Data-Driven Koopman Reduced-Order Models for Kinetic Plasmas and Electromagnetic Cavities

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Indranil Nayak, B.Tech (Hons), M.S.

Graduate Program in Electrical and Computer Engineering

The Ohio State University

2024

Dissertation Committee:

Dr. Fernando L. Teixeira, Advisor

Dr. Mrinal Kumar, Co-Advisor

Dr. Balasubramaniam Shanker

Dr. Ben McCorkle

Abstract

We present an exposition on Koopman operator-based reduced-order modeling of high-dimensional electromagnetic (EM) systems exhibiting both linear and nonlinear dynamics. Since the emergence of the digital age, numerical methods have been pivotal in understanding physical phenomena through computer simulations. Computational electromagnetics (CEM) and computational plasma physics (CPP) are related yet distinct branches, each addressing complex linear and nonlinear electromagnetic phenomena. CEM primarily focuses on solving Maxwell's equations for intricate structures such as antennas, cavities, high-frequency circuits, waveguides, and scattering problems. In contrast, CPP aims to capturing the complex behavior of charged particles under electromagnetic fields. This work specifically focuses on the numerical simulation of electromagnetic cavities and particle-in-cell (PIC) kinetic plasma simulations.

Studying electromagnetic field coupling inside metallic cavities is crucial for various applications, including electromagnetic interference (EMI), electromagnetic compatibility (EMC), shielded enclosures, cavity filters, and antennas. However, time-domain simulations can be computationally intensive and time-consuming, especially

as the scale and complexity of the problem increase. Similarly, PIC simulations, which are extensively used for simulating kinetic plasmas in the design of high-power microwave devices, vacuum electronic devices, and in astrophysical studies, can be computationally demanding, especially when simulating thousands to millions of charged particles. Moreover, the nonlinear nature of the complex wave-particle interactions complicates the modeling task.

Data-driven reduced-order models (ROMs), which have recently gained prominence due to advances in machine learning techniques and hardware capabilities, offer a practical approach for constructing "light" models from high-fidelity data. The Koopman operator-based data-driven ROM is a powerful method for modeling high-dimensional dynamical systems, particularly those exhibiting nonlinear behavior. We explore the realizations of the finite-dimensional Koopman operator through dynamic mode decomposition (DMD) and Koopman autoencoders (KAEs). We demonstrate how DMD and KAE can model the fields and currents in resonating cavities and plasma systems. KAEs leverage the expressivity of neural networks and can outperform DMD for highly nonlinear problems. Furthermore, neural network-based approaches such as KAE offer a straightforward way to incorporate physical constraints into the model, leading to more accurate and stable long-term predictions. Additionally, we develop an on-the-fly DMD algorithm to detect in real-time when to stop the high-fidelity time-domain simulations by identifying the onset of self-repeating behavior.

To my parents.

Acknowledgments

Embarking on a Ph.D. journey after my bachelor's degree was exciting but daunting at the same time. I am grateful to both my advisors, Dr. Fernando Teixeira and Dr. Mrinal Kumar, for setting my research direction in my initial years. Their support and trust in me to carry out research in the later years helped me become an independent researcher. They have been very generous in trusting me with computational resources as well as new ideas. They have been considerate during my tough times when publications did not come easily. Without their constant support and backing, my Ph.D. experience would not have been smooth.

I would like to acknowledge the mentorship of several faculty members and principal investigators I came in touch with during my Ph.D. Prof. Kubiley Sertel's research philosophies and working principles, Dr. Robert Burkholder's leadership and management qualities, and Prof. Joel Johnson's practical and grounded approach have all helped me become a better researcher. I would like to especially mention Prof. Balasubramanium Shanker, with whom I came in touch mainly during the final year of my Ph.D.; interactions with him enriched me greatly, from research philosophies to how to be successful in academia. I would like to thank him for graciously

agreeing to participate in my dissertation committee. I would also like to thank Dr. Debdipta Goswami, who joined the university during the later years of my Ph.D., for being an excellent collaborator and mentor.

My Ph.D. journey would not have been as exciting without my friends and colleagues, my lab group members, and friends from the Electroscience Lab. My collaborators from Electrical and Computer Engineering and Mechanical and Aerospace Engineering have enriched my experience at The Ohio State University through both intellectual and non-intellectual discussions. I extend special thanks to my girlfriend, who has been a constant source of strength and encouragement over the past few years.

Above all, I would like to express my deepest love and gratitude to my family—my parents and sisters—whose unwavering support and unconditional love have made this journey possible.

This work would not have been possible without the support of the sponsors. This work was partially supported by the Defense Threat Reduction Agency under Grant HDTRA1-18-1-0050, the Air Force Office of Scientific Research under Grant No. FA9550-20-1-0083, the Department of Energy Grant DE-SC0022982 through the NSF/DOE Partnership in Basic Plasma Science and Engineering, the Defense Advanced Research Projects Agency (DARPA) Cooperative Agreement HR 0011-21-2-0020, and in part by the Ohio Supercomputer Center Grant PAS0061. Approved for public release; distribution is unlimited.

Vita

May 26, 1995	Born - West Bengal, India.
2014-2018	B.Tech (Hons.), Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur.
2018-2024	M.S. (2021) & Ph.D., Electrical and Computer Engineering, The Ohio State University.

Publications

Research Publications

Journal Publications

I. Nayak, F. L. Teixeira, D. Y. Na, M. Kumar, and Y. A. Omelchenko, "Accelerating particle-in-cell kinetic plasma simulations via reduced-order modeling of space-charge dynamics using dynamic mode decomposition," Phys. Rev. E, vol. 109, no. 6, p. 065307, Jun. 2024.

I. Nayak, F. L. Teixeira and R. J. Burkholder, "On-the-Fly Dynamic Mode Decomposition for Rapid Time-Extrapolation and Analysis of Cavity Resonances," in *IEEE Transactions on Antennas and Propagation.*

I. Nayak, M. Kumar, and F. L. Teixeira, "Detection and prediction of equilibrium states in kinetic plasma simulations via mode tracking using reduced-order dynamic mode decomposition," *J. Comp. Phys.*, vol. 447, p. 110671, 2021.

I. Nayak, D. Na, J. L. Nicolini, Y. A. Omelchenko and F. L. Teixeira, "Progress in Kinetic Plasma Modeling for High-Power Microwave Devices: Analysis of Multipactor Mitigation in Coaxial Cables," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 2, pp. 501-508, Feb. 2020.

