

---

# Accelerating Flow Simulations using Online Dynamic Mode Decomposition

---

**Seung Won Suh**

Department of Mechanical Science and Engineering  
University of Illinois, Urbana-Champaign  
Champaign, IL  
suh29@illinois.edu

**Seung Whan Chung**

Lawrence Livermore National Laboratory  
Livermore, CA  
chung28@llnl.gov

**Peer-Timo Bremer**

Lawrence Livermore National Laboratory  
Livermore, CA  
bremer5@llnl.gov

**Youngsoo Choi**

Lawrence Livermore National Laboratory  
Livermore, CA  
choi15@llnl.gov

## Abstract

We develop an on-the-fly reduced-order model (ROM) integrated with a flow simulation, gradually replacing a corresponding full-order model (FOM) of a physics solver. Unlike offline methods requiring a separate FOM-only simulation prior to model reduction, our approach constructs a ROM dynamically during the simulation, replacing the FOM when deemed credible. Dynamic mode decomposition (DMD) is employed for online ROM construction, with a single snapshot vector used for rank-1 updates in each iteration. Demonstrated on a flow over a cylinder with  $Re = 100$ , our hybrid FOM/ROM simulation is verified in terms of the Strouhal number, resulting in a 4.4 times speedup compared to the FOM solver.

## 1 Introduction

Iterative solvers for systems with a large number of degrees of freedom (DOFs) can be found in almost all areas of computational science and engineering. Physics simulations based on partial differential equations (PDEs) are one of the classic examples, where the solution is represented with a finite number of DOFs in space and marches in time as matrix equations are solved over multiple time steps. Optimization is another example where one first solves a primal problem to evaluate a quantity of interest and then solves its dual to compute sensitivities iteratively for gradient descents, e.g., PDE-constrained optimization or training of neural networks.

Although FOM-based solvers may generate predictions of high fidelity, they easily get computationally infeasible if one tries to resolve a much wider range of scales. At times when it is more important to get a rough estimate rapidly, one might need to consider building ROMs which are less accurate but cheaper than FOMs. Often used for building a ROM are modal analysis techniques such as the DMD, which is a data-driven algorithm that extracts the dynamics of the temporal evolution of a system [1].

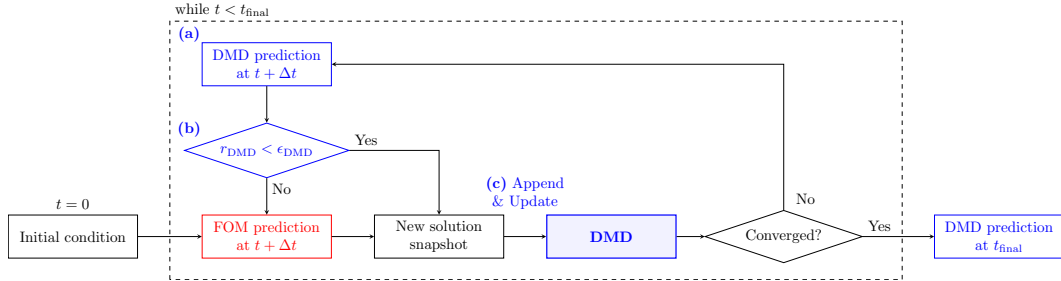


Figure 1: Workflow of the hybrid FOM/DMD simulation of a dynamical system.

Here we present an efficient workflow for constructing the ROM for flow simulations. Before getting started, we emphasize that such traditional reduced-order modeling is not different from the techniques used to reduce dimensionalities in the field of machine learning. For instance, the principal component analysis (PCA) is merely another name of the proper orthogonal decomposition (POD) [2], which is widely used for reduced-order modeling in physics simulations [3–16]. Therefore, our work of reducing the size of flow simulations using the DMD can be regarded adjacent to machine learning–based surrogate modeling and the workflow that we are presenting is not limited to the DMD but can be applied to any type of technique for dimensionality reduction.

