{\Phi}_{\boldsymbol{r}} \in \mathcal{V}_{M_r}(\mathbb{R}^{N_p})$, $\hat{\boldsymbol{r}} \in \mathbb{R}^{M_{\boldsymbol{r}}}$, and $M_{\boldsymbol{r}} \ll N$ is the number of retained SVD singular vectors in $\boldsymbol{U}_{\boldsymbol{r}}$. Next, the residual minimization over a sparse set of entries is performed by the following linear least-squares problem,

$$\hat{\boldsymbol{r}} = \underset{z_r \in \mathbb{R}^{M_{\boldsymbol{r}}}}{\arg\min} \| \boldsymbol{P}\boldsymbol{\Phi}_{\mathbf{r}} z_r - \boldsymbol{P}\boldsymbol{r}(\tilde{\mathbf{u}}) \|_2^2, \tag{29}$$

where the matrix $\boldsymbol{P} \in \{0, 1\}^{n_p \times N_d}$ is a sampling matrix consisting of sparse n_p selected rows of the identity matrix, which also correspond to the same rows in the time-discrete residual vector, where $n_p \ll N_p$, and for convenience in a later discussion, let $\mathcal{N}_p \subseteq \mathcal{N}$ be the set of n_p corresponding sampled particles. The solution to Eq.29 yields

$$\hat{\boldsymbol{r}} = [\boldsymbol{P}\boldsymbol{\Phi}_{\boldsymbol{r}}]^{+}\boldsymbol{P}\boldsymbol{r}(\tilde{\boldsymbol{u}}). \tag{30}$$

Substituting Eq. 30 into $\tilde{\boldsymbol{r}} = \boldsymbol{\Phi}_{\boldsymbol{r}}\hat{\boldsymbol{r}}$ yields the residual approximation

$$\tilde{\boldsymbol{r}} = \boldsymbol{\Phi}_{\boldsymbol{r}}[\boldsymbol{P}\boldsymbol{\Phi}_{\boldsymbol{r}}]^{+}\boldsymbol{P}\boldsymbol{r}(\tilde{\boldsymbol{u}}), \tag{31}$$

whereby via the approximation, $\tilde{\boldsymbol{r}} \approx \boldsymbol{r}$, the substitution of Eq. 31 into the weighted LSPG minimization in Eq. 28 yields the following residual minimization,

$$\hat{\boldsymbol{u}} \approx \underset{\tilde{\boldsymbol{x}} \in \mathbb{R}^{M}}{\arg\min} ||\boldsymbol{\Theta}\boldsymbol{r}(\tilde{\boldsymbol{u}})||_2^2, \tag{32}$$

where the weighing matrix is defined by $\boldsymbol{\Theta} := [\boldsymbol{P}\boldsymbol{\Phi}_{\boldsymbol{r}}]^{+}\boldsymbol{P}$. To enable the sparse residual minimization of Eq. 32, the sampling matrix $\boldsymbol{P}$ is constructed via the same greedy algorithm approach presented by Algorithm 1 in reference [1] or Algorithm 3 in reference [11].

3.3.1 Offline training and online deployment

To perform LSPG with GNAT hyper-reduction, a series of offline training steps must be executed. The current work adopts training *procedure II* from [7]. This offline procedure includes performing the following:

Stage 1: Perform FOM SPH training simulations over a sampled set of points in a desired parametric space $\mathcal{D}$ and store the state-vector time-history in a snapshot matrix.

Stage 2: Compute the POD basis, $\boldsymbol{\Phi}$, of the parametric state-vector snapshot matrix $\boldsymbol{\mathcal{S}}$

Stage 3: Perform LSPG simulations without hyper-reduction over the same training points in Stage 1. Collect the time-histories of the residual vectors and each of their iterations, construct a residual snapshot matrix, and construct a residual POD basis, $\boldsymbol{\Phi}_{\boldsymbol{r}}$.

Stage 4: Construct a sampling matrix, $\boldsymbol{P}$, following a similar procedures outlined by Algorithm 1 in reference [1] or Algorithm 3 in reference [11].

Stage 5: Compute and store the weighting matrix $\boldsymbol{\Theta} = [\boldsymbol{P}\boldsymbol{\Phi}_{\boldsymbol{r}}]^{+}\boldsymbol{P}$ that enables hyper-reduction online.

Once the offline training procedures have been deployed, the online procedures, presented by Algorithm 1 and Algorithm 2 can be executed.

Finally, it is important to discuss the notation used to describe the process of efficiently querying the domain with LSPG/GNAT. Let $\mathcal{N}_q \subseteq \mathcal{N}$ be a set of $n_q \in \mathbb{N}$ particles probed online by a multi-query loop. Note that the queried particles can be the sampled particles, a subset of sampled particles, or not in the set of sampled particles, and $n_q \ll N_p$. Then, a queried particle state of interest is denoted by $\check{\boldsymbol{u}} = \check{\boldsymbol{\Phi}}\hat{\boldsymbol{u}}$, where $\check{\boldsymbol{\Phi}} := \boldsymbol{\Phi}(\mathcal{N}_q, \cdot) \in \mathbb{R}^{n_q \times M}$ denotes the map from low-dimensional embeddings to the queried particle states.

4. FULL-ORDER AND REDUCED-ORDER MODELING OF HEAT-DEPOSITION PROBLEMS

All numerical experiments were performed on a Linux machine equipped with AMD Ryzen Threadripper 3990x 64-core processors x 128 machine with 252 GB of RAM. All simulations were coded in C++ with OpenMP parallelization. It is important to note that the FOM does not require any linear algebra package since the Jacobian during the implicit time integration only has entries on its diagonal, i.e., $\boldsymbol{J}_{ii}$. Hence, inverses can be performed naïvely for the linear system of equation in the implicit time integration, i.e., $\boldsymbol{J}_{ii}^{-1} = \frac{1}{\boldsymbol{J}_{ii}}$. On the other hand, all ROM linear algebra was performed by the *Eigen* library [12]. Although *Eigen* deals with dense-dense matrix multiplication in parallel via OpenMP, it does not perform dense inverse operations or matrix decomposition in parallel. Future work will look into using dense linear algebra parallel packages, such as ScaLAPACK [13], to perform the SVD required during ROM training and to perform inverse operations during least-squares solves in the ROM.

Algorithm 1: OnlinePROM: Online LSPG/GNAT loop

```
Input: POD basis $\boldsymbol{\Phi}$; Particle index set of queried particles, $\mathcal{N}_q$, where $\check{\boldsymbol{\Phi}} := \boldsymbol{\Phi}(\mathcal{N}_q, \cdot)$;
       Pre-computed
hyper-reduction matrix, $\boldsymbol{A} := \left[\boldsymbol{P}\bar{\boldsymbol{\Phi}}_{\boldsymbol{r}}\right]^{+}$; Parametric vector $\bar{\boldsymbol{\mu}}$; Initial conditions $\bar{\bar{\boldsymbol{u}}}^0$ and $\bar{\bar{\boldsymbol{f}}}^0$; Material
properties, $\boldsymbol{\Gamma}$; Precomputed SPH gradient of the kernel, $\nabla\bar{W}(\bar{\boldsymbol{\Xi}})$; Neighbor list, $\boldsymbol{\Xi}$; Index set of
sampled particles, $\mathcal{N}_p$. An over-bar denotes minimum cardinality as a result of hyper-reduction. For
instance, $\bar{\boldsymbol{f}} := \{f_{\mathcal{N}_p(1)} \dots f_{\mathcal{N}_p(n_p)}\}^T$.
Output: Time histories of queried state vectors, $\check{\boldsymbol{u}}$.
$k = 1$ // initialize Gauss-Newton loop iteration counter
tol = user-defined tolerance
for $n = 1 \dots N_t$ do // loop over all time steps
    while $\epsilon <$ tol do
        $[\bar{\boldsymbol{r}}, \bar{\boldsymbol{J}}] =$ ComputeSPH$(\bar{\boldsymbol{\mu}}, \bar{\bar{\boldsymbol{u}}}, \bar{\bar{\boldsymbol{f}}}, \bar{\boldsymbol{\Gamma}}, \nabla\bar{W}(\bar{\boldsymbol{\Xi}}))$ // SPH computations
        $\boldsymbol{C} := \boldsymbol{P}\bar{\boldsymbol{J}}(\bar{\bar{\boldsymbol{u}}}_k^n)\bar{\boldsymbol{\Phi}}$ and $\boldsymbol{D} := \bar{\boldsymbol{P}}\bar{\boldsymbol{r}}(\bar{\bar{\boldsymbol{u}}}_k^n)$
        $\hat{\boldsymbol{u}}_{k+1} = \arg\min_{\boldsymbol{z} \in \mathbb{R}^M} ||\boldsymbol{A}\boldsymbol{C}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{D}||_2$ // compute the linear least-squares equivalent to Eq. 32
        $\hat{\boldsymbol{u}}_{k+1}^n = \hat{\boldsymbol{u}}_k^n + \alpha\Delta\hat{\boldsymbol{u}}_k$ // $\alpha$ is computed via line-search or set to 1
        $\bar{\bar{\boldsymbol{u}}}_{k+1}^n = \bar{\boldsymbol{\Phi}}\hat{\boldsymbol{u}}_{k+1}^n$ // compute states of sampled particles for SPH computations
        $\epsilon = ||\bar{\boldsymbol{\Phi}}^{\mathrm{T}}\bar{\boldsymbol{J}}(\bar{\bar{\boldsymbol{u}}}_k^n)^{\mathrm{T}}\bar{\boldsymbol{r}}(\bar{\bar{\boldsymbol{u}}}_k^n)||_2 / ||\bar{\boldsymbol{\Phi}}^{\mathrm{T}}\bar{\boldsymbol{J}}(\breve{\boldsymbol{u}})^{\mathrm{T}}\bar{\boldsymbol{r}}(\breve{\boldsymbol{u}})||_2$, where $\breve{\boldsymbol{u}} := \bar{\bar{\boldsymbol{u}}}_1^n$ // check relative reduced
         residual error
        $k \leftarrow k + 1$
    end
    $\check{\boldsymbol{u}}^n = \check{\boldsymbol{\Phi}}\hat{\boldsymbol{u}}_k^n$ // compute queried particle states
    $k = 1$
end
```