A. Cortez, B. Ford, **I. Nayak**, S. Narayanan, and M. Kumar, "Hybrid A* path search with resource constraints and dynamic obstacles," *Frontiers in Aerospace Engineering*, vol. 1, 2023.

Conference Publications

I. Nayak, F. L. Teixeira and M. Kumar, "Koopman Autoencoder Architecture for Current Density Modeling in Kinetic Plasma Simulations," *2021 International Applied Computational Electromagnetics Society Symposium (ACES)*, 2021, pp. 1-3.

I. Nayak and F. L. Teixeira, "Reduced-Order Modeling of Backward Wave Oscillator Fields for High-Power Microwave Applications," in *Proc. 2024 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, 2024, pp. 249-250.

R. J. Burkholder, G. Kimball, D. Nippa, S. Dogan, J. T. Johnson, J. L. Nicolini, **I. Nayak**, F. L. Teixeira, C. J. McKelvey, T.-H. Lee, et al., "A Full-Spectrum Approach to Electromagnetic Coupling into Complex Enclosures," in *2023 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (USNC-URSI)*, 2023, pp. 813–814.

J. Murray, B. Chamberlain, N. Hemleben, D. Ospina-Acero, **I. Nayak**, A. VanFossen, F. Zahiri, and M. Kumar, "Reconceptualizing the Prognostics Digital Twin for Smart Manufacturing with Data-Driven Evolutionary Models and Adaptive Uncertainty Quantification," in *Proceedings of the Annual Conference of the PHM Society*, vol. 15, no. 1, 2023.

I. Nayak, J. L. Nicolini, D.-Y. Na, and F. L. Teixeira, "Recent Progress in Reduced Order Models for Electromagnetic Particle-in-Cell Kinetic Plasma Simulations," in *Proc. 2023 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (USNC-URSI)*, 2023, pp. 1617-1618.

S. Narayanan, **I. Nayak**, and M. Kumar, "An iterative scheme to learn system dynamics of space objects from partial state information," in *AIAA SCITECH 2023 Forum*, 2023, p. 0124.

A. C. Cortez, B. T. Ford, **I. Nayak**, S. Narayanan, and M. Kumar, "Path Planning for a Dubins Agent with Resource Constraints and Dynamic Obstacles," in *Proc. AIAA SCITECH 2023 Forum*, 2023, p. 1054.

S. M. Chowdhury, **I. Nayak**, Q. M. Marashdeh, and F. L. Teixeira, "Developments on Imaging and Velocimetry of Dynamic Two-Phase Flows with Electrical Capacitance Tomography Sensors," in *Proc. 2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, 2022, pp. 1708-1709.

I. Nayak and F. L. Teixeira, "Data-Driven Modeling of High-Q Cavity Fields Using Dynamic Mode Decomposition," *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, 2022, pp. 1118-1119.

S. Narayanan, **I. Nayak**, and M. Kumar, "On the application of time delay embedding for the data driven discovery of nonlinear systems from partial state information," in *AIAA SCITECH 2022 Forum*, 2022, p. 2440.

I. Nayak and F. L. Teixeira, "Dynamic Mode Decomposition Reduced-Order Models for Multiscale Kinetic Plasma Analysis," in *Proc. 2021 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, 2021, pp. 261-262.

I. Nayak, M. Kumar, and F. Teixeira, "Detecting equilibrium state of dynamical systems using sliding-window reduced-order dynamic mode decomposition," in *AIAA Scitech 2021 Forum*, p. 1858, 2021.

I. Nayak and F. L. Teixeira, "Dynamic Mode Decomposition for Prediction of Kinetic Plasma Behavior," *2020 International Applied Computational Electromagnetics Society Symposium (ACES)*, 2020, pp. 1-2.

D.-Y. Na, **I. Nayak**, and F. L. Teixeira, "Numerical Analysis on Multipactor Effects in Coaxial Cables via Particle-in-Cell Algorithm," in *Proc. 2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, 2019, pp. 1849-1850.

Book Chapters

I. Nayak, M. Kumar, and F. L. Teixeira, "Koopman Autoencoders for Reduced-Order Modeling of Kinetic Plasmas," in Advances in Electromagnetics Empowered by Artificial Intelligence and Deep Learning, S. Campbell and D. Werner, Eds., 2023.

Fields of Study

Major Field: Electrical and Computer Engineering

Contents

Page

List of Tables

List of Figures

Figure **Page**

List of Symbols

$\mathbf{x}$	Original state
N	Original state-space dimension
n	Time step index
F	Nonlinear flowmap
$\mathcal{M}$	Original manifold
$g(\cdot)$	Scalar Koopman observable
$\mathbf{g}(\cdot)$	Vector-valued Koopman observable
$\mathbf{z}$	Latent state in Koopman observable space ($= \mathbf{g}(\mathbf{x})$)
$\mathcal{K}$	Infinite-dimensional discrete-time Koopman operator
$\phi(\cdot)$	Koopman eigenfunctions
$\mathbf{v}$	Koopman modes
λ	Eigenvalues of discrete-time Koopman operator
$\widehat{\lambda}(\cdot)$	Barycentric coordinate
$\mathbf{K}$	Finite-dimensional discrete-time Koopman operator.
$\mathcal{S}_{\mathcal{K}}$	Koopman-invariant subspace
$d_{\mathcal{S}}$	Dimension of $\mathcal{S}_{\mathcal{K}}$
$\mathbf{u}$	Right eigenvectors of $\mathbf{K}$
$\mathbf{w}$	Left eigenvectors of $\mathbf{K}$
Δt	Time step interval for numerical simulation
k	Time sample or time snapshot index

Δ_n	Number of time steps between consecutive time samples
Δ_t	Time interval between consecutive time samples
l	Number of time samples in training window
n_k	Time step index corresponding to k^{th} time sample
$\mathbf{X}_k$	Snapshot matrix starting from k^{th} time sample
$\mathbf{A}$	DMD state transition matrix
$(\cdot)^\dagger$	Moore-Penrose pseudo inverse
$\mathbf{U}$	Left singular matrix
$\mathbf{\Sigma}$	Singular values matrix
$\mathbf{V}$	Right singular matrix
$(\cdot)^H$	Complex-conjugate transpose
r	Reduced rank (dimension)
$\tilde{\mathbf{U}}, \tilde{\mathbf{\Sigma}}, \tilde{\mathbf{V}}$	Rank-reduced singular (SVD) matrices
$\tilde{\mathbf{A}}$	Low-dimensional projection of $\mathbf{A}$
$\boldsymbol{\phi}$	DMD modes
ϑ	DMD modal amplitudes
$\widehat{\omega}$	DMD frequency
$(\cdot)^*$	Complex-conjugate
d	Number of Hankel stacks
$[\![\mathbf{X}]\!]_\nu$	Hankel snapshot matrix
$[\![\mathbf{A}]\!]$	Hankel DMD state transition matrix
$\mathbf{\Psi}_e$	KAE encoder layer
$\mathbf{\Psi}_d$	KAE decoder layer
$\mathbf{K}_f$	KAE forward dynamics layer
$\mathbf{K}_b$	KAE backward dynamics layer
N_{in}	Number of nodes in KAE input layer
N_{out}	Number of nodes in KAE output layer