A plethora of works have already shown the capability of the DMD to learn the nonlinear dynamics including that of fluid flows [17], pore-collapse [18], and radiative diffusion problem [19]. However, most of the studies are limited to offline training. That is, many full-order model simulations need to be run to generate the whole set of samples, and then the ROM is trained separately from the simulation using those samples. This may be undesirable when the dimensionality of the underlying parameter space is high, leading to the curse of dimensionality and requiring a large number of sampled training points. In such scenarios, efficiency can be enhanced by adopting an on-the-fly approach—interrupting the full-order model simulation and substituting the reduced-order model (ROM) at the relevant time step—rather than adhering to separate offline training and online inference steps. Via such a hybrid FOM/ROM method, we anticipate achieving two things at once: (1) accelerating the simulation itself, and (2) constructing a ROM that can be reused later for other simulations having similar setups, e.g., those for parametric studies. The existing on-the-fly data-driven approach includes projection-based reduced order modeling on PDE-constrained optimization [20, 21]. Similar approaches of using online POD have also been introduced by [22, 23], as well as those using artificial neural networks by [24] and [25].

In this work, we demonstrate a workflow for hybrid FOM/DMD flow simulations—which has not yet been attempted up to the authors’ knowledge—where the FOM is gradually replaced by the DMD trained on the fly. Each iteration consists of two key steps: (1) to update the DMD and (2) to replace the FOM with the DMD if possible. In addition, if the DMD is thought to have been converged, we end the simulation and directly make a prediction at the final time step, skipping all the remaining time steps and thereby significantly reducing the cost. This hybrid method has been applied to the simulation of a flow over a cylinder, followed by a comparison to a FOM–only simulation.

## 2 Methodology

### 2.1 Workflow

Figure 1 describes the workflow of a hybrid FOM/DMD simulation. Like standard solvers for time-dependent systems, we start the time integration with an initial condition. At the beginning of the first time step, there is no DMD initialized so we call the FOM and solve the system. Once we get a new snapshot of the solution, we use it to initialize the DMD, and then proceed to the next time step. At the beginning of the second time step, we first make a prediction on the solution at the second time step with our initialized DMD. If it is accurate enough, i.e. the residual of the prediction is smaller than the tolerance, we use that prediction as our new snapshot and move on to the next time step without calling the FOM. If not, we call the FOM and use the FOM prediction as the new

snapshot to update the previously-trained DMD model in a rank-1 update fashion. While the DMD gradually replaces the FOM, we also check whether the DMD itself has converged and if so, we exit the loop and go directly to  $t_{\text{final}}$  to make a prediction.

## 2.2 Error Estimator

One way to evaluate the accuracy of the prediction at the  $k$ -th time step made by the constructed DMD is to measure the local truncation error  $e_{\text{DMD}}^k = \mathbf{u}_{\text{DMD}}^k - \mathbf{u}_{\text{FOM}}^k$  where  $\mathbf{u}^k$  is the prediction of the solution at the  $k$ -th time step by either the DMD or the FOM. However, computing the error should be avoided since it requires solving the FOM for  $\mathbf{u}_{\text{FOM}}^k$ . Instead, we compute the residual as a representative of the local error. If a FOM solves a linear system  $\mathbf{A}\mathbf{u} = \mathbf{b}$ , the residual is defined as  $\mathbf{r}(\mathbf{u}) = \mathbf{b} - \mathbf{A}\mathbf{u}$  which does not require solving the matrix equation. As  $\mathbf{r}(\mathbf{u}_{\text{DMD}}) = -\mathbf{A}e_{\text{DMD}}$ , the residual is a linear map of the error and is a valid estimator of the error. Likewise, for nonlinear systems, one can derive residual-based error bounds, indicating that the residual norm is a good error indicator. In a nutshell, whenever we make a prediction with the DMD, we compute the residual as  $\mathbf{r}(\mathbf{u}_{\text{DMD}})$  and if its norm is smaller than a certain level of tolerance  $\epsilon_{\text{DMD}}$ , we trust the DMD and use its prediction as our new snapshot. Note that the residual depends on the numerical scheme used for the FOM.

