A Data-Driven, Non-Linear, Parameterized Reduced Order Model of Metal 3D Printing

Aaron L. Brown
Department of Mechanical Engineering, Stanford University
Stanford, CA, 94305
abrown97@stanford.edu

Eric B. Chin
Lawrence Livermore National Laboratory
Livermore, CA, 94550
chin23@llnl.gov

Youngsoo Choi
Lawrence Livermore National Laboratory
Livermore, CA, 94550
choi15@llnl.gov

Saad A. Khairallah
Lawrence Livermore National Laboratory
Livermore, CA 94550
khairallah1@llnl.gov

Joseph T. McKeown
Lawrence Livermore National Laboratory
Livermore, CA 94550
mckeown3@llnl.gov

Abstract

Directed energy deposition (DED) is a promising metal additive manufacturing technology capable of 3D printing metal parts with complex geometries at lower cost compared to traditional manufacturing. The technology is most effective when process parameters like laser scan speed and power are optimized for a particular geometry and alloy. To accelerate optimization, we apply a data-driven, parameterized, non-linear reduced-order model (ROM) called Gaussian Process Latent Space Dynamics Identification (GPLaSDI) to physics-based DED simulation data. With an appropriate choice of hyperparameters, GPLaSDI is an effective ROM for this application, with a worst-case error of about 8% and a speed-up of about $1,000,000$x with respect to the corresponding physics-based data.

1 Introduction

In Directed Energy Deposition (DED) additive manufacturing, or metal 3D printing, metal powder is fed through a nozzle and directed down to a build plate, where a laser fuses the incoming powder into the previous layer or substrate. An important goal in DED is to optimize process parameters such as laser power and scan speed to control the temperature history and prevent defects in prints. Currently, the additive manufacturing community resorts to a trial and error approach involving numerous experiments, each of which can be time-consuming and costly. This process must be repeated for any new printing material, and the results may vary depending on the complexity of the part or the 3D printer itself.

Previous approaches have attempted analytic solutions to this problem, which in its simplest form asks the temperature field for a moving, distributed heat source [1–3]. However, analytic approaches do not offer the flexibility to handle irregular domains and complicated initial and boundary conditions found in many additive manufacturing applications. Such flexibility is afforded by high-fidelity finite element-based models of DED [4], but simulations can take weeks to run, making them unsuitable for parameter optimization.
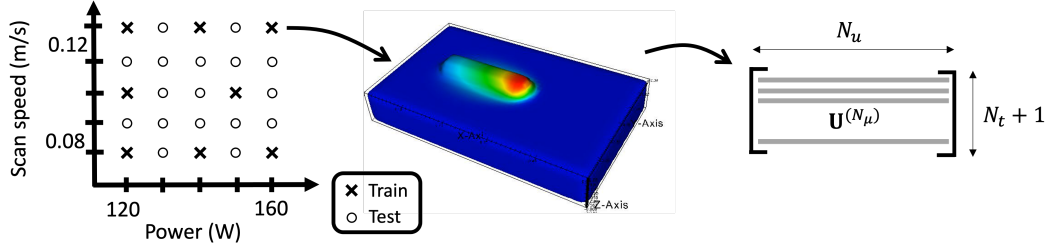
Figure 1: Schematic of dataset generation. Single-track DED simulations were generated for parameters (laser power and laser scan speed) on a 5x5 grid. Temperature data was extracted for each simulation and arranged into a training snapshot matrix.

Our approach is to use reduced order models (ROMs), informed by simulations or experiments. ROMs can be orders of magnitude faster with a marginal reduction in accuracy and have been successfully applied to various physics problems [5–16]. They may fully replace full-order simulations or experiments or be used as surrogate models such as in the Surrogate Management Framework [17]. In this work, we apply the recently-developed ROM known as Gaussian Process Latent Space Dynamics Identification (GPLaSDI) [18] to parameterized DED simulation data to predict, for given process parameters, the time-dependent temperature field, which may then be used as an input to a microstructure evolution model. A major advantage of GPLaSDI is its use of an autoencoder to spatially compress data in a nonlinear manner, which, compared to linear compression methods such as the singular value decomposition, can better capture the advection-like behavior introduced by the motion of the DED print head.