N_h	Number of nodes in KAE hidden layers
N_b	Number of nodes in KAE bottleneck layer
k_m	Lookahead (or lookback) steps for KAE training
$\mathcal{L}_{id}$	Identity loss for KAE training
γ_{id}	Weight corresponding to $\mathcal{L}_{id}$
$\mathcal{L}_{fwd}$	Forward loss for KAE training
γ_{fwd}	Weight corresponding to $\mathcal{L}_{fwd}$
$\mathcal{L}_{bwd}$	Backward loss for KAE training
γ_{bwd}	Weight corresponding to $\mathcal{L}_{bwd}$
$\mathcal{L}_{con}$	Consistency loss for KAE training
γ_{con}	Weight corresponding to $\mathcal{L}_{con}$
$\mathbf{I}$	Identity matrix
$\mathbf{I}_N$	Identity matrix with dimension $N \times N$
$\mathbf{X}_k^{(w)}$	$\mathbf{X}_k$ corresponding to w^{th} DMD window
$\boldsymbol{\phi}_i^{(w)}$	i^{th} DMD mode corresponding to w^{th} DMD window
$\lambda_i^{(w)}$	i^{th} DMD eigenvalue corresponding to w^{th} DMD window
$(\cdot)^{\text{T}}$	Transpose
ρ	Absolute value of modal assurance criterion
d_λ	Absolute value of the difference between two complex DMD eigenvalues
$\mathbf{E}$	Electric field intensity
$\mathbf{H}$	Magnetic field intensity
$\mathbf{J}$	Electric current density
$\mathbf{M}$	Magnetic current density
ϵ	Permittivity of the medium
μ	Permeability of the medium
σ	Electric conductivity of the medium
$\mathbf{D}$	Electric flux density

$\mathbf{B}$	Magnetic flux density
ρ_c	Radius in cylindrical coordinate system
ϕ_c	Azimuthal angle in cylindrical coordinate system
L	Maximum cavity dimension
$e_{z,i}$	z directed electric field component at i^{th} FDTD grid point
$\mathbf{e}_z$	Vector made of $e_{z,i}$ s
ρ_q	Charge density
e_i	Degrees of freedom (DoF) of self electric field at i^{th} edge in FETD setting
$\mathbf{e}$	Vector made of e_i s
b_i	DoF of self magnetic field at i^{th} element (traingle) in FETD setting
$\mathbf{b}$	Vector made of b_i s
j_i	DoF of current density at i^{th} edge in FETD setting
$\mathbf{j}$	Vector made of j_i s
N_0	Number of nodes in the spatial grid
N_1	Number of edges in the spatial grid
N_2	Number of triangles in the spatial grid
N_p	Number of superparticles in the solution domain
$\mathbf{C}$	Discrete curl operator on primal grid
$\tilde{\mathbf{C}}$	Discrete curl operator on dual grid
$\mathbf{S}$	Discrete divergence operator on primal grid
$\tilde{\mathbf{S}}$	Discrete divergence operator on dual grid
$[\star_\epsilon]$	Discrete Hodge star operator corresponding to ϵ
$[\star_{\mu^{-1}}]$	Discrete Hodge star operator corresponding to μ
$\mathbf{r}_p$	Position of p^{th} superparticle
$\mathbf{v}_p$	Relativistic velocity of p^{th} superparticle
γ_p	Lorentz factor corresponding to p^{th} superparticle
c	Velocity of light in free space

q_p	Charge of the p^{th} superparticle
q_{sp}	Charge of each superparticle
r_{sp}	Superparticle ratio
m_{sp}	Mass of each superparticle
$\mathbf{W}^0$	Vector proxy of Whitney zero-form
$\mathbf{W}^1$	Vector proxy of Whitney one-form
$\mathbf{W}^2$	Vector proxy of Whitney two-form

Chapter 1: Introduction

1.1 Motivation

Since the introduction of computers and digital computing, numerical techniques have been a key tool in simulating complex physical phenomena. Often the practical problems we deal with in science and engineering do not have a closed form analytical solution, necessitating the use of numerical algorithms which break the solution domain in discrete smaller portions and solve for the quantities of interest from first principles. Fast, accurate and reliable electromagnetic models are essential for efficient and accelerated design of high-performance microwave and optical devices such as antennas, filters, high-frequency circuit components etc. These simulations are also essential for the study of electromagnetic interference (EMI) and electromagnetic compatibility (EMC) effects which have become critical with the advent of internet of things (IoT), 5G and ever increasing density of electronic components in everyday gadgets. The present analytical tools are limited in their ability since they are often restricted to simple geometries. Computational electromagnetics (CEM)

has been an essential tool for physicists and engineers for simulating electromagnetic wave propagation through complex structures, electromagnetic scattering, radiation etc. While the numerical tools such as finite-difference time-domain (FDTD), finite-element time-domain (FETD), finite-element method (FEM) etc [3, 4]. are able to accurately capture electromagnetic behavior for a broad range of structures [5], these methods don't scale well as the problem becomes electrically large or complicated. This challenge is further compounded in space applications, study of astrophysical phenomena, design and analysis of particle accelerators, high-power microwave sources, fusion reactors etc. [6, 7, 8, 9], where interaction of electromagnetic waves with plasma needs to be accounted for. A plasma can be defined as a quasi-neutral ionized mixture of neutral, positive-, and negatively-charged, mutually interacting particles. Computational plasma physics (CPP) deals with numerical simulation of plasma dynamics which can be broadly divided into a fluids model, which deals with magnetohydrodynamics (MHD) eqution [10, 11, 12] and a kinetic model, which is governed by governed by Maxwell-Vlasov equations [6, 7, 13]. The MHD or fluids model captures well the large-scale behavior of certain plasmas at length scales larger that the mean free-path. Kinetic models, on the other hand, are designed for scenarios where the mean free-path is much larger than the characteristics dimensions of the problem. The Particle-in-cell (PIC) kinetic plasma simulations [14, 15] can accurately capture intricate wave-particle interactions including electron bunching, kinetic instabilities, Landau damping, microscopic turbulence, space-charge effects,

etc. otherwise not possible through MHD [16, 17, 18, 19]. Electromagnetic particle-EMPIC plasma simulations play an important role in the analysis of such "collisionless" plasmas, which are present in high-power microwave sources, laser ignited devices, particle accelerators, vacuum electronic devices, and many other applications [6, 7, 13, 20, 21, 22, 23, 24].