## 2.3 Online DMD

The DMD approximates either linear or nonlinear dynamics using a linear operator. Consider a dynamical system that evolves over time as  $\mathbf{u}^{k+1} = \mathbf{F}(\mathbf{u}^k)$  where  $\mathbf{u}^k \in \mathbb{R}^N$  is an  $N$ -dimensional state vector at time steps  $k = 0, 1, \dots, n$  and  $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$  is an operator that describes the temporal evolution of the state. Then we can construct two matrices composed of snapshot vectors  $\mathbf{U}^+ = [\mathbf{u}^n | \mathbf{u}^{n-1} | \dots | \mathbf{u}^1] \in \mathbb{R}^{N \times n}$  and  $\mathbf{U}^- = [\mathbf{u}^{n-1} | \mathbf{u}^{n-2} | \dots | \mathbf{u}^0] \in \mathbb{R}^{N \times n}$ . The DMD matrix is then given as  $\mathbf{A} = \mathbf{U}^+(\mathbf{U}^-)^\dagger \approx \mathbf{U}^+ \mathbf{R} \mathbf{S}^{-1} \mathbf{L}^* = \mathbf{A}_r \in \mathbb{R}^{N \times N}$  where  $\mathbf{U}^- = \mathbf{L} \mathbf{S} \mathbf{R}^*$  is the truncated SVD at rank  $r \leq n$  ( $\mathbf{L} \in \mathbb{R}^{N \times r}$ ,  $\mathbf{S} \in \mathbb{R}^{r \times r}$ ,  $\mathbf{R} \in \mathbb{R}^{m \times r}$ ) and  $\dagger$  is the Moore-Penrose inverse of a matrix. Furthermore, the full-rank reduced DMD matrix is defined as  $\tilde{\mathbf{A}}_r = \mathbf{L}^* \mathbf{U}^+ \mathbf{R} \mathbf{S}^{-1} \in \mathbb{R}^{r \times r}$  and the projected DMD modes are  $\Phi = \mathbf{L} \mathbf{V} \in \mathbb{R}^{N \times r}$  where  $\tilde{\mathbf{A}}_r = \mathbf{V} \Lambda \mathbf{V}^{-1}$  is the eigenvalue-decomposition of  $\tilde{\mathbf{A}}_r$ . Once the DMD matrices are constructed, one can predict the solution at any time  $t$  based on the analytical form  $\mathbf{u}(t) = \Phi \Lambda^{t/\Delta t} \Phi^\dagger \mathbf{u}(0)$ . Hence, the DMD forms a ROM that makes a prediction at a constant time complexity of  $\mathcal{O}(Nr)$ , independent of  $t$ .

In our hybrid approach, updating the DMD matrices by a single rank is sufficient as only a single snapshot is appended to the snapshot matrix every iteration. Incremental SVD by [26] is an algorithm that efficiently makes a rank-1 update to the SVD of a matrix. The essence of the algorithm is to avoid costly matrix-matrix multiplications and utilize matrix-vector multiplications to get a speedup by  $\mathcal{O}(r)$ . We can employ the ideas from incremental SVD to efficiently make rank-1 updates on the DMD matrices. Although several authors have already introduced the idea of online DMD [27, 28], ours differ from theirs as [28] is based on the Sherman–Morrison formula and [27] used the modified version of the original algorithm [26], while we stick to the original version to update the DMD matrices. In the end, each DMD update costs  $\mathcal{O}(Nr)$ , which is more efficient than naively constructing the matrices from scratch that costs  $\mathcal{O}(Nnr)$  for  $N \gg n \gg r$ . Details of the implementation are not presented in this paper due to the limited space.

## 3 Numerical Experiments & Results

The prescribed hybrid workflow was applied to simulate a three-dimensional incompressible flow over a cylinder. The cylinder has a diameter of  $D$  with its axis along the  $z$ -direction at  $(x_c, y_c) = (5D, 0)$ . The computational domain was defined as  $\Omega = [0, L_x] \times [-L_y/2, L_y/2] \times [-L_z/2, L_z/2]$  with the dimensions  $(L_x, L_y, L_z) = (25D, 4.1D, 4.1D)$ . Dirichlet boundary conditions were set for the velocities at the inflow ( $x = 0$ ) and the top and bottom walls ( $y = \pm L_y/2$ ) as  $(u, v) = (U, 0)$ , and pure Neumann boundary conditions were set elsewhere. For the pressure, pure Neumann boundary conditions were applied at the inflow and the top and bottom walls, and the Dirichlet boundary condition was used at the outflow ( $x = L_x$ ) as  $p = 0$ . The kinematic viscosity  $\nu$  was set to match the Reynolds number  $\text{Re} = UD/\nu = 100$  where we can anticipate mild vortex streets. Although