4.1 Full-order modeling set-up

All numerical experiments in this report consider a heat-deposition transverse pass over a rectangular 316L SS stainless component with the following width, length, and height, respectively: $w = 2$ mm, $l = 4$ mm, $h = 0.5$ mm. The rectangular component is discretized using SPH such that $N_w = 50$, $N_l = 100$, and $N_h = 14$ are the number of particles laid along the width, length, and height direction, respectively, for a total of $N = 70,000$ particles. The Wendland kernel is employed, and the smoothing length considered is $h = 1.5\max(\Delta x, \Delta y, \Delta z)$. Four heat-deposition problems were considered in this study, which accounted for variable laser power and speed. Parameters considered for the numerical experiments are listed in Table 1. Full-order SPH models were performed for 4ms with a time step of $\Delta t = 4\mu$s. A relative residual tolerance was set to tol $= 10^{-3}$ and each simulation had a maximum restriction of 10 iterations per time step. The initial conditions for all experiments were set to the ambient temperature of $\boldsymbol{u}^0 = 293.15K$.

4.2 Reduced-order modeling set-up

The reduced-order models in this report entailed only a reproductive setting. In other words, the deployment of the PROM framework presented in this report was not considered under a multiquery setting

Algorithm 2: ComputeSPH; Smoothed-particle hydrodynamics computations

```
Input: Sampled parametric vector $\bar{\boldsymbol{\mu}}$, approximate state vector, $\bar{\tilde{\boldsymbol{u}}}$, approximate velocity from prior iterations
and time steps, $\bar{\tilde{\boldsymbol{f}}}$, temperature-dependent material properties, $\bar{\boldsymbol{\Gamma}}$, and pre-computed kernel gradient $\nabla\bar{W}(\Xi)$.
Output: Sampled residual $\bar{\boldsymbol{r}}$, and residual Jacobian $\bar{\boldsymbol{J}}$.
$\bar{\boldsymbol{\Gamma}} = $ UpdateMaterial$(\bar{\boldsymbol{\Gamma}}, \bar{\tilde{\boldsymbol{u}}})$ // update mat. properties via linear interpolation of data in Fig. 2
for $i = \mathcal{N}_p(1), \dots, \mathcal{N}_p(n_p)$ do
    $\mathcal{N}_{\text{neigh}} := \Xi(i)$ // define the set of neighboring particles for the current sampled particle $i$
    $n_{\text{neigh}} := |\Xi(i)|$ // define the number of neighboring particles to loop over
    for $j = \mathcal{N}_{\text{neigh}}(1), \dots, \mathcal{N}_{\text{neigh}}(n_{\text{neigh}})$ do
        $\tilde{f}_i(\Xi(i)) = \frac{1}{\rho_i c_{v,i}} \sum_j^{n_{\text{neigh}}} \frac{m_j}{\rho_j} \frac{4k_i k_j}{(k_i + k_j)} (\tilde{u}_i - \tilde{u}_j) \frac{\boldsymbol{r}_{ij}}{r_{ij}^2} \cdot \nabla W_{ij}(\Xi(i)) + \frac{q_i A_i}{c_{v,i} m_i}$ // SPH computations
    end
end
$\bar{\boldsymbol{r}} = \bar{\tilde{\boldsymbol{u}}}_k^t - \bar{\tilde{\boldsymbol{u}}}^{t-1} - \Delta t \bar{\tilde{\boldsymbol{f}}}$, // Compute the residual
$\bar{\boldsymbol{J}}_{\text{vel}} = \partial \tilde{\boldsymbol{f}} / \partial \boldsymbol{r}$, // compute the inexact Jacobian
$\bar{\boldsymbol{J}} = \bar{\boldsymbol{I}} - \Delta t \bar{\boldsymbol{J}}_{\text{vel}}$. // Compute the residual Jacobian.  Note $\bar{\boldsymbol{I}}$ is the identity matrix at sampled entries
```

Table 1—Heat-deposition parameters

	q_0 (W/m^2)	u_x (m/s)
Experiment 1	40.0	0.5
Experiment 2	100.0	0.5
Experiment 3	40.0	1.0
Experiment 4	100.0	1.0

and results do not reflect prediction of points in parametric space on which the training was not deployed. Future work will employ the current framework in a multiquery setting, such as inverse analysis used identify heat deposition profile and performance parameters. In this work, we mainly focus on assessing the impact that dimensionality of the reduced-order basis (ROB) and relative residual tolerance has on the performance and reproducibility of the FOM. Special focus is given to wall-time, CPU time, and ROM probe error relative to the FOM probes. PROMs employed in this study considered two ROB cases: Case 1) 99.9990% of the state statistical energy (SSE) is employed for the ROB, and Case 2) 99.9999% of the SSE is employed. However, every experiment employs 99.925% of residual statistical energy (RSE). It is also important to note that the rank of the residual POD basis employed is also set to the number of sampled particles employed in the hyper-reduction, i.e., $n_p = \text{ran}(\boldsymbol{\Phi}_r)$. Recall that SPH requires state information from neighboring particles. Thus, in addition to acquiring information from sampled target particles in the set $\mathcal{N}$, the framework must also gather information from source particles in the set $\mathcal{N}_s$, where $n_s = |\mathcal{N}_s|$ corresponds to the number of sources (or total neighbors of the sample particles). Table 2 lists all the parameter information for the

PROMs deployed. Note: Percent of total particles (POTP) needed for the hyper-reduced particles is defined as POTP = $100 \times (n_p + n_s)/N$.

Table 2—PROM parameters

	SSE (%)	ran($\mathbf{\Phi}$)	RSE (%)	ran($\mathbf{\Phi_r}$)	n_p	n_s	POTP (%)
Exp. 1, ROB Case 1	99.9990	148	99.925	630	630	7517	11.64
Exp. 1, ROB Case 2	99.9999	421	99.925	609	609	7697	11.72
Exp. 2, ROB Case 1	99.9990	388	99.925	933	933	7807	12.49
Exp. 2, ROB Case 2	99.9999	664	99.925	915	915	7728	12.45
Exp. 3, ROB Case 1	99.9990	108	99.925	612	612	10816	16.33
Exp. 3, ROB Case 2	99.9999	576	99.925	583	583	10844	16.32
Exp. 4, ROB Case 1	99.9990	472	99.925	1178	1178	11903	18.69
Exp. 4, ROB Case 2	99.9999	807	99.925	1143	1143	11754	18.42

4.3 Definition of performance metrics

To assess the performance of the full-order model with regard to savings in wall-time (SWT) and savings in CPU hours (SCPUH), the following definitions were employed:

$$\text{SWT} = \frac{WT_{\text{FOM}}}{WT_{\text{ROM}}}, \tag{33}$$

$$\text{SCPUH} = \frac{\sum_{i=1}^{\#\text{ of cores}} T_{\text{FOM},i}}{\sum_{i=1}^{\#\text{ of cores}} T_{\text{ROM},i}}. \tag{34}$$

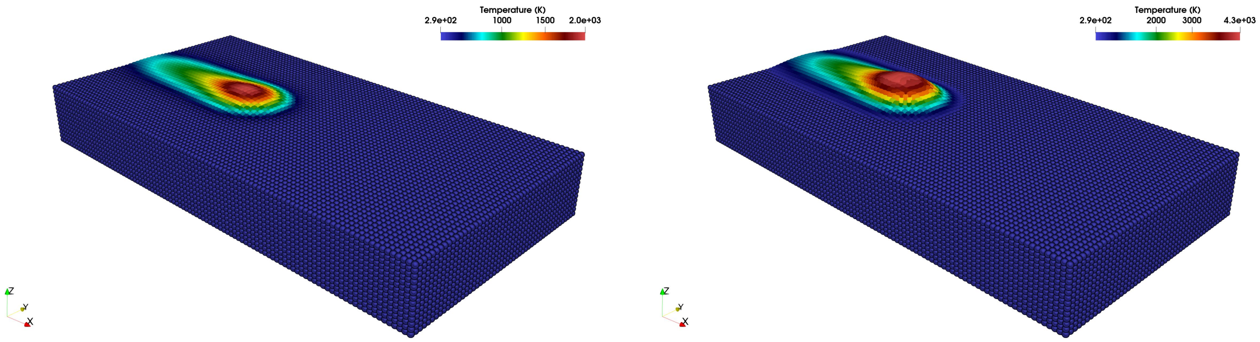
To assess the performance with regard to reproducibility of the FOM, the following definition was employed:

$$\text{Percent error} = 100 \times \left| \frac{\boldsymbol{u}^n_{ROM,i} - \boldsymbol{u}^n_{FOM,i}}{\boldsymbol{u}^n_{FOM,i}} \right|, \tag{35}$$

where the subscripts i and n correspond to a probe at particle i at time step n.