EMPIC can be seen as an extension to the FDTD or FETD where one must additionally deal with nonlinear interactions of charged particles with electromagnetic fields. Both the EMPIC and FDTD/FETD suffer from the drawbacks of any grid-based numerical method, namely, the curse of dimensionality. The "curse" manifests in the form of an exponential growth in the number of elements required for discretization as a function of the dimensionality of the solution domain. In this thesis, we are primarily concerned with time-domain numerical methods. The time-domain numerical methods provide certain advantages over frequency domain method for simulating transient phenomena, nonlinear effects, wideband devices and electromagnetic cavities especially with high Q-factor. The PIC plasma simulations are inherently time-domain since they need to capture the dynamic interaction between charged particles and electromagnetic fields, in order to simulate the time-evolution of the particle phase-space. The PIC simulations typically use FDTD (structured grid) or FETD (unstructured grid) for the field update. However, the computational challenges in PIC simulations are further compounded by the several thousands of charged particles (superparticles) which need to simulated individually at

each timestep. These observations dovetail with the recent surge in data-driven modeling in the scientific community [25, 26] due to advances in "big data" processing [27, 28, 29], hardware capabilities [30, 31], and data-driven (including machine learning based) techniques [32, 33, 34]. Part of the motivation for some of these developments comes from the fact that oftentimes there is only partial information (or no information) available regarding the underlying physics or guiding principles of the problem under study [35, 36]. In other cases, the dynamic system is so large and involves so many uncorrelated variables that modeling from first principles is simply not feasible (such as pandemic dynamics). In such cases, data-driven methods present a viable option. Finally, in disciplines such as computational fluid dynamics (CFD), CPP and CEM, the underlying physics is largely well understood, however majority of the problems can only be solved numerically requiring extensive computational resources.

The concept of a data-driven reduced-order model (ROM) is to leverage data from one or more high-fidelity simulations to create a low-dimensional model that captures the dominant behavior of the system. These ROMs can then be used for efficient model interpolation or extrapolation. Given our focus on accelerating high-fidelity time-domain simulations, we concentrate on time-extrapolatory ROMs applicable on a case-by-case basis. The process involves running the high-fidelity time-domain solver for a brief period, constructing a data-driven ROM from the collected data, and then using the ROM to predict future solutions with minimal

computational cost. This approach is depicted in Fig. 1.1. The primary benefit of this methodology is addressing one of the key challenges in scientific machine learning (SciML), namely the generation of training data. Typical interpolatory parametric models, particularly those based on neural networks, require thousands of training samples. Executing thousands of high-fidelity simulations, each potentially taking several hours to days, is impractical even with parallel computing architectures. In our approach (Fig. 1.1), the simulation proceeds as usual until an on-the-fly algorithm signals that sufficient data has been generated. At this point, the high-fidelity simulation stops, and the collected data is used to build a ROM, which then predicts future solutions. The downside is that the high-fidelity simulation must still run for a short duration, which can range from several minutes to days. However, this method can significantly reduce overall simulation time. For example, a simulation initially requiring a week could be reduced to a few days. In this thesis, we specifically explore Koopman ROMs, which are linear and advantageous for control applications.

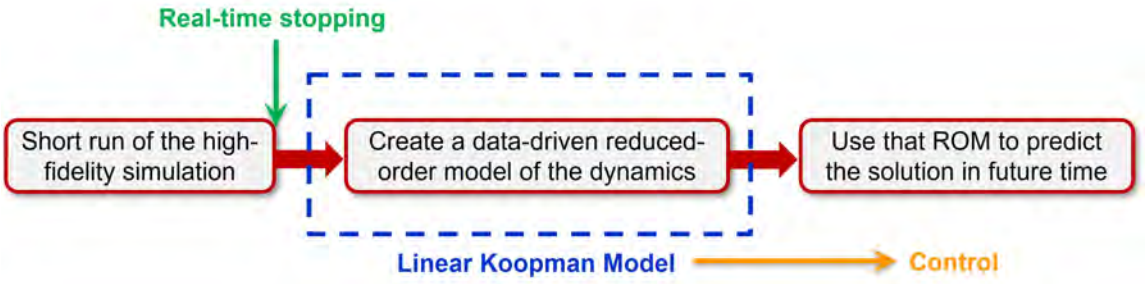


Figure 1.1: Schematic illustration of the time-extrapolatory Koopman ROMs for accelerating high-fidelity time-domain simulations. The high-fidelity simulation is run for a short time. Based on the available data a data-driven ROM is built, which is used for future prediction. We want to build a linear Koopman model which will be beneficial for control application as well.

Koopman operator theory provides an operator theoretic approach for modeling nonlinear dynamical systems [37]. The key idea is to transform the original state space to a *feature* or *observable* space where the dynamics is linear. For linear dynamical dynamical systems, this transformation is essentially the identity transformation. The key challenge lies in identifying such as transformation which leads to a finite-dimensional Koopman observable space. Since we are dealing with high-dimensional systems (large number of mesh elements in high-fidelity simulations), we would ideally prefer this Koopman invariant subspace to have a dimension smaller than the original state space. There are several data-driven techniques for approximating such Koopman reduced-order models (Koopman ROMs), most popular being the dynamic mode decomposition (DMD) [38, 39, 40, 41] and Koopman autoencoders

(KAEs) [42, 43, 44]. Given a short time-window of high-fidelity (spatio-temporal) data, DMD extracts a set of DMD modes (spatial features), DMD frequencies (temporal signatures), and modal amplitudes to approximate the underlying dynamics through a reduced-order linear system. Although originally developed for modeling fluid flows, DMD has also been recently applied in the successful modeling of electromagnetic radiation [45] and nonlinear evolution of plasma-wave systems [46, 47]. The popularity of DMD across several fields from fluid dynamics to plasma physics is rooted in its close relation to the Koopman operator theory [48], which is a powerful framework for the analysis of very generic, nonlinear dynamical systems. The Koopman autoencoder exploits the expressivity of neural networks to find the low dimensional Koopman observable space. The KAE architecture resembles the traditional autoencoder architecture with an extra linear layer between the encoder and decoder layers to advance the dynamics in a linear fashion [42, 43, 44, 2, 49]. We will explore the effectiveness of DMD and KAE for modeling electromagnetic fields in cavities [50] and self-felds and currents in kinetic plasma simulations [47, 51, 49].