2 Methods

2.1 DED simulation data

In this work, we consider data obtained from DED simulations performed in ALE3D [19]. The data was parameterized with respect to two important parameters in DED: laser power P and laser scan speed S. Specifically, we generated single-track DED simulations for each sample on a 5x5 grid in parameter space – $P \in [120, 130, 140, 150, 160]$ W, $S \in [0.08, 0.09, 0.10, 0.11, 0.12]$ m/s. Each simulation modeled the deposition of a single track of titanium alloy Ti-6Al-4V (Ti64) at a rate of 5 g/min over a length of 2.5mm. The computational mesh consisted of $N_u = 25,990$ nodes, and each simulation was run between 100 and 150 ms, from which we extract $N_t + 1 = 101$ evenly spaced time steps for our dataset. Each simulation took on average 2.5 hours on a single core of an Intel(R) Xeon(R) Platinum 8479 CPU. From this dataset, we select N_μ samples for the training set; how the training set is chosen will be described later. The remainder are part of the test set.

Let $\boldsymbol{\mu}^{(i)} = (P^{(i)}, S^{(i)})$ be the ith parameter vector in the training set. The quantity of interest from these simulations is the time-dependent temperature field, u. Let $\mathbf{u}_n^{(i)} \in \mathbb{R}^{N_u}$ be the snapshot vector of temperature data for all nodes at time step n for parameter vector $\boldsymbol{\mu}^{(i)}$, and let $\mathbf{U}^{(i)} = \left[\mathbf{u}_0^{(i)}, ..., \mathbf{u}_{N_t}^{(i)}\right] \in \mathbb{R}^{(N_t+1) \times N_u}$ be the data matrix for all nodes and all time steps for parameter vector $\boldsymbol{\mu}^{(i)}$. Combining all these matrices together, we define a 3rd order tensor $\mathbf{U} = \left[\mathbf{U}^{(1)}, ..., \mathbf{U}^{(N_\mu)}\right] \in \mathbb{R}^{N_\mu \times (N_t+1) \times N_u}$, which constitutes the training dataset in this work. See Figure 1 for a visual description of this dataset.

2.2 GPLaSDI

GPLaSDI is a parameterized, data-driven reduced order modeling method which consists of three main features: 1) an autoencoder to learn a nonlinear spatial compression of data to a latent space; 2) dynamics identification within the latent space using Sparse Identification of Nonlinear Dynamics (SINDy) [20]; 3) Gaussian process interpolation of latent space dynamical systems over the parameter

space to achieve parameterization. Here, we briefly describe GPLaSDI. More details may be found in [18].

The autoencoder is composed of two neural networks: an encoder ϕ_e parameterized by weights and biases θ_e and a decoder ϕ_d parameterized by weights and biases θ_d. The encoder spatially compresses the full space data matrix $\mathbf{U} \in \mathbb{R}^{N_\mu \times (N_t+1) \times N_u}$ to a latent space data matrix $\mathbf{Z} \in \mathbb{R}^{N_\mu \times (N_t+1) \times N_z}$, where $N_z \ll N_u$. Conversely, the decoder spatially decompresses $\mathbf{Z}$ into a reconstructed full space representation $\widehat{\mathbf{U}} \in \mathbb{R}^{N_\mu \times (N_t+1) \times N_u}$. We define an autoencoder reconstruction loss as the mean squared error between $\mathbf{U}$ and $\widehat{\mathbf{U}}$,

$$\mathcal{L}_{AE}(\theta_e, \theta_d) = ||\mathbf{U} - \widehat{\mathbf{U}}||_2^2. \tag{1}$$

$\mathbf{Z}$ may be interpreted as a collection of latent space trajectories $\mathbf{Z} = \left[\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(N_\mu)} \right]$, where $\mathbf{Z}^{(i)} = \left[\mathbf{z}_0^{(i)}, ..., \mathbf{z}_{N_t}^{(i)} \right]$ corresponds to parameter vector $\boldsymbol{\mu}^{(i)}$. In the SINDy interpretation, each trajectory is assumed to come from a dynamical system,

$$\frac{d\mathbf{z}}{dt} = \mathbf{f}^{(i)}(\mathbf{z}); \quad \mathbf{z}(0) = \mathbf{z}_0^{(i)}. \tag{2}$$