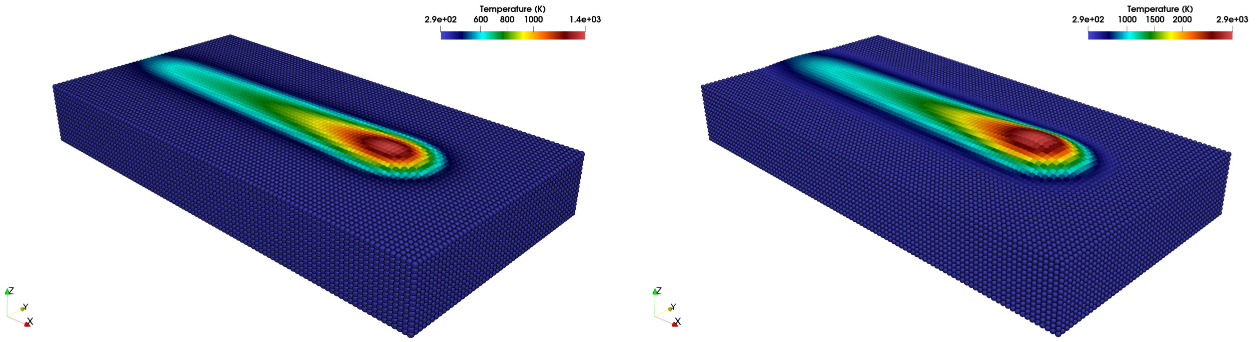
4.4 Results

Full-order model temperature plots are shown in Figure 3. Experiments reflect relatively high-temperature profiles when the laser pass moves slowly or when the heat flux power is high. Four temperature probes were placed on the surface of the specimen to collect temperature data that will be compared against the ROM. The location of each probe is listed in Table 3. Reduced-order model temperature plots are shown in Figure 4. Experiments reflect relatively high temperature profiles when the laser pass moves slowly or when the heat flux power is high. Four temperature probes were placed at the same location as the probes used for the FOM to collect temperature data. Note that the gray particles illustrated in Fig. 4 represent the outline of the full particle domain. However, the gray particles are not loaded or active in the ROM simulation,



(a) Experiment 1: $q_0 = 40$ W/m^2 and $u_x = 0.5$ m/s^2.

(b) Experiment 2: $q_0 = 100$ W/m^2 and $u_x = 0.5$ m/s^2.



(c) Experiment 3: $q_0 = 40$ W/m^2 and $u_x = 1$ m/s^2.

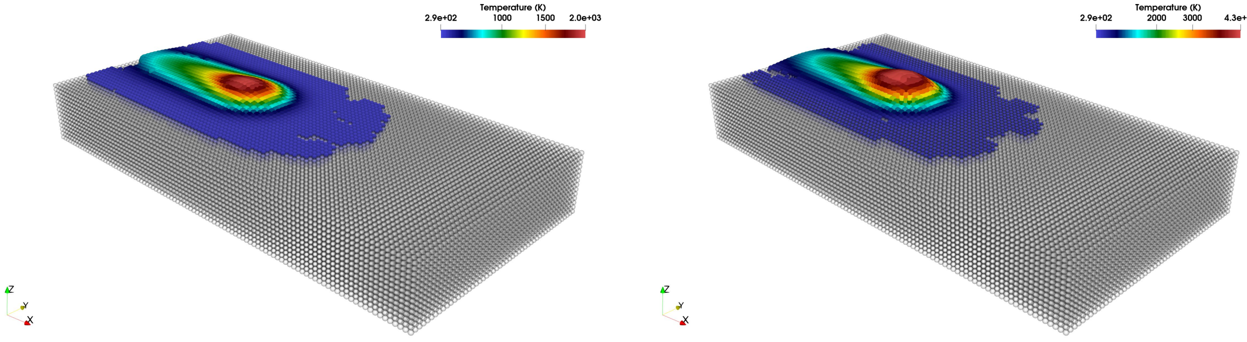
(d) Experiment 4: $q_0 = 100$ W/m^2 and $u_x = 1$ m/s^2.

Fig. 3—Temperature plots of the FOM numerical experiments at $t = 3.2\mu s$ corresponding to Table 1.

Table 3—FOM and ROM probe locations

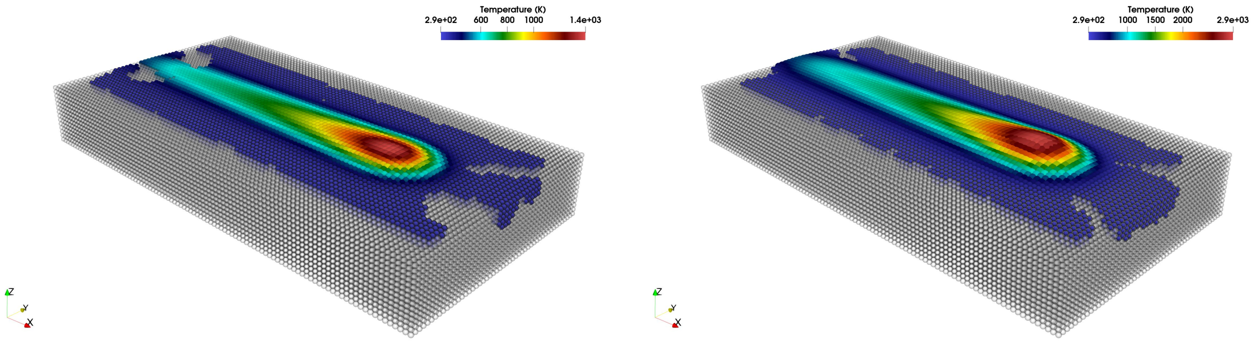
	x (mm)	y (mm)	z (mm)
Probe 1	0.364	-0.265	0.5
Probe 2	0.121	-0.429	0.5
Probe 3	0.323	-0.184	0.5
Probe 4	3.59	-0.061	0.5

and only the jet-colored particles are active in the ROM simulation, i.e., jet-colored particles are part of the hyper-reduced particle mesh. It is important to highlight that the sampling algorithm employed to define the hyper-reduced particle mesh selects particles along the traversal of the heat pass, which is the main source of error in the simulation. The outline of the sampled mesh is consistent with the intended algorithmic purpose, which employs a greedy algorithm to minimize residual error in a given ROM simulation. It is also important to highlight that the sampling algorithm selects particles within the bounds of the traveled path during FOM training runs. In other words, the hyper-reduced particle mesh for Experiment 1 and 2 extends only to about half the length of the specimen, while for Experiment 3 and 4 it extends the full length of the specimen.



(a) Experiment 1: $q_0 = 40$ W/m^2 and $u_x = 0.5$ m/s^2.

(b) Experiment 2: $q_0 = 100$ W/m^2 and $u_x = 0.5$ m/s^2.