1.2 Contribution of This Dissertation

The primary contributions of this dissertation are following:

- We establish the suitability of Hankel-DMD formulation with time-delayed fields for modeling electromagnetic resonances. We identify the condition for the one-to-one correspondence between the numerically extracted DMD modes

and physical cavity modes, and discuss the scenarios under which these conditions are not observed. This helps build confidence in the data-driven extrapolation performed by the DMD, unlike most of the other data-driven methods which are blackbox (i.e. not interpretable) in nature. The DMD modes and frequencies also provide a convenient way to extract the spatial pattern of resonance modes and corresponding resonance frequencies in complex cavities from limited time-series data with no need for time consuming frequency-domain eigenmode solvers. We also demonstrate the use of the extracted DMD frequencies to directly evaluate the $Q-$factor corresponding to particular modes.

- We analyze DMD's time-extrapolation capabilities for reducing the computation time of time-domain cavity simulations by accurate prediction of fields in future time. This not only helps in rapid late-time query of fields, but also accelerates the frequency domain analysis which typically requires long time-series data for high-resolution fast-Fourier transform (FFT).

- We develop DMD-based reduced-order model for modeling the self fields in EMPIC simulations and analyze DMD's ability to provide insights into underlying key patterns as well as its prediction ability in future time. We also study the effect of DMD extrapolated fields on the superparticle dynamics, i.e. phase-space distribution of the charged particles.

- We develop DMD-EMPIC, an intrusive physics-informed reduced-order model of space-charge dynamics, to accelerate EMPIC simulations exhibiting self-repeating behavior, such as limit cycle behavior.

- We develop on-the-fly (online) DMD algorithms for real-time detection of when to stop the high-fidelity simulations such as FDTD/FETD or EMPIC, indicating that sufficient simulation data has been collected for long-term extrapolation using data-driven techniques like DMD.

- We develop on-the-fly tracking algorithm for DMD modes and frequencies as the system transitions from transient to equilibrium or *quasi* (or *pseudo*) equilibrium states. Tracking DMD modes and frequencies helps analyze how transient DMD features evolve into equilibrium features and identify the optimal stopping point for high-fidelity simulations in real-time.

1.3 Organization of This Dissertation

This dissertation is organized as follows:

Chapter 2 introduces the Koopman operator theory for dynamical systems and explores data-driven Koopman reduced-order models, such as DMD and KAE. In Chapter 3, we present the proposed on-the-fly DMD algorithms for real-time termination of high-fidelity time-domain simulations. This chapter discusses two algorithms: one based on DMD mode tracking and the other on the convergence of DMD predictions.Chapter 4 examines the effectiveness of DMD in modeling electric fields within

electromagnetic cavities, especially from partial observations. We demonstrate how combining online (on-the-fly) and offline DMD can expedite FDTD cavity simulations. In Chapter 5, we investigate the use of DMD for modeling and predicting nonlinear self-fields from Electromagnetic Particle-in-Cell (EMPIC) plasma simulations. This chapter also discusses the impact of DMD-predicted self-fields on (super)particle dynamics. Chapter 6 showcases how DMD modeling of current density can accelerate EMPIC simulations. Finally, Chapter 7 explores neural network-based Koopman reduced-order models, such as KAE, for modeling nonlinear current density in EMPIC simulations.

Chapter 2: Koopman Reduced-Order Models

B.O. Koopman in the early 1930s demonstrated that Hamiltonian flows governed by nonlinear dynamics can be analyzed via an infinite-dimensional linear operator on the Hilbert space of observable functions [37]. The spectral analysis of this infinite-dimensional linear operator, or the so called "Koopman operator theory" provides an operator theoretic understanding of dynamical systems. The Koopman operator theory essentially tells us that the state space corresponding to a *nonlinear* dynamical system can be transformed into an infinite-dimensional observable space where the dynamics is *linear*. Since the Koopman operator operates on functionals, it is infinite-dimensional in nature. In this chapter we will explore the basics of Koopman operator theory and finite-dimensional approximation of Koopman operator via data-driven techniques like DMD and KAE. This chapter is primarily based on [49, 52, 50].

2.1 Koopman Operator Theory

Let us consider a discrete time nonlinear dynamical system evolving on a N-dimensional manifold $\mathcal{M}$ [40]. The discrete time evolution of state $\mathbf{x}$ is given by the

flow map $F : \mathcal{M} \mapsto \mathcal{M}$, where $\mathcal{M}$ is the N-dimensional manifold ($\mathbf{x} = [x_1\ x_2\ \dots\ x_N]^{\mathrm{T}}$)

$$\mathbf{x}^{(n+1)} = F(\mathbf{x}^{(n)}). \tag{2.1}$$

The discrete-time Koopman operator, denoted by $\mathcal{K}$, operates on the Hilbert space of observable functions $g(\mathbf{x})$ (observable functions take the state as input and produces a scalar valued output, $g : \mathcal{M} \mapsto \mathbb{C}$), such that

$$\mathcal{K}g(\mathbf{x}^{(n)}) = g(F(\mathbf{x}^{(n)})) = g(\mathbf{x}^{(n+1)}). \tag{2.2}$$

In other words, $\mathcal{K}g = g \circ F$, where "$\circ$" represents the composition of functions. The infinite dimensionality of the Koopman operator stems from the fact that the observable functions $g(\mathbf{x})$ which satisfy (2.2) form an infinite-dimensional Hilbert space, i.e. $g(\mathbf{x}) = \sum_{i=1}^{\infty} \alpha_i \zeta_i(\mathbf{x})$, where $\zeta_i(\mathbf{x})$ are the basis observable functions and $\alpha_i \in \mathbb{C}$. Naturally, the eigenvalues $\lambda_i \in \mathbb{C}$ and the eigenfunctions $\phi_i : \mathcal{M} \mapsto \mathbb{C}$ associated with $\mathcal{K}$ are infinite in number such that $\mathcal{K}\phi_i(\mathbf{x}) = \lambda_i \phi_i(\mathbf{x})$, $i = 1, 2 \dots \infty$. Given that the Koopman eigenfunctions span the observable functions, one can write $g(\mathbf{x}) = \sum_{i=1}^{\infty} \phi_i(\mathbf{x}) v_i$, where $v_i \in \mathbb{C}$. In a similar fashion, a vector valued observable $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x})\ g_2(\mathbf{x})\ \dots\ g_p(\mathbf{x})]^{\mathrm{T}}$ can be expressed as $\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{\infty} \phi_i(\mathbf{x}) \mathbf{v}_i$, where $\mathbf{v}_i = [v_{1i}\ v_{2i}\ \dots\ v_{pi}]^{\mathrm{T}}$ are defined as Koopman modes. Using the initial state, $\mathbf{x}^{(0)}$ along with linearity of the Koopman operator, we have