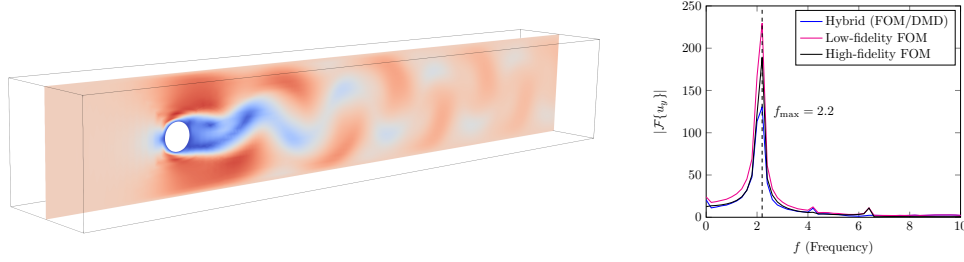


Figure 2: Velocity magnitude plotted on a 2-D slice normal to the  $z$ -axis at  $t^* = 50$  (left); temporal spectrum of  $u_y$  sampled downstream.

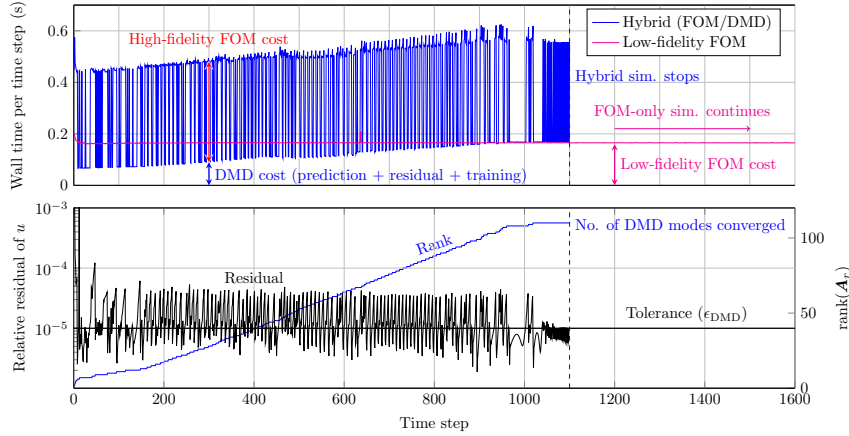


Figure 3: Wall time per time step: hybrid vs. FOM (top); history of the relative residual of  $u_{\text{DMD}}$  and the rank of  $A_r$  (bottom). The horizontal axis only shows until the 1,600<sup>th</sup> time step out of 10,000 time steps of the FOM-only simulation.

the top and bottom boundaries were quite close to the cylinder, we were able to observe a distinct vortex-shedding structure throughout the simulation.

An incompressible flow solver in MFEM [29] was used as the FOM, which uses a nodal  $H^1$  finite element discretization with polynomial order of 3 in space and an implicit (of order 3) / explicit (of order 3) time integration of the viscous and convective terms in the momentum equation, respectively. More detailed explanations of the numerical scheme can be found in [29–31].

An unstructured coarse baseline mesh was further refined within the solver where the total number of DOFs was 585,732. A preliminary high-fidelity FOM simulation was first run for  $t^* = Ut/D = 50$  starting from the zero flow until it was thought to have entered the limit cycle of the vortex shedding. The time step size was adjusted so the maximum CFL number remained around 0.2 throughout the simulation. Then, the final snapshot of the solution was used to initialize flow fields for the hybrid and restarted FOM-only simulations, which were both run until  $t^* = 100$ . For a fair comparison, the level of fidelity of the restarted FOM-only simulation has been lowered down to that of the DMD: the tolerance level of the conjugate-gradient solver ( $\epsilon_{\text{CG}}$ ) for the FOM-only simulation and the tolerance level for the accuracy of the DMD ( $\epsilon_{\text{DMD}}$ ) were both set to  $10^{-5}$ . Note that the FOM for the hybrid simulation retains the high fidelity of  $\epsilon_{\text{CG}} = 10^{-12}$ .

Temporal spectra of the  $y$ -velocity at a single point  $(x, y, z) = (10D, 0, 0)$  from the hybrid and a high-fidelity ( $\epsilon_{\text{CG}} = 10^{-12}$ ) FOM-only simulations were compared for verification. Frequencies of the vortex shedding predicted by the two simulations were identical within the precision of the discrete values of frequency. The Strouhal number  $St = fD/U$ —the dimensionless frequency of the vortex shedding—was 0.22 in both simulations, which is around the reported value of 0.16 at  $Re = 100$  [32]. The discrepancy is thought to be due to the boundary effect.