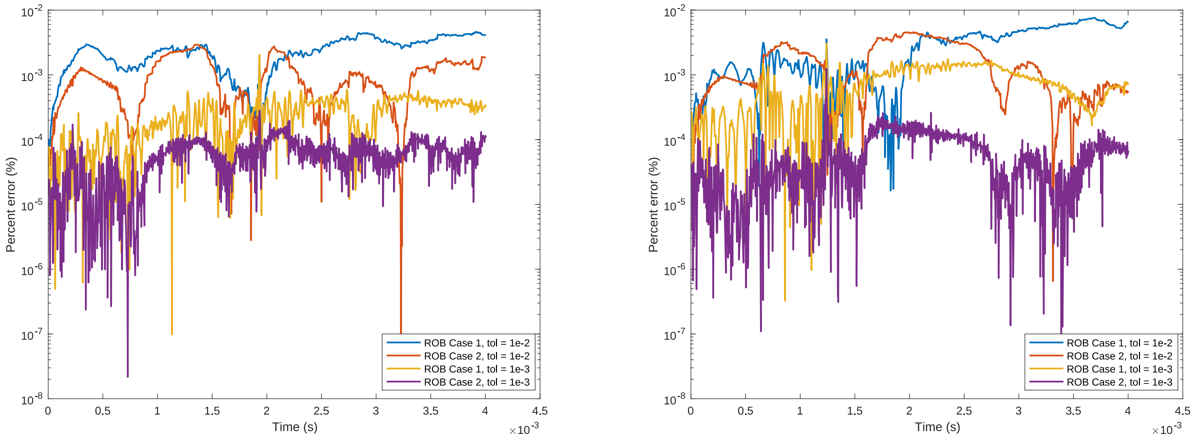
(c) Experiment 3: $q_0 = 40$ W/m^2 and $u_x = 1$ m/s^2.

(d) Experiment 4: $q_0 = 100$ W/m^2 and $u_x = 1$ m/s^2.

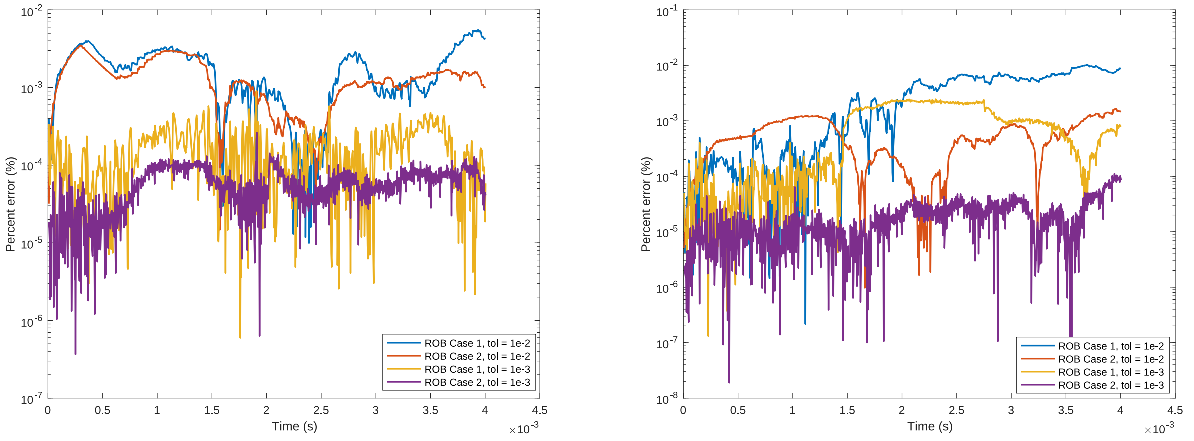
Fig. 4—Temperature plots of the ROM numerical experiments at $t = 3.2\mu s$ corresponding to Table 1 and all plots correspond to the ROB Case 1. Note that gray particles represent particles not used in the ROM and are not part of the hyper-reduced particle domain.

Percent error plots, quantified by Eq. 35, for all experiments are presented in Figs. 5-8. The error data shows the evolution of the ROM error over time for ROB Case 1 and Case 2. In addition, ROMs were deployed for a loose relative residual tolerance of tol = 10^{-2} to highlight the effectiveness of the ROM despite course error tolerances. Results show that tight tolerances reduce reproducible error by almost two orders of magnitude in some cases (a trivial conclusion). It is also seen that increasing the rank of the ROB improves error. Overall, error plots show at most a percent error of 0.68 % from Fig. 6(a) (Experiment 2; Probe 1;

ROB Case 2; tol = 10^{-2}), which indicate a reasonable reproducible error upper bound for the experiments in this study.



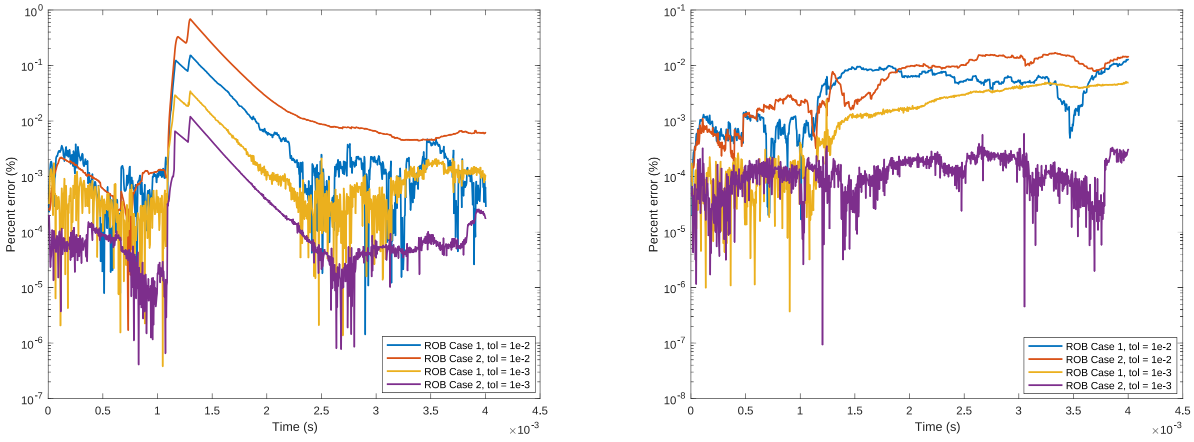
(a) Probe 1: $x = 0.364$mm, $y = -0.265$mm, $z = 0.5$mm. (b) Probe 2: $x = 0.121$mm, $y = -0.429$mm, $z = 0.5$mm.



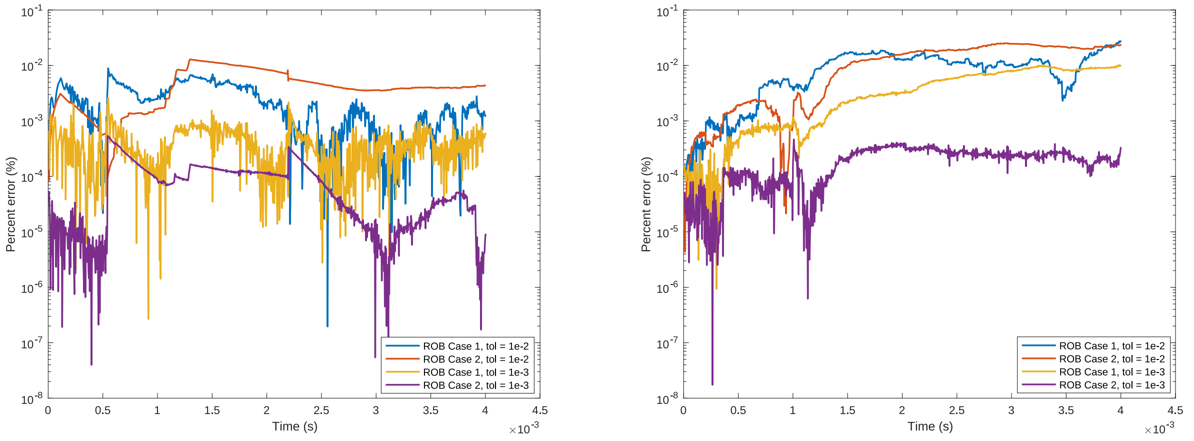
(c) Probe 3: $x = 0.323$mm, $y = -0.184$mm, $z = 0.5$mm. (d) Probe 4: $x = 3.59$mm, $y = -0.061$mm, $z = 0.5$mm.

Fig. 5—Error plots between FOM and ROM temperature probes for experiment 1.