The velocity $\mathbf{f}^{(i)}(\mathbf{z})$ is approximated by a linear combination of user-defined "library" functions of $\mathbf{z}$,

$$\mathbf{f}^{(i)}(\mathbf{z}) = \boldsymbol{\Phi}(\mathbf{z}) \cdot \boldsymbol{\Xi}^{(i)T}, \tag{3}$$

where $\boldsymbol{\Phi}(\mathbf{z})$ is a matrix of candidate terms involving $\mathbf{z}$, and $\boldsymbol{\Xi}^{(i)}$ is an (as yet unknown) matrix of coefficients. In this work, we limit the library to constant and linear terms, thereby approximating the latent space dynamics with an affine ODE system. $\boldsymbol{\Xi}^{(i)}$ is determined by first computing latent space velocities $\dot{\mathbf{Z}}^{(i)}$ using a first-order finite difference scheme on $\mathbf{Z}^{(i)}$, then solving the linear regression problem

$$\dot{\mathbf{Z}}^{(i)} = \boldsymbol{\Phi}(\mathbf{Z}^{(i)}) \cdot \boldsymbol{\Xi}^{(i)T}. \tag{4}$$

Note that separate $\boldsymbol{\Xi}^{(i)}$ are learned for each $\boldsymbol{\mu}^{(i)}$. The collection of dynamics coefficient matrices is denoted $\boldsymbol{\Xi} = \left[\boldsymbol{\Xi}^{(1)}, ..., \boldsymbol{\Xi}^{(N_\mu)} \right]$. Associated with the dynamics identification is a loss term

$$\mathcal{L}_{SINDy}(\boldsymbol{\Xi}) = \frac{1}{N_\mu} \sum_{i=1}^{N_\mu} ||\dot{\mathbf{Z}}^{(i)} - \boldsymbol{\Phi}(\mathbf{z}) \cdot \boldsymbol{\Xi}^{(i)T}||_2^2. \tag{5}$$

The model is trained by minimizing the following total loss function,

$$\mathcal{L}(\theta_e, \theta_d, \boldsymbol{\Xi}) = \beta_1 \mathcal{L}_{AE}(\theta_e, \theta_d) + \beta_2 \mathcal{L}_{SINDy}(\boldsymbol{\Xi}) + \beta_3 ||\boldsymbol{\Xi}||_2^2, \tag{6}$$

where the third term is an important regularization term, and β_1, β_2, and β_3 are weighting hyperparameters.

The trained model can then be used to predict a time-dependent temperature field, given a new parameter vector $\boldsymbol{\mu}^{(*)}$ and a full space initial condition $\mathbf{u}_0^{(*)}$, which in our case is always uniform room temperature. First, $\mathbf{u}_0^{(*)}$ is compressed to the latent space using the encoder ϕ_e, yielding a latent space initial condition $\mathbf{z}_0^{(*)}$. Then, a predicted latent space trajectory $\widetilde{\mathbf{Z}}^{(*)}$ is obtained by integrating in time, where the coefficient matrix $\boldsymbol{\Xi}^{(*)}$ is obtained by interpolating the $\boldsymbol{\Xi}^{(i)}$ from training samples over the parameter space. Gaussian process interpolation is used, which has the additional benefit of built-in uncertainty quantification. Finally, the decoder ϕ_d is used to generate the predicted full space trajectory $\widetilde{\mathbf{U}}^{(*)}$. Note that due to the uncertainty in $\boldsymbol{\Xi}^{(*)}$, multiple predicted trajectories produce slightly different results, and the mean and variance may be computed.

Another feature of GPLaSDI is a variance-based greedy sampling algorithm. During training, samples may be added to the training set every N_{greedy} epochs. If there are currently N_μ samples in the training set, the additional $(N_\mu + 1)^{th}$ sample is chosen as that with the largest prediction uncertainty, as measured by the variance of the full order prediction.

3 Results

The following hyperparameters were found to yield satisfactory preliminary results. The encoder architecture follows a 25,990-1000-200-50-20-5 structure, comprising four hidden linear layers of size 1000, 200, 50, and 20, and a latent space of dimension 5. The decoder is symmetric with respect to the encoder. All layers except for the last in the encoder and decoder are connected with Softplus activation functions. The model is trained for $N_{epoch} = 1,000,000$ epochs using the Adam optimizer with a learning rate of $\alpha = 10^{-4}$. The training dataset begins with $N_\mu = 4$ samples corresponding to the corners of the parameter grid. Starting with corner points is an arbitrary but intuitive choice, since it ensures that predictions for all other points are interpolations rather than extrapolations. A new sample is added to the training set every $N_{greedy} = 200,000$ epochs. For the loss weights, we choose $\beta_1 = \beta_2 = 1$. The performance depends strongly on the value of β_3, and we compare results for β_3 of 10^{-3} and 10^1.