$$\mathbf{g}(\mathbf{x}^{(n)}) = \mathcal{K}^n \mathbf{g}(\mathbf{x}^{(0)}) = \mathcal{K}^n \sum_{i=1}^{\infty} \phi_i(\mathbf{x}^{(0)}) \mathbf{v}_i = \sum_{i=1}^{\infty} \mathcal{K}^n \phi_i(\mathbf{x}^{(0)}) \mathbf{v}_i = \sum_{i=1}^{\infty} \lambda_i^n \phi_i(\mathbf{x}^{(0)}) \mathbf{v}_i \quad . \tag{2.3}$$

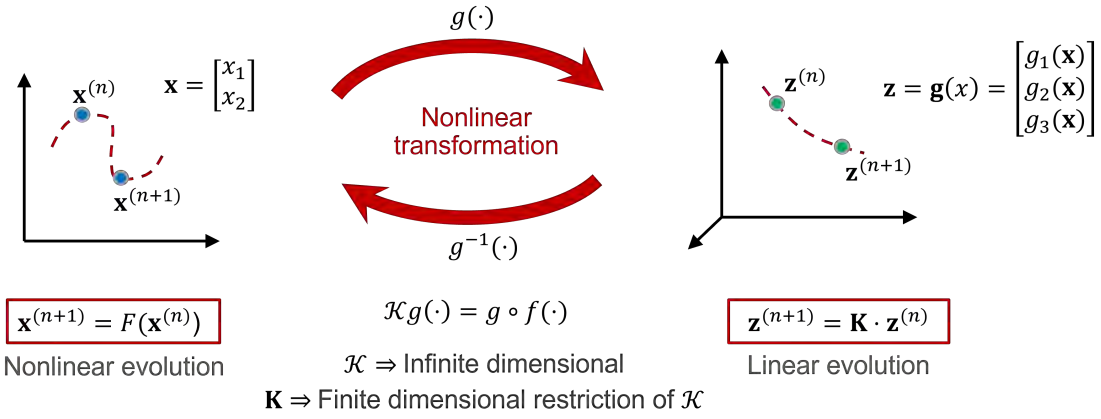


Figure 2.1: Schematic illustration of the Koopman approach for modeling the time evolution of nonlinear systems. The nonlinear dynamical system (on left) is transformed to a feature space (on right) where the dynamics is linear.

Although (2.3) helps us decompose the underlying dynamics in terms of Koopman modes $\mathbf{v}_i$ and corresponding eigenvalues λ_i, from a computational perspective it is not practical to work in an infinite-dimensional space. However, for a finite-dimensional Koopman invariant subspace, it is possible to find a finite-dimensional matrix representation $\mathbf{K}$ (Fig. 2.1), of the infinite-dimensional Koopman operator $\mathcal{K}$ [53]. Consider a class of observable functions $g_j(\mathbf{x})$ belonging to the Koopman invariant subspace $\mathcal{S}_\mathcal{K}$ of dimension $d_\mathcal{S}$. As the subspace is $\mathcal{K}$-invariant, $g_j(\mathbf{x}) \in \mathcal{S}_\mathcal{K} \implies \mathcal{K}g_j(\mathbf{x}) \in \mathcal{S}_\mathcal{K}$. Furthermore, let us assume some basis Koopman observables

$s_i(\mathbf{x})$ spanning the subspace $\mathcal{S}_\mathcal{K}$. For some $\alpha_{ji} \in \mathbb{C}$, we can write

$$g_j(\mathbf{x}^{(n)}) = \sum_{i=1}^{d_\mathcal{S}} \alpha_{ji} s_i(\mathbf{x}^{(n)}) \tag{2.4a}$$

$$\mathcal{K} g_j(\mathbf{x}^{(n)}) = g_j(\mathbf{x}^{(n+1)}) = \sum_{i=1}^{d_\mathcal{S}} \alpha_{ji} s_i(\mathbf{x}^{(n+1)}) = \sum_{i=1}^{d_\mathcal{S}} \beta_{ji} s_i(\mathbf{x}^{(n)}) \ , \tag{2.4b}$$

$$\begin{bmatrix} g_1(\mathbf{x}^{(n)}) \\ g_2(\mathbf{x}^{(n)}) \\ \vdots \\ g_p(\mathbf{x}^{(n)}) \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1d_\mathcal{S}} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2d_\mathcal{S}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{p1} & \alpha_{p2} & \dots & \alpha_{pd_\mathcal{S}} \end{bmatrix} \begin{bmatrix} s_1(\mathbf{x}^{(n)}) \\ s_2(\mathbf{x}^{(n)}) \\ \vdots \\ s_{d_\mathcal{S}}(\mathbf{x}^{(n)}) \end{bmatrix} \ , \tag{2.5}$$

$$\Rightarrow \mathbf{g}(\mathbf{x}^{(n)}) = \mathbf{A}\ \mathbf{s}(\mathbf{x}^{(n)}) \quad , \quad [\text{where } [\mathbf{A}]_{j,i} = \alpha_{ij}] \tag{2.6}$$

In a similar fashion, we can write $\mathbf{g}(\mathbf{x}^{(n+1)}) = \mathbf{B}\ \mathbf{s}(\mathbf{x}^{(n)})$ with $[\mathbf{B}]_{j,i} = \beta_{ji}$. Assuming the existence of $\mathbf{K}$ such that

$$\mathbf{g}(\mathbf{x}^{(n+1)}) = \mathbf{K}\ \mathbf{g}(\mathbf{x}^{(n)}) \tag{2.7}$$

$$\Rightarrow \mathbf{B}\ \mathbf{s}(\mathbf{x}^{(n)}) = \mathbf{K}\mathbf{A}\ \mathbf{s}(\mathbf{x}^{(n)})$$

$$\Rightarrow (\mathbf{B} - \mathbf{K}\mathbf{A})\ \mathbf{s} = \mathbf{0} \tag{2.8}$$

Eq. (2.8) will hold for any set of basis s_i, if $\mathbf{B} = \mathbf{K}\mathbf{A}$. There will exist a unique $\mathbf{K}$ if $\mathbf{A}$ is square and full-rank which necessiates the observables $g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_p(\mathbf{x})$ to be linearly independent and span $\mathcal{S}_\mathcal{K}$ with $p = d_\mathcal{S}$. Clearly, a careful choice of the observable functions $g(\mathbf{x})$ is crucial for the effectiveness of the Koopman approach, and the development of algorithms or heuristics for identifying optimal observables is still an active area of research. The primary takeaway is that, with an appropriate

choice of observables, we can bypass the infinite sum in (2.3) and make the problem tractable.