Wall time results are presented in Fig. 9. All FOMs and ROMs were parallelized across 1, 16, 32, and 64 cores via OpenMP to assess performance with respect to wall time. The lowest wall time spent on FOMs occurs when 64 cores are deployed across all experiments. The lowest wall times spent on ROMs occurs at 16 cores for Experiments 1, 32 cores for Experiment 2, 16 and 32 cores (depending on the rank of the ROB) for Experiment 3, and 16 and 32 cores (depending on the rank of the ROB) for Experiment 4. It is important to highlight that the choice of hyper-parameters for ROMs and parallelization architecture

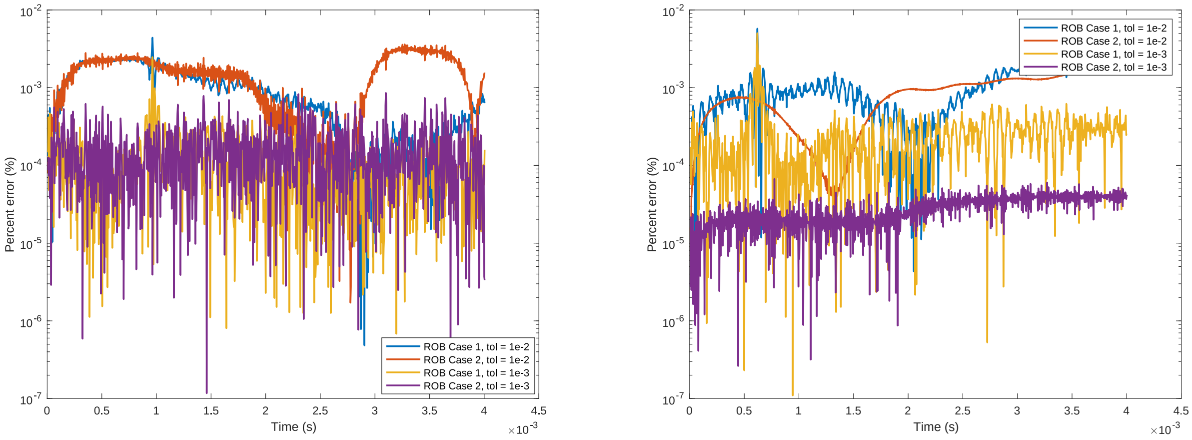


(a) Probe 1: $x = 0.364\text{mm}$, $y = -0.265\text{mm}$, $z = 0.5\text{mm}$. (b) Probe 2: $x = 0.121\text{mm}$, $y = -0.429\text{mm}$, $z = 0.5\text{mm}$.

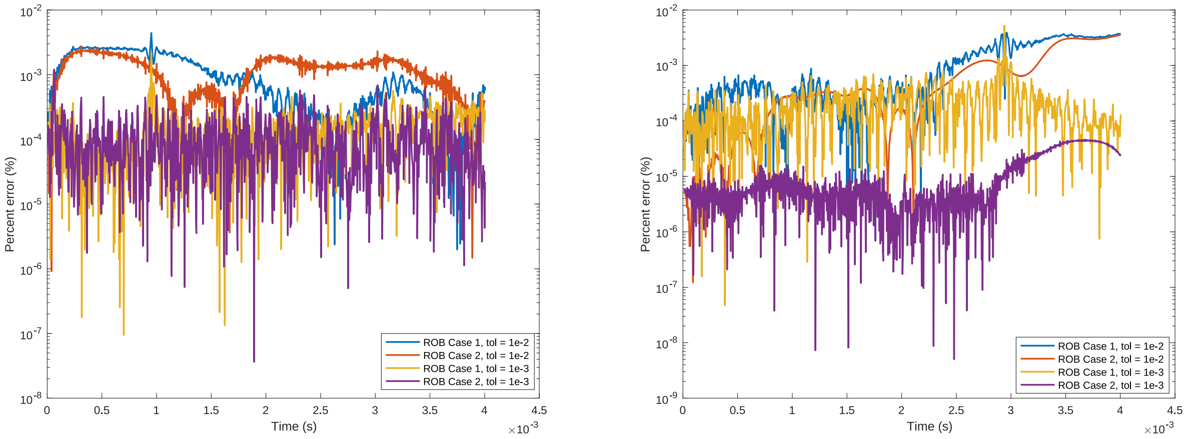


(c) Probe 3: $x = 0.323\text{mm}$, $y = -0.184\text{mm}$, $z = 0.5\text{mm}$. (d) Probe 4: $x = 3.59\text{mm}$, $y = -0.061\text{mm}$, $z = 0.5\text{mm}$.

Fig. 6—Error plots between FOM and ROM temperature probes for experiment 2.

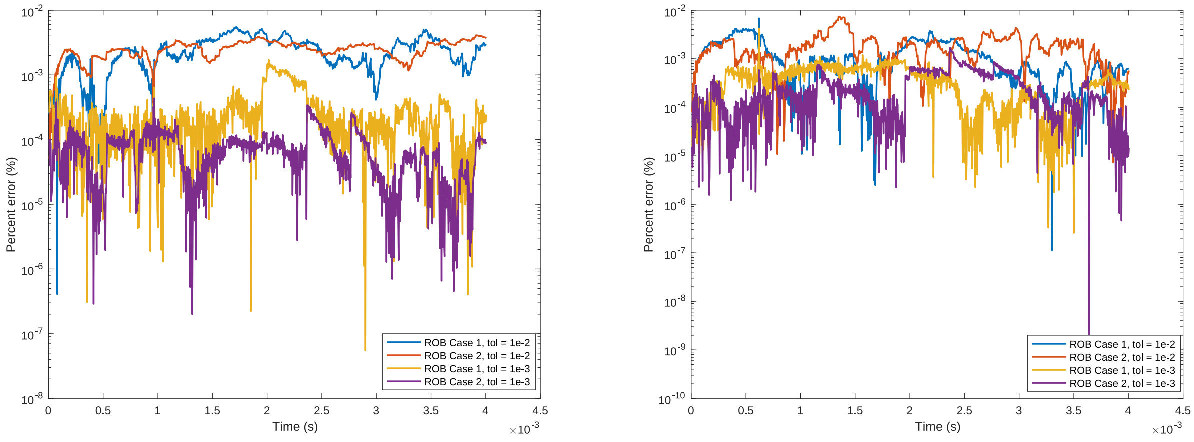


(a) Probe 1: $x = 0.364\text{mm}$, $y = -0.265\text{mm}$, $z = 0.5\text{mm}$.
(b) Probe 2: $x = 0.121\text{mm}$, $y = -0.429\text{mm}$, $z = 0.5\text{mm}$.



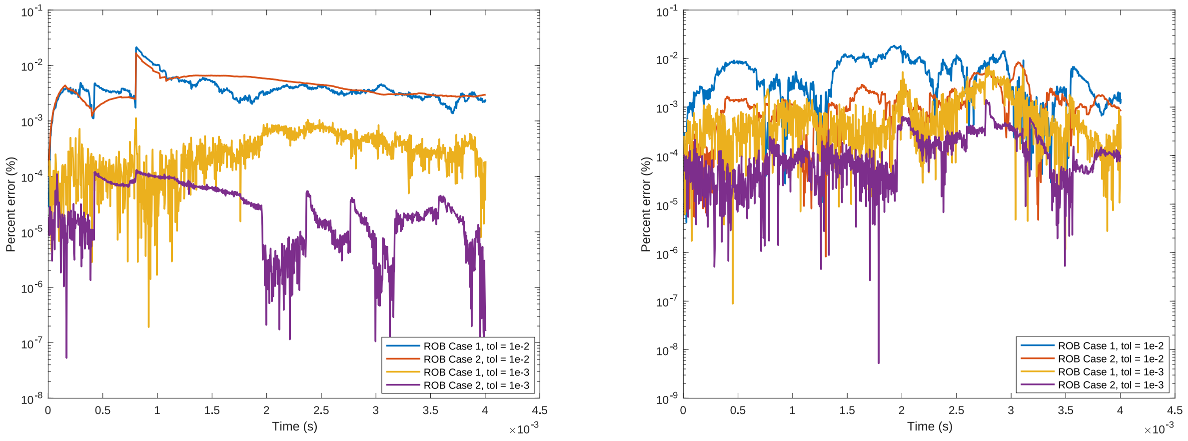
(c) Probe 3: $x = 0.323\text{mm}$, $y = -0.184\text{mm}$, $z = 0.5\text{mm}$.
(d) Probe 4: $x = 3.59\text{mm}$, $y = -0.061\text{mm}$, $z = 0.5\text{mm}$.

Fig. 7—Error plots between FOM and ROM temperature probes for experiment 3.



(a) Probe 1: $x = 0.364\text{mm}$, $y = -0.265\text{mm}$, $z = 0.5\text{mm}$.

(b) Probe 2: $x = 0.121\text{mm}$, $y = -0.429\text{mm}$, $z = 0.5\text{mm}$.



(c) Probe 3: $x = 0.323\text{mm}$, $y = -0.184\text{mm}$, $z = 0.5\text{mm}$.

(d) Probe 4: $x = 3.59\text{mm}$, $y = -0.061\text{mm}$, $z = 0.5\text{mm}$.

Fig. 8—Error plots between FOM and ROM temperature probes for experiment 4.

is important to achieve notable speed-up in simulation. It can be seen that a high rank ROB, as required for Experiments 2 and 4, significantly reduce any speed-up that can be realized with hyper-reduction or dimensionality reduction due to poor scaling of dense matrix operations. On the other hand, Experiments 1 and 3 show notable speed-up due to their significantly lower ROB rank. It is also important to show that enabling loose tolerances to achieve fewer Gauss–Newton iterations does not decrease wall time by significant amounts. The hyper-parameters which achieve the best wall time and error in their respective experiments are listed in Table 4.