To assess model performance, we use the maximum relative error, defined as

$$e(\widetilde{\mathbf{U}}^{(*)}, \mathbf{U}^{(*)}) = \max_n \left(\frac{||\widetilde{\mathbf{u}}_n^{(*)} - \mathbf{u}_n^{(*)}||_2}{||\mathbf{u}_n^{(*)}||_2} \right). \tag{7}$$

In Figure 2, the left column shows the GPLaSDI results for $\beta_3 = 10^{-3}$, while the right column shows results for $\beta_3 = 10^1$. The top row shows the maximum relative error over the parameter grid, with training samples indicated by a black border. Clearly, β_3 is a very important hyperparameter, with extremely high errors for $\beta_3 = 10^{-3}$ and only 8.11% test error in the worst case for $\beta_3 = 10^1$. In the second row, we compare the autoencoder projection errors vs. time, obtained by encoding and decoding the ground truth test data, for a typical test sample $\boldsymbol{\mu}^{(*)} = (140\text{W}, 0.1\text{m/s})$. For both values of β_3, projection errors are low, indicating that the autoencoder is not responsible for the vast difference in performance. Rather, the difference stems from disparate latent space dynamics behavior. In the third row, we show the latent space trajectories – both the true $\mathbf{Z}^{(*)}$ and the (multiple) approximated $\widetilde{\mathbf{Z}}^{(*)}$ – for the same typical test sample. For $\beta_3 = 10^{-3}$, the approximated trajectories are extremely unstable and blow up, while for $\beta_3 = 10^1$, the approximated trajectories match the true trajectory well. The stark difference in behavior arises from differences in the character of the dynamical system. In the fourth row, we visualize the same trajectories in a projected phase space, plotting the second latent space variable z_1 (orange lines) vs. the first latent space variable z_0 (blue lines). In addition, the arrows indicate the identified latent space velocity $\mathbf{f}^{(*)}(\mathbf{z}) = \boldsymbol{\Phi}(\mathbf{z}) \cdot \boldsymbol{\Xi}^{(*)T}$ projected onto the z_0-z_1 plane. Arrows are drawn at points on and slightly offset from the true trajectory. As shown, the latent space velocity field for $\beta_3 = 10^{-3}$ is irregular and varies rapidly, promoting instability, while the velocity field for $\beta_3 = 10^1$ is very smooth, resulting in stable and accurate latent space trajectories. We note that increasing β_3 promotes smoothness because it tends to reduce the coefficients in $\boldsymbol{\Xi}^{(i)}$, which is closely related to $\nabla \mathbf{f}^{(i)}(\mathbf{z})$. With respect to computational cost, each GPLaSDI prediction takes on average 7.84ms on an IBM Power9 CPU core with one NVIDIA V100 GPU, resulting in a roughly $1,000,000$ times speed-up compared to the physics-based simulation.

In GPLaSDI test cases [18], adequate performance was achieved with β_3 values of 10^{-5} or 10^{-6}, much smaller than that required in our study. We suspect that the parameter grid density plays a important role, given the 21×21 grid in [18] versus our 5×5 grid over a similar sized region of parameter space. From Figure 2, we note that despite instability in identified dynamics, the poor-performing $\beta_3 = 10^{-3}$ model has relatively low error on training samples, leading us to attribute poor test sample predictions to the *interaction* of unstable dynamics and interpolation over the parameter space. We propose that increasing grid density mitigates instability on test samples by reducing the interpolation challenge. Thus, our investigation of the role of β_3 may introduce a way to retain model performance with fewer training samples. Other factors, such as underlying physics and training data timestep size, may also influence the required β_3 size.