However, there is a last piece of the puzzle that needs to be addressed. The Koopman mode decomposition (2.3) consists of the eigenvalues λ_i, eigenfunctions $\phi_i(\mathbf{x})$, and Koopman modes $\mathbf{v}_i$ corresponding to the infinite-dimensional operator $\mathcal{K}$. We need to establish how these quantities relate to the eigenvalues λ_{ki} and eigenvectors of the finite-dimensional $\mathbf{K}$. For the linear dynamical system,

$$\mathbf{g}(\mathbf{x}^{(n+1)}) = \mathbf{K} \cdot \mathbf{g}(\mathbf{x}^{(n)}) \tag{2.9}$$

$$\mathbf{K}\mathbf{u}_i = \lambda_{ki}\mathbf{u}_i \quad , \quad \mathbf{K}^*\mathbf{w}_i = \bar{\lambda}_{ki}\mathbf{w}_i \quad , \quad [i = 1, 2, \ldots, p], \tag{2.10}$$

where $\mathbf{u}_i$ and $\mathbf{w}_i$ are the right and left eigenvectors of $\mathbf{K}$ respectively. The " $*$ " denotes complex conjugate transpose and the overbar denotes complex conjugate. The $\mathbf{w}_i$ are defined in such a way that $\langle \mathbf{u}_i, \mathbf{w}_j \rangle = \delta_{ij}$, where $\langle \cdot, \cdot \rangle$ denotes inner product in $\mathbb{C}^p$. Let us define a scalar valued observable $\psi_i(\mathbf{x})$ such that

$$\psi_i(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}), \mathbf{w}_i \rangle = \bar{w}_{i1} g_1(\mathbf{x}) + \bar{w}_{i2} g_2(\mathbf{x}) + \ldots + \bar{w}_{ip} g_p(\mathbf{x}) \quad , \quad [i = 1, 2, \ldots, p], \tag{2.11}$$

where $\mathbf{w}_i = [w_{i1}\ w_{i2}\ \ldots\ w_{ip}]^{\mathrm{T}}$. Note that $\psi_i(\mathbf{x})$ is a linear combination of Koopman observables. As a result, from linearity of the Koopman operator, $\psi_i(\mathbf{x})$ is also a

Koopman observable. Thus, $\mathcal{K}$ upon operating on $\psi_i(\mathbf{x}^{(n)})$ results in

$$\begin{aligned}\mathcal{K}\psi_i(\mathbf{x}^{(n)}) = \psi_i(\mathbf{x}^{(n+1)}) &= \langle \mathbf{g}(\mathbf{x}^{(n+1)}), \mathbf{w}_i \rangle = \langle \mathbf{K}\mathbf{g}(\mathbf{x}^{(n)}), \mathbf{w}_i \rangle \\ &= \langle \mathbf{g}(\mathbf{x}^{(n)}), \mathbf{K}^*\mathbf{w}_i \rangle = \langle \mathbf{g}(\mathbf{x}^{(n)}), \bar{\lambda}_{ki}\mathbf{w}_i \rangle \\ &\Rightarrow \mathcal{K}\psi_i(\mathbf{x}^{(n)}) = \lambda_{ki}\langle \mathbf{g}(\mathbf{x}^{(n)}), \mathbf{w}_i \rangle \\ &\Rightarrow \mathcal{K}\psi_i(\mathbf{x}^{(n)}) = \lambda_{ki}\psi_i(\mathbf{x}^{(n)}). \end{aligned} \tag{2.12}$$

It turns out that eigenvalues of $\mathbf{K}$ are also the eigenvalues of $\mathcal{K}$ ($\lambda_{ki} = \lambda_i$) and $\psi_i(\mathbf{x})$ are nothing but the eigenfunctions $\phi_i(\mathbf{x})$,

$$\phi_i(\mathbf{x}) = \psi_i(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}), \mathbf{w}_i \rangle = \bar{w}_{i1}g_1(\mathbf{x}) + \bar{w}_{i2}g_2(\mathbf{x}) + \ldots + \bar{w}_{ip}g_p(\mathbf{x}). \tag{2.13}$$

In other words, $\phi_i(\mathbf{x}) \in \text{span}\{g_j(\mathbf{x})\}$, $j = 1, 2, \ldots, p$ [40]. However, unlike $\mathbf{K}$, $\mathcal{K}$ has infinite eigenvalues. Through simple algebraic manipulations we can show that λ_i^k ($k \in \mathbb{Z}^+$) is also an eigenvalue of $\mathcal{K}$ with eigenfunction $\phi_i^k(\mathbf{x})$ [48]. Up to this point, no assumptions have been made about $\mathbf{K}$, except that it exists. Let us assume that $\mathbf{K}$ has a full set of eigenvectors, i.e. $\mathbf{u}_i$ are linearly independent and form a basis. With this assumption, we can write the solution for a linear system as follows

$$\begin{aligned}\mathbf{g}(\mathbf{x}^{(n)}) = \sum_{i=1}^{p} \langle \mathbf{g}(\mathbf{x}^{(n)}), \mathbf{w}_i \rangle \mathbf{u}_i &= \sum_{i=1}^{p} \langle \mathbf{K}^n \mathbf{g}(\mathbf{x}^{(0)}), \mathbf{w}_i \rangle \mathbf{u}_i \\ &= \sum_{i=1}^{p} \langle \mathbf{g}(\mathbf{x}^{(0)}), (\mathbf{K}^*)^n \mathbf{w}_i \rangle \mathbf{u}_i \end{aligned} \tag{2.14}$$

$$\Rightarrow \ \mathbf{g}(\mathbf{x}^{(n)}) = \sum_{i=1}^{p} \langle \mathbf{g}(\mathbf{x}^{(0)}), (\mathbf{K}^*)^n \mathbf{w}_i \rangle \mathbf{u}_i = \sum_{i=1}^{p} \langle \mathbf{g}(\mathbf{x}^{(0)}), (\bar{\lambda}_i)^n \mathbf{w}_i \rangle \mathbf{u}_i$$

$$= \sum_{i=1}^{p} \lambda_i^n \langle \mathbf{g}(\mathbf{x}^{(0)}), \mathbf{w}_i \rangle \mathbf{u}_i$$

$$\Rightarrow \ \mathbf{g}(\mathbf{x}^{(n)}) = \sum_{i=1}^{p} \lambda_i^n \phi_i(\mathbf{x}^{(0)}) \mathbf{u}_i \ \ , \ \ \text{[Using (2.13)]} \tag{2.15}$$