Table 4—Wall-time savings compared to the FOM using 64 cores

	tolerance	ROM # of cores	SWT	SCPUH
Exp. 1, ROB Case 1	10^{-3}	16	1.7683	7.0320
Exp. 2, ROB Case 1	10^{-3}	32	0.5999	1.2321
Exp. 3, ROB Case 1	10^{-3}	32	3.4044	4.8428
Exp. 4, ROB Case 1	10^{-3}	16	0.3759	1.6499

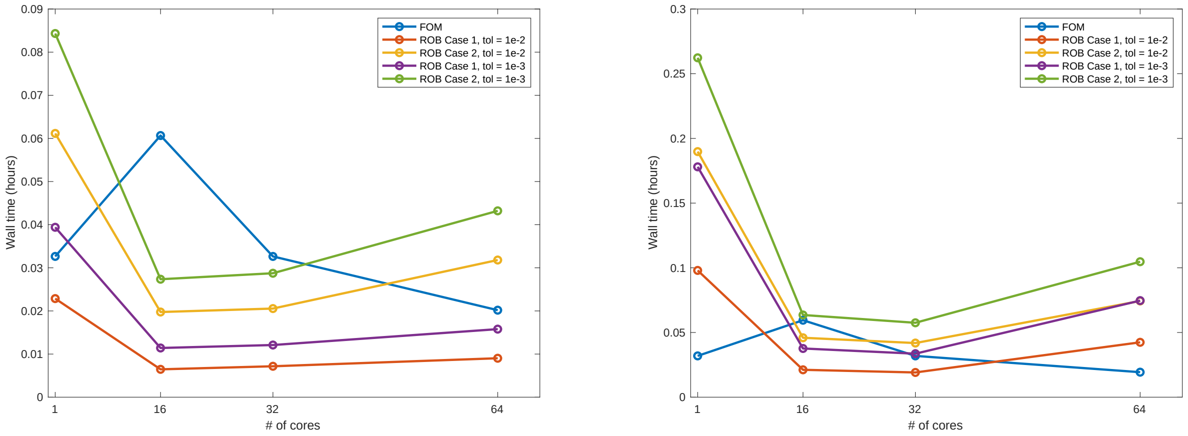
CPU hours spent on numerical experiments are presented in Fig. 10. All FOMs and ROMs were parallelized across 1, 16, 32, and 64 cores via OpenMP to assess the performance with respect to CPU hours. The lowest CPU hours spent on FOMs occurred when 16 cores were deployed across all experiments. The lowest CPU hours spent on ROMs occurred when only 1 core was deployed across all experiments. It is important to highlight that poor CPU hours scaling is poor likely due to the lack of parallelization in the least-squares solves. As mentioned earlier, the *Eigen* linear algebra library is employed to perform the least-square solves, but does not provide a parallel algorithm for least-squares solves with dense matrix operations. Future work will either include developing an in-house parallel MPI least-squares solver or employing ScaLAPACK [13]. The hyper-parameters which achieve the best CPU hours savings and error across experiments are listed in Table 5.

Table 5—CPU-hours savings compared to the FOM using 16 cores

	tolerance	ROM # of cores	SWT	SCPUH
Exp. 1, ROB Case 1	10^{-3}	1	0.5121	32.5907
Exp. 2, ROB Case 1	10^{-3}	1	0.1133	6.9445
Exp. 3, ROB Case 1	10^{-3}	1	1.2179	53.2810
Exp. 4, ROB Case 1	10^{-3}	1	0.0759	4.5826

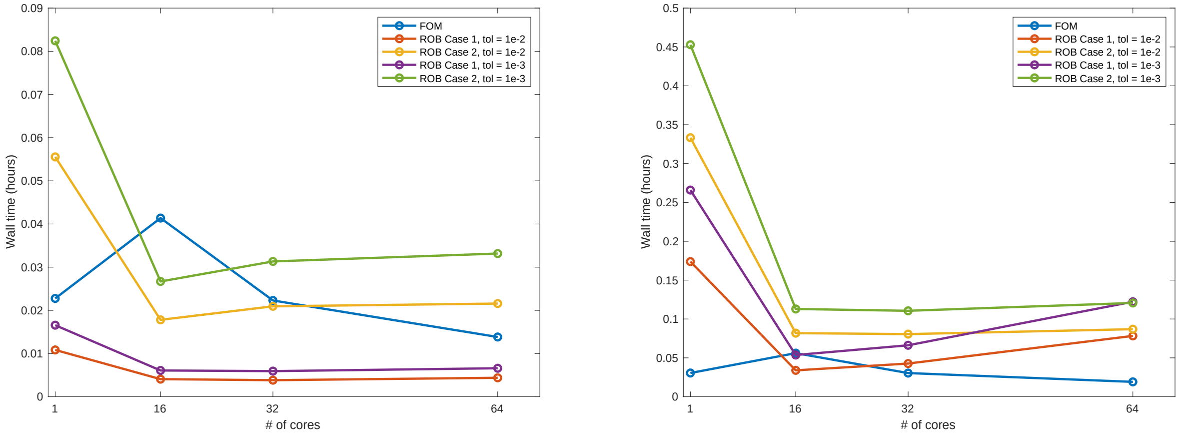
5. CONCLUSIONS AND OUTLOOK

The LSPG ROM framework equipped with GNAT hyper-reduction was deployed on a heat-deposition temperature-dependent problem with SPH. Error analyses show that the ROM framework employed is capable of reproducing FOM numerical experiments with considerable accuracy. The ROM framework exhibited wall time and CPU savings. However, it is important to note that computational savings achieved by the ROM were not of considerable levels with respect to those often reported in the literature for the LSPG/GNAT framework in fluid dynamics [7], and at times the ROM was even slower than the FOM. To the best of the authors' knowledge, there are three possible points prohibiting the current ROM from achieving considerable wall time and CPU savings that match or exceed results in the literature for heat-deposition problems: 1) The Kolmogorov $N-$width limitation of the affine POD subspace employed requires high ranks to enable a low number of iterations to achieve convergence and lower error. However, a high rank POD subspace



(a) Experiment 1 wall time.

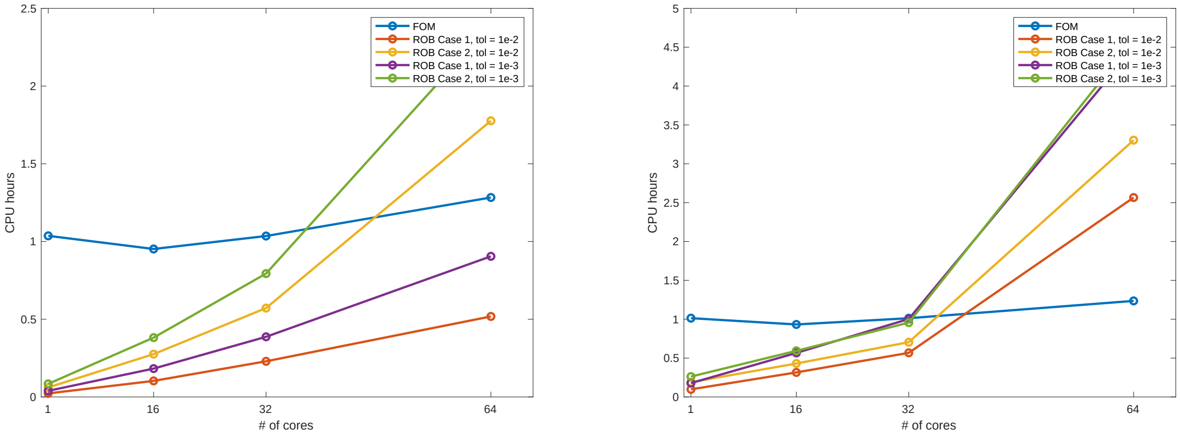
(b) Experiment 2 wall time.



(c) Experiment 3 wall time.

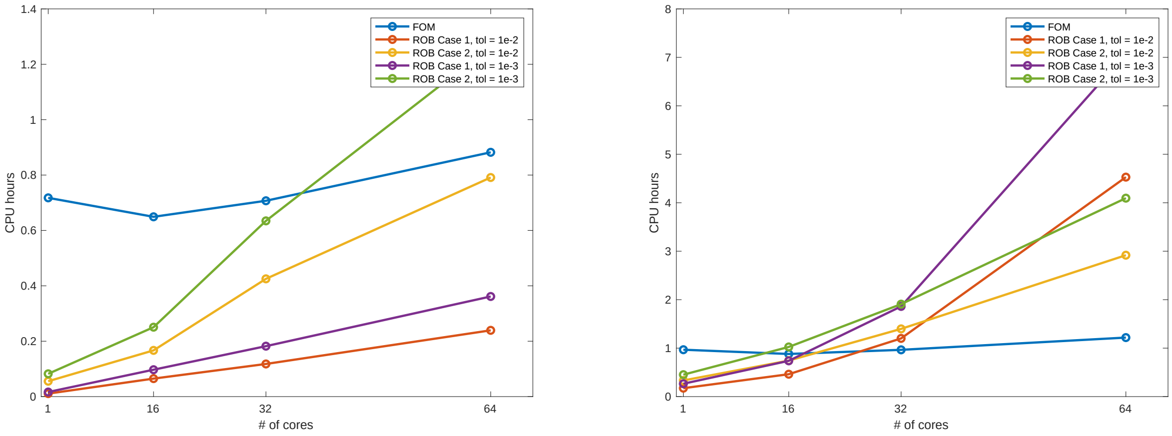
(d) Experiment 4 wall time.