4 Conclusion

In this work, we applied GPLaSDI to time-dependent temperature data from single-track DED simulations for varying laser power and scan speed. With appropriate choice of hyperparameters, GPLaSDI is an effective reduced order model for this data, achieving $1,000,000$x speed-up compared
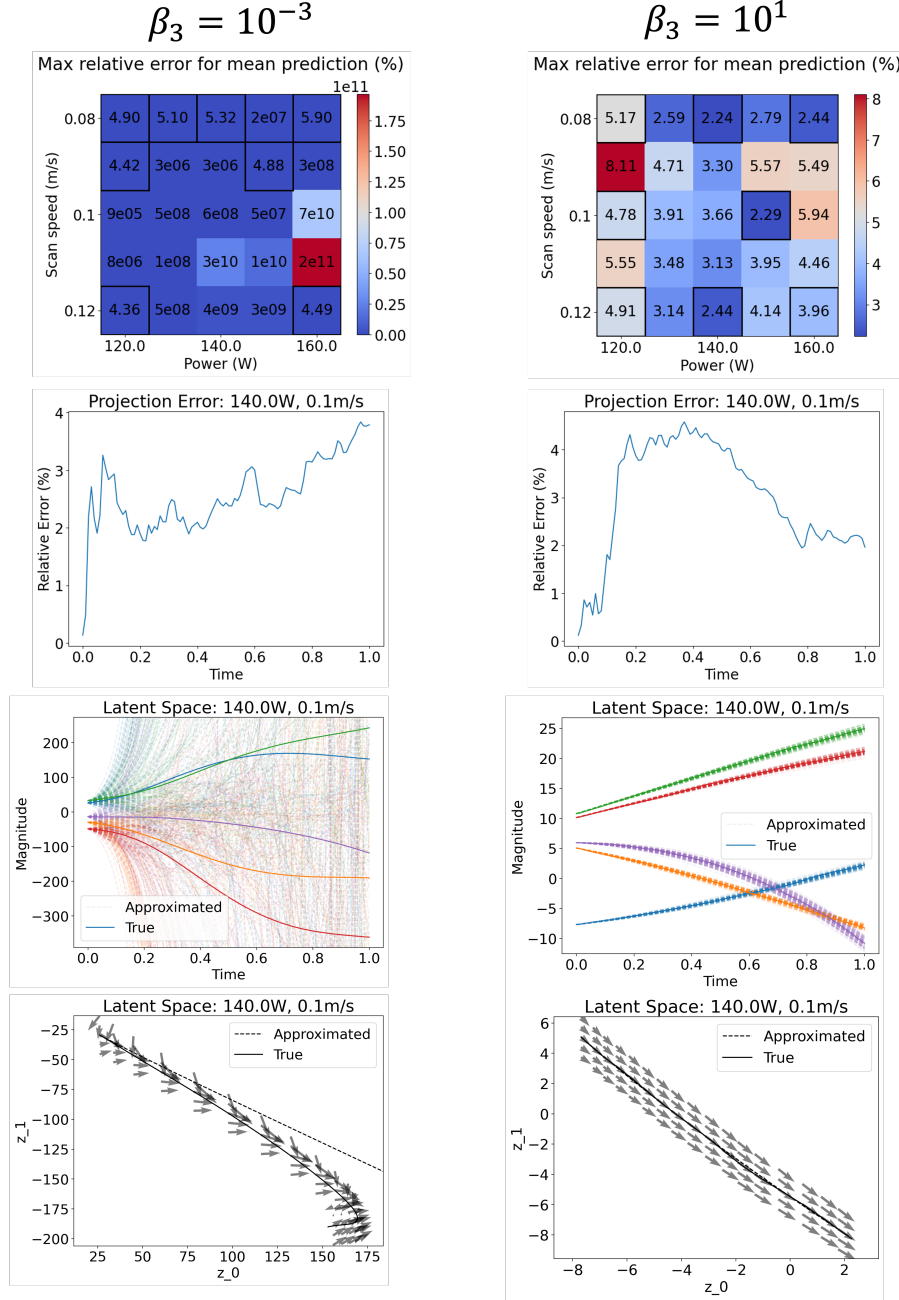
Figure 2: GPLaSDI results for $\beta_3 = 10^{-3}$ (left column) and $\beta_3 = 10^1$ (right column). First row: Maximum relative error for the mean GPLaSDI prediction over the parameter grid. Black squares denote training samples. Second row: Autoencoder projection error for $\boldsymbol{\mu}^{(*)} = (140\text{W}, 0.1\text{m/s})$. Third row: Latent space trajectories $\mathbf{Z}^{(*)}$ (true) and multiple $\widetilde{\mathbf{Z}}^{(*)}$ (approximated) for $\boldsymbol{\mu}^{(*)}$. Fourth row: Latent space trajectories (lines) and velocity (arrows) projected onto z_0-z_1 space for $\boldsymbol{\mu}^{(*)}$.

to full-order, physics-based simulations with only about 8% error in the worst case. In particular, the coefficient regularization weight β_3 has a large impact on model performance for the dataset considered in this work. For small β_3, learned latent space dynamics lead to unstable blow up of latent space trajectories on test samples. Increasing β_3 tends to smooth the latent space velocity field, preventing instabilities in the latent space trajectories and greatly improving prediction accuracy.

5

Acknowledgments and Disclosure of Funding

References

[1] TW Eagar, NS Tsai, et al. Temperature fields produced by traveling distributed heat sources. *Welding journal*, 62(12):346–355, 1983.