Figure 3 shows the cost of the two simulations and the history of the residual of  $u$  from the hybrid simulation. Notice that at the early stage of the simulation, the residual keeps going up and down

	Hybrid (FOM/DMD)	Low-fidelity FOM	Speedup
Total simulation time (s)	375	1648	<b>4.39</b>
Single- $\Delta t$ prediction time (s)	0.0286	0.165	<b>5.77</b>

Table 1: Two different types of speedups, in terms of the total simulation time and the cost of a single time-stepping. The total simulation time for the FOM/DMD approach includes the time for making a prediction, checking accuracy, and training. The cost of a single time-stepping only measures the time spent on making a prediction, based on the converged DMD.

around the level of tolerance ( $10^{-5}$ ) so the FOM is frequently called. As the simulation goes on, the model gets more accurate and hits below the level of tolerance more often, leading to less frequent calls of the FOM. This depicts how the FOM is gradually replaced by the DMD throughout the simulation as we expected. In total, the hybrid and the low-fidelity FOM-only simulations took 375s and 1648s for  $10^4$  time steps, respectively, thereby showing about a 4.4-times speedup. Moreover, even when we choose not to skip all the remaining time steps of the simulation, a single time-stepping using the converged DMD is 5.77 times faster than using the low-fidelity FOM so the total cost of our approach will remain cheaper.

## 4 Conclusion

We used the DMD to reduce the dimension of the FOM that is evaluated iteratively during flow simulations, where the DMD is constructed on the fly and gradually replaces the FOM. This has been demonstrated on a simulation of a flow over a cylinder, and we were able to gain a 4.4-times speedup.

The fact that the DMD discovers underlying physics in a data-driven way without any prior knowledge of the governing equations resembles the way machine learning is used for scientific discovery. Therefore, our work has both aspects of traditional reduced-order modeling and machine learning and will be a good resource for following studies on model reduction of dynamical systems in both fields.

## Acknowledgments and Disclosure of Funding

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344 and was supported by Laboratory Directed Research and Development funding under projects 22-SI-004 and 22-SI-006. Y. Choi was supported for this work by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, as part of the CHARMNET Mathematical Multifaceted Integrated Capability Center (MMICC) program, under Award Number DE-SC0023164.

## References

- [1] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.
- [2] G. Berkooz, P. Holmes, and J. L. Lumley, “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993.
- [3] K. Carlberg, Y. Choi, and S. Sargsyan, “Conservative model reduction for finite-volume models,” *Journal of Computational Physics*, vol. 371, pp. 280–314, 2018.
- [4] Y. Choi, G. Boncoraglio, S. Anderson, D. Amsellem, and C. Farhat, “Gradient-based constrained optimization using a database of linear reduced-order models,” *Journal of Computational Physics*, vol. 423, p. 109787, 2020.
- [5] Y. Choi and K. Carlberg, “Space–time least-squares petrov–galerkin projection for nonlinear model reduction,” *SIAM Journal on Scientific Computing*, vol. 41, no. 1, pp. A26–A58, 2019.
- [6] Y. Choi, P. Brown, W. Arrighi, R. Anderson, and K. Huynh, “Space–time reduced order model for large-scale linear dynamical systems with application to boltzmann transport problems,” *Journal of Computational Physics*, vol. 424, p. 109845, 2021.