Fig. 9—Wall time for FOM and ROMs across all four experiments.



(a) Experiment 1 CPU hours.

(b) Experiment 2 CPU hours.



(c) Experiment 3 CPU hours.

(d) Experiment 4 CPU hours.

Fig. 10—CPU hours spent on FOM and ROMs across all four experiments.

severely limits any computational savings due to poor scaling in solving the dense matrix least-squares problem presented by Eq. 32. 2) The *Eigen* linear algebra library employed does not include a parallel dense least-squares solver, which prohibits the ROM framework from realizing full parallelization benefits. 3) Meshless nonlocal numerical frameworks, such as SPH, have a much higher neighbor count than their local method counterparts, i.e., finite elements, finite volumes, finite differences, to compute solutions. Therefore, nonlocal methods cannot achieve the same levels of sparsity in the sampled mesh as local methods for which the GNAT hyper-reduction was originally built. As a result, the computational savings that hyper-reduction enables in local methods is not directly transferred to nonlocal methods, such as SPH.

The three possible limitations observed and listed above are currently being addressed as an extension to the work in the Karles Fellowship. It is important to note that the first observation listed above regarding the Kolmogorov $N-$width limitation of the affine POD subspaces was recently addressed by Lee and Carlberg [8] within the context of the LSPG/GNAT framework via nonlinear subspaces. As a result, we are currently investigating the application of nonlinear subspaces to address the limitation posed by the Kolmogorov $N-$width in the current heat-deposition problem. Current work is also underway to restructure the in-house ROM code to improve its parallelization architecture and to employ third-party libraries to enable parallel dense matrix least-squares solves. On a final note, it is important to highlight that a major contribution of the work performed during the Karles Fellowship addressed point 3) listed above, i.e., overcoming nonlocality in hyper-reduction of the LSPG/GNAT framework. As previously noted, nonlocal numerical frameworks' sampled particles require information from more neighbors than local methods, thereby limiting the levels of sparsity that could be achieved for considerable computational savings. To address the limiting sparsity issues associated with nonlocal methods, the *projection-tree reduced-order model* (PTROM) was developed as an auxiliary methodology under the Karles Fellowship [1]. The theoretical framework of the PTROM was developed on a simplistic two-dimensional nonlocal framework for fluid dynamics problems, but was generalized for the purposes of transferring its framework to SPH and heat-deposition problems. Current work is underway to deploy the PTROM on the SPH heat-deposition problem to enable improved computational savings. However, because the PTROM is an auxiliary method under the Karles Fellowship a detailed description of the method is out of the scope of the current report. Nevertheless, a brief description and overview of the method are presented in Appendix A but the interested reader can refer to the preprint under review [1].

In summary, the two main contributions of the work conducted under the Karles Fellowship include 1) the development of the novel PTROM and 2) the first reported application of the LSPG/GNAT framework on SPH heat-deposition problems. The resulting work of the fellowship has paved the way for natural extensions of the work presented herein. Examples include, but are not limited to, deploying the PTROM across multiphysics applications outside of additive manufacturing, developing new dimensional reduction techniques that include mass-deposition in additive manufacturing, and employing nonlinear subspace methods.

REFERENCES

1. S. N. Rodriguez, A. P. Iliopoulos, K. T. Carlberg, S. L. Brunton, J. C. Steuben, and J. G. Michopoulos, "Projection-tree reduced order modeling for fast N-body computations," *arXiv preprint arXiv:2103.01983* (2021).

2. D. Violeau, *Fluid mechanics and the SPH method: theory and applications* (Oxford University Press, 2012).

3. G. R. Liu and M. B. Liu, *Smoothed particle hydrodynamics: a meshfree particle method* (World scientific, 2003).

4. K. Fraser, L. St-Georges, and L. Kiss, "Adaptive thermal boundary conditions for smoothed particle hydrodynamics," Proceedings of the Proceedings of the 14th Internation LS-DYNA Conference, Detroit, MI, USA, 2016, pp. 12–14.

5. J. Michopoulos, J. Steuben, A. Birnbaum, A. Iliopoulos, J. Aroh, A. Rollett, and B. Gould, "Morphological Analysis of 316L Laser Powder Bed Fusion Melt-Pool via the Enriched Analytical Solution Method," Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, volume 83983 (American Society of Mechanical Engineers), 2020, p. V009T09A021.

6. K. Carlberg, C. Bou-Mosleh, and C. Farhat, "Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations," *International Journal for numerical methods in engineering* **86**(2), 155–181 (2011).

7. K. T. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, "The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows," *Journal of Computational Physics* **242**, 623–647 (2013).

8. K. Lee and K. T. Carlberg, "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," *Journal of Computational Physics* **404**, 108973 (2020).

9. J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

10. R. Everson and L. Sirovich, "Karhunen–Loeve procedure for gappy data," *JOSA A* **12**(8), 1657–1664 (1995).

11. K. Carlberg, R. Tuminaro, and P. Boggs, "Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics," *SIAM Journal on Scientific Computing* **37**(2), B153–B184 (2015).

12. G. Guennebaud, B. Jacob, et al., "Eigen v3," http://eigen.tuxfamily.org, 2010.

13. L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997), ISBN 0-89871-397-8 (paperback).

This page intentionally left blank

Appendix A

PROJECTION-TREE REDUCED-ORDER MODELING IN FLUID DYNAMICS

The projection-tree reduced-order modeling (PTROM) framework is a ROM method that addresses the lack of sparsity in nonlocal numerical frameworks that prohibits them from achieving considerable computational savings under hyper-reduction in ROMs. The PTROM framework takes motivation from the LSPG/GNAT ROM methodology and hierarchical decomposition to achieve unprecedented computational savings for nonlocal ROM methods.

From an overhead perspective nonlocal methods model a physical system of partial differential equations (PDEs) by discretizing a continuum as a series of adaptive, reactive, and interactive collocation points, often known as particles. The discretized space consists of a set of target and source particles that interact with each other and behave according to the governing set of PDEs. An example of a nonlocal numerical framework is the free-vortex wake method (FVM), which employs the Biot-Savart kernel to model the vorticity and wake aerodynamics in an aerospace structure. Figure A1 illustrates an example of the FVM (top of image), the nonlocal interactions of vortex particles (bottom right), and an illustration of the resulting two-dimensional dense nonlocal pairwise matrix (bottom left).

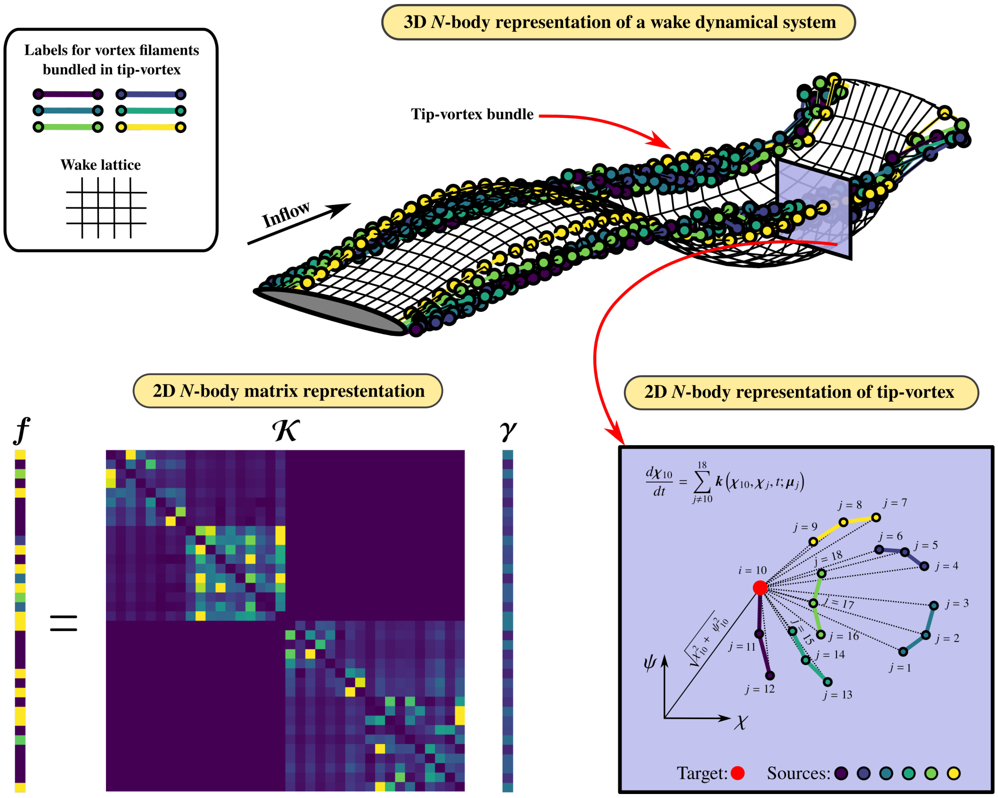


Fig. A1—Image depicting nonlocal pairwise interaction of a tip vortex shed off of an elliptical wing. Image from Rodriguez et al. [1].