[2] P Honarmandi, R Seede, L Xue, D Shoukr, P Morcos, B Zhang, C Zhang, A Elwany, I Karaman, and R Arroyave. A rigorous test and improvement of the eagar-tsai model for melt pool characteristics in laser powder bed fusion additive manufacturing. *Additive Manufacturing*, 47:102300, 2021.

[3] Alexander J Wolfer, Jeremy Aires, Kevin Wheeler, Jean-Pierre Delplanque, Alexander Rubenchik, Andy Anderson, and Saad Khairallah. Fast solution strategy for transient heat conduction for arbitrary scan paths in additive manufacturing. *Additive Manufacturing*, 30:100898, 2019.

[4] Saad A Khairallah, Eric B Chin, Michael J Juhasz, Alan L Dayton, Arlie Capps, Paul H Tsuji, Kaila M Bertsch, Aurelien Perron, Scott K Mccall, and Joseph T Mckeown. High fidelity model of directed energy deposition: Laser-powder-melt pool interaction and effect of laser beam profile on solidification microstructure. *Additive Manufacturing*, 73:103684, 2023.

[5] Kevin Carlberg, Youngsoo Choi, and Syuzanna Sargsyan. Conservative model reduction for finite-volume models. *Journal of Computational Physics*, 371:280–314, 2018.

[6] Siu Wun Cheung, Youngsoo Choi, H. Keo Springer, and Teeratorn Kadeethum. Data-scarce surrogate modeling of shock-induced pore collapse process. 2023.

[7] Jessica T Lauzon, Siu Wun Cheung, Yeonjong Shin, Youngsoo Choi, Dylan Matthew Copeland, and Kevin Huynh. S-opt: A points selection algorithm for hyper-reduction in reduced order models. *arXiv preprint arXiv:2203.16494*, 2022.

[8] Ping-Hsuan Tsai, Seung Whan Chung, Debojyoti Ghosh, John Loffeld, Youngsoo Choi, and Jonathan L. Belof. Accelerating kinetic simulations of electrostatic plasmas with reduced-order modeling. 2023.

[9] William D Fries, Xiaolong He, and Youngsoo Choi. Lasdi: Parametric latent space dynamics identification. *Computer Methods in Applied Mechanics and Engineering*, 399:115436, 2022.

[10] Youngkyu Kim, Karen Wang, and Youngsoo Choi. Efficient space–time reduced order model for linear dynamical systems in python using less than 120 lines of code. *Mathematics*, 9(14):1690, 2021.

[11] Dylan Matthew Copeland, Siu Wun Cheung, Kevin Huynh, and Youngsoo Choi. Reduced order models for lagrangian hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 388:114259, 2022.

[12] Quincy A Huhn, Mauricio E Tano, Jean C Ragusa, and Youngsoo Choi. Parametric dynamic mode decomposition for reduced order modeling. *Journal of Computational Physics*, 475:111852, 2023.

[13] Sean McBane, Youngsoo Choi, and Karen Willcox. Stress-constrained topology optimization of lattice-like structures using component-wise reduced order models. *Computer Methods in Applied Mechanics and Engineering*, 400:115525, 2022.

[14] Youngsoo Choi, Deshawn Coombs, and Robert Anderson. Sns: A solution-based nonlinear subspace method for time-dependent model order reduction. *SIAM Journal on Scientific Computing*, 42(2):A1116–A1146, 2020.

[15] Siu Wun Cheung, Youngsoo Choi, Dylan Matthew Copeland, and Kevin Huynh. Local lagrangian reduced-order modeling for the rayleigh-taylor instability by solution manifold decomposition. *Journal of Computational Physics*, 472:111655, 2023.

[16] Chi Hoang, Youngsoo Choi, and Kevin Carlberg. Domain-decomposition least-squares petrov–galerkin (dd-lspg) nonlinear model reduction. *Computer methods in applied mechanics and engineering*, 384:113997, 2021.

[17] Alison L Marsden, Meng Wang, John E Dennis, and Parviz Moin. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering*, 5:235–262, 2004.

[18] Christophe Bonneville, Youngsoo Choi, Debojyoti Ghosh, and Jonathan L Belof. GPLaSDI: Gaussian process-based interpretable latent space dynamics identification through deep autoencoder. *Computer Methods in Applied Mechanics and Engineering*, 418:116535, 2024.

[19] LLNL. ALE3D for industry, https://ale3d4i.llnl.gov.

[20] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.