Eqs. (2.3) and (2.15) don't contradict each other if we consider $\mathbf{u}_i$, the right eigenvectors of $\mathbf{K}$, as the Koopman modes $\mathbf{v}_i$. In that case (2.15) is just the finite truncation of infinite summation in (2.3). Eq. (2.15) may also be interpreted as the set of eigenfunctions $\phi_i(\mathbf{x})$, $[i = 1, 2, \ldots, p]$ forming the finite-dimensional $\mathcal{K}$-invariant subspace in question. Note that this is in agreement with the initial assumption that Koopman eigenfunctions span the observable space. In summary, in a suitable observable space where the dynamics is linear with $\mathbf{g}(\mathbf{x}^{(n+1)}) = \mathbf{K} \cdot \mathbf{g}(\mathbf{x}^{(n)})$, the following points can be noted:

- The eigenvalues of $\mathbf{K}$ are the eigenvalues of the linear infinite-dimensional Koopman operator $\mathcal{K}$.

- If $\mathbf{K}$ has full set of eigenvectors, then the (right) eigenvectors represent Koopman modes associated with $\mathcal{K}$.

- If $\mathbf{K}$ does not have a full set of eigenvectors, then it remains an open question whether they represent Koopman modes [39].

2.2 Dynamic Mode Decomposition (DMD)

Let us consider a discrete-time dynamical system

$$\mathbf{x}^{(n+1)} = F(\mathbf{x}^{(n)}), \tag{2.16}$$

where $\mathbf{x}^{(n)}$ is the state of the system at n^{th} time step with step size of Δt . The state $\mathbf{x}$ evolves on a $N-$dimensional manifold $\mathcal{M}$ according to the flow-map $F : \mathcal{M} \mapsto \mathcal{M}$. In general, for practical applications, $\mathbf{x} \in \mathbb{R}^N$ with $F : \mathbb{R}^N \mapsto \mathbb{R}^N$. The first step in the DMD process involves snapshot collection and formation of the snapshot matrices. The state $\mathbf{x}$ is sampled (snapshots) at every Δ_n^{th} time step starting from $(k-1)^{\text{th}}$ time sample at $n = n_{k-1} = (k-1)\Delta_n$ or equivalently $t = t_{k-1} = (k-1)\Delta_n\Delta t$. Total $(l+1)$ snapshots are collected inside the harvesting window (DMD window) spanning from n_{k-1} to $n_{k-1}+l\Delta_n$. With $\mathbf{x}_k = \mathbf{x}^{(n_k)}$, the snapshot columns are stacked to form the snapshot matrices $\mathbf{X}_{k-1}$ and $\mathbf{X}_k$,

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_k & \mathbf{x}_{k+1} & . & . & . & \mathbf{x}_{k+l-1} \end{bmatrix}, \tag{2.17}$$

$$\mathbf{X}_{k-1} = \begin{bmatrix} \mathbf{x}_{k-1} & \mathbf{x}_k & . & . & . & \mathbf{x}_{k+l-2} \end{bmatrix}, \tag{2.18}$$

$\mathbf{X}_k$ differs from $\mathbf{X}_{k-1}$ by shift of one snapshot. DMD assumes a linear relationship between the snapshot matrices and proceeds to find a least-squares solution to the problem $\mathbf{X}_k \approx \mathbf{A}\mathbf{X}_{k-1}$, given by $\mathbf{A} \approx \mathbf{X}_k\mathbf{X}_{k-1}^{\dagger}$, where '†' denotes the Moore-Penrose pseudoinverse. In order to calculate the pseudoinverse, singular value decomposition (SVD) is performed on $\mathbf{X}_{k-1}$,

$$\mathbf{X}_{k-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \tag{2.19}$$

where superscript 'H' denotes the complex-conjugate transpose. The columns of $\mathbf{U}_{N\times l}$ capture the dominant spatial pattern ranked by their singular values which are essentially the elements of the diagonal matrix $\mathbf{\Sigma}_{l\times l}$, and the rows of $\mathbf{V}^H_{N\times l}$ are corresponding time signatures. Sharp decay in the magnitudes of the singular values acros the singular value spectrum indicates that the first few columns of $\mathbf{U}$ (associated to the dominant singular values) are sufficient to accurately capture the subspace over which the relevant signal dynamics is evolving, thus confirming the existence of an underlying lower-dimensional structure. We retain the first r columns of $\mathbf{U}$, first r singular values of $\mathbf{\Sigma}$ and first r rows of $\mathbf{V}^H$, resulting in the *reduced* SVD matrices $\tilde{\mathbf{U}}$,$\tilde{\mathbf{\Sigma}}$ and $\tilde{\mathbf{V}}^H$, so that $\mathbf{X}_{k-1} \approx \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^H$. With $\mathbf{X}^{\dagger}_{k-1} \approx \tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^H$, we can write $\mathbf{A} \approx \mathbf{X}_k\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^H$. Note that $\mathbf{A}_{N\times N}$ is a large matrix (N can be in thousands or millions), making its eigendecomposition invariably computationally expensive. Hence, $\mathbf{A}$ is projected to a lower dimensional subspace spanned by the columns of $\tilde{\mathbf{U}}$ as $\tilde{\mathbf{A}} = \tilde{\mathbf{U}}^H\mathbf{A}\tilde{\mathbf{U}}$, where $\tilde{\mathbf{A}}$ is a $r \times r$ matrix with usually $r \ll N$. We can write

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}^H\mathbf{X}_k\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}. \tag{2.20}$$

From eigendecomposition of $\tilde{\mathbf{A}}$ we get

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}, \tag{2.21}$$

where columns of $\mathbf{W}$ represent the right eigenvectors of $\tilde{\mathbf{A}}$ and the diagonal matrix $\mathbf{\Lambda}$ contains its eigenvalues (DMD eigenvalues $\