As discussed in the main sections of this report, nonlocal methods are faced with a lack of sparsity when restructured in a ROM framework that incorporates hyper-reduction. The lack of sparsity in hyper-reduction of nonlocal methods stems from a relatively high number of sources per target particle, which results in the retention of a high number of particles in the sampled hyper-reduced particle mesh deployed for online simulations. In some instances, as is the case when employing the Biot–Savart kernel in fluid dynamics, the number of sources is equal to the total number of particles in the domain. These nonlocal interactions result in $O(N^2)$ operational count complexity (OCC) for FOM computations and can result in $O(N)$ OCC when naïvely deploying LSPG/GNAT on the Biot–Savart kernel in fluid dynamics [1]. An overhead view of how the PTROM reduces the number sources in a hyper-reduced mesh for nonlocal frameworks is now presented.

A.1 Overcoming nonlocality via hierarchical decomposition

The PTROM is based on constructing separate low-dimensional bases for target particles and source particles. For target particles, constructing the low-dimensional basis is equivalent to the procedure presented in Section 3.1. For source particles, however, constructing the low-dimensional basis is multi-tiered, as depicted in Fig. A2. To begin, the POD modes obtained for targets are weighed by their corresponding singular values and super-imposed to form a space (called a cluster space [1]) on which clustering of the weighted super-imposed POD modes can be performed. Next, hierarchical decomposition is performed on the cluster space, and agglomerated particles and the hierarchical data structure are recorded. Finally, using the information recorded for agglomerated particles in cluster space and the hierarchical data, each POD mode is clustered in the row-wise direction to reduce the number of neighbor particles queried in the nonlocal ROM framework, as depicted in Fig. A3. The interested reader is referred to [1] for a more detailed discussion on how to construct and deploy the PTROM framework.

A.2 Future work with projection-tree reduced-order modeling

The PTROM framework was tested on a series of vortex dynamics simulations, which resulted in a speed up factor of 2000, i.e., the PTROM was 2000 times faster than the FOM, and where the corresponding reproductive error reached sub-0.1% error [1]. Results presented by the PTROM show promise in the framework, but there are many improvements to be made before the framework can be made available for complex multiphysics problems, such as the heat-deposition problem presented in the main sections of this report. Some examples of improvements to the framework include substituting the affine POD subspace with a nonlinear subspace approach, incorporate parallelization for dense matrix least-squares solves, and identifying more memory efficient hierarchical decomposition methods.

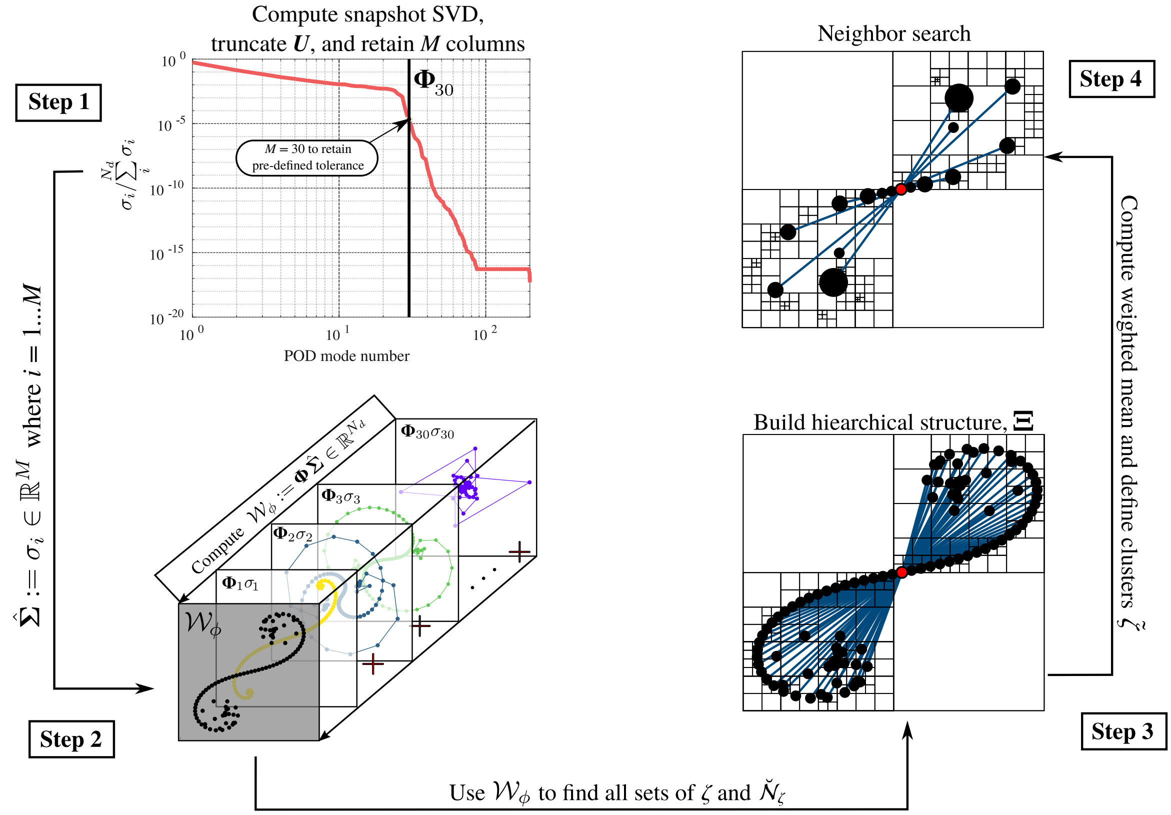


Fig. A2—Procedure employed to perform hierarchical decomposition on target POD modes to define a low-dimensional basis for source particles . Image from Rodriguez et al. [1].

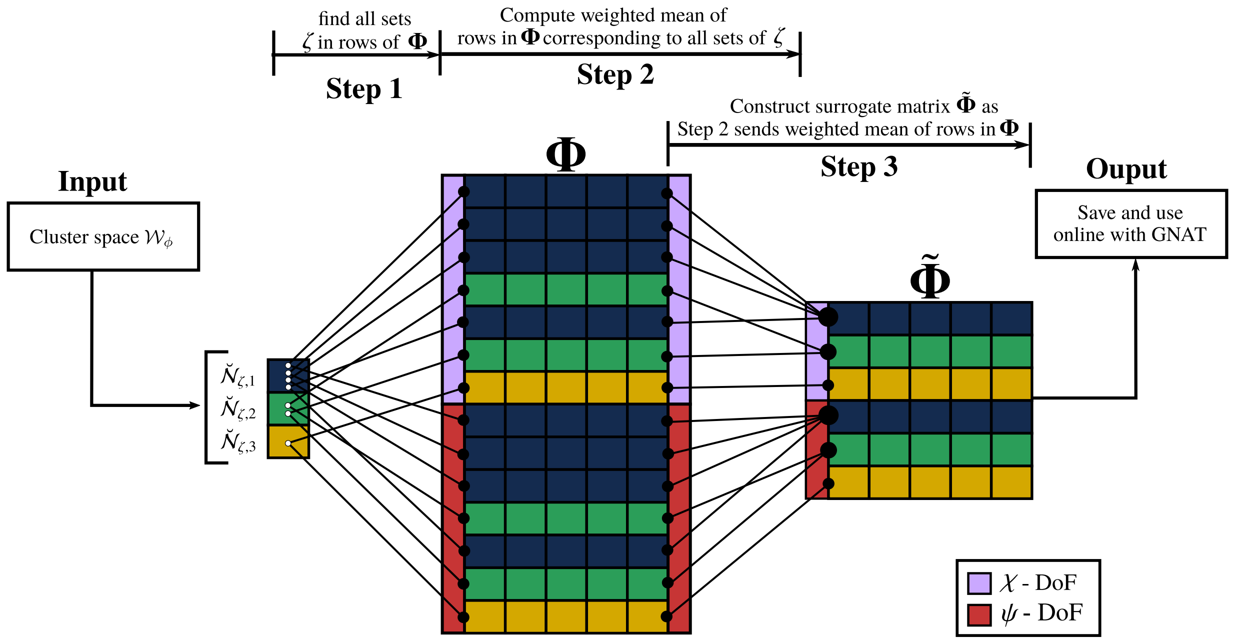


Fig. A3—Clustering of the target POD matrix according the hierarchical decomposition data structure generated to form a low-dimensional bases for source particles. Image from Rodriguez et al. [1].