- [7] C. Hoang, Y. Choi, and K. Carlberg, “Domain-decomposition least-squares petrov–galerkin (dd-lspg) nonlinear model reduction,” *Computer methods in applied mechanics and engineering*, vol. 384, p. 113997, 2021.
- [8] S. McBane and Y. Choi, “Component-wise reduced order model lattice-type structure design,” *Computer methods in applied mechanics and engineering*, vol. 381, p. 113813, 2021.
- [9] Y. Choi, D. Coombs, and R. Anderson, “Sns: A solution-based nonlinear subspace method for time-dependent model order reduction,” *SIAM Journal on Scientific Computing*, vol. 42, no. 2, pp. A1116–A1146, 2020.
- [10] D. M. Copeland, S. W. Cheung, K. Huynh, and Y. Choi, “Reduced order models for lagrangian hydrodynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 388, p. 114259, 2022.
- [11] Y. Kim, K. Wang, and Y. Choi, “Efficient space–time reduced order model for linear dynamical systems in python using less than 120 lines of code,” *Mathematics*, vol. 9, no. 14, p. 1690, 2021.
- [12] W. D. Fries, X. He, and Y. Choi, “Lasdi: Parametric latent space dynamics identification,” *Computer Methods in Applied Mechanics and Engineering*, vol. 399, p. 115436, 2022.
- [13] J. T. Lauzon, S. W. Cheung, Y. Shin, Y. Choi, D. M. Copeland, and K. Huynh, “S-opt: A points selection algorithm for hyper-reduction in reduced order models,” *arXiv preprint arXiv:2203.16494*, 2022.
- [14] S. W. Cheung, Y. Choi, D. M. Copeland, and K. Huynh, “Local lagrangian reduced-order modeling for the rayleigh-taylor instability by solution manifold decomposition,” *Journal of Computational Physics*, vol. 472, p. 111655, 2023.
- [15] S. McBane, Y. Choi, and K. Willcox, “Stress-constrained topology optimization of lattice-like structures using component-wise reduced order models,” *Computer Methods in Applied Mechanics and Engineering*, vol. 400, p. 115525, 2022.
- [16] P.-H. Tsai, S. W. Chung, D. Ghosh, J. Loffeld, Y. Choi, and J. L. Belof, “Accelerating kinetic simulations of electrostatic plasmas with reduced-order modeling,” 2023.
- [17] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
- [18] S. W. Cheung, Y. Choi, H. K. Springer, and T. Kadeethum, “Data-scarce surrogate modeling of shock-induced pore collapse process,” 2023.
- [19] Q. A. Huhn, M. E. Tano, J. C. Ragusa, and Y. Choi, “Parametric dynamic mode decomposition for reduced order modeling,” *Journal of Computational Physics*, vol. 475, p. 111852, 2023.
- [20] Y. Choi, G. Oxberry, D. White, and T. Kirchdoerfer, “Accelerating design optimization using reduced order models,” *arXiv preprint arXiv:1909.11320*, 2019.
- [21] T. Wen and M. J. Zahr, “A globally convergent method to accelerate large-scale optimization using on-the-fly model hyperreduction: application to shape optimization,” *Journal of Computational Physics*, vol. 484, p. 112082, 2023.
- [22] I. Rocha, F. van der Meer, and L. J. Sluys, “An adaptive domain-based pod/ecm hyper-reduced modeling framework without offline training,” *Computer Methods in Applied Mechanics and Engineering*, vol. 358, p. 112650, 2020.
- [23] C. Huang and K. Duraisamy, “Predictive reduced order modeling of chaotic multi-scale problems using adaptively sampled projections,” *arXiv preprint arXiv:2301.09006*, 2023.
- [24] F. Fritzen, M. Fernández, and F. Larsson, “On-the-fly adaptivity for nonlinear twoscale simulations using artificial neural networks and reduced order modeling,” *Frontiers in Materials*, vol. 6, p. 75, 2019.

- [25] I. Rocha, P. Kerfriden, and F. van Der Meer, “On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning,” *Journal of Computational Physics: X*, vol. 9, p. 100083, 2021.
- [26] M. Brand, “Fast low-rank modifications of the thin singular value decomposition,” *Linear algebra and its applications*, vol. 415, no. 1, pp. 20–30, 2006.
- [27] D. Matsumoto and T. Indinger, “On-the-fly algorithm for dynamic mode decomposition using incremental singular value decomposition and total least squares,” *arXiv preprint arXiv:1703.11004*, 2017.
- [28] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, “Online dynamic mode decomposition for time-varying systems,” *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 3, pp. 1586–1609, 2019.
- [29] “MFEM: Modular finite element methods [Software].” [mfem.org](http://mfem.org).
- [30] A. Tomboulides, J. Lee, and S. Orszag, “Numerical simulation of low mach number reactive flows,” *Journal of Scientific Computing*, vol. 12, pp. 139–167, 1997.
- [31] M. Franco, J.-S. Camier, J. Andrej, and W. Pazner, “High-order matrix-free incompressible flow solvers with gpu acceleration and low-order refined preconditioners,” *Computers & Fluids*, vol. 203, p. 104541, 2020.
- [32] C. H. Williamson, “Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low reynolds numbers,” *Journal of Fluid Mechanics*, vol. 206, pp. 579–627, 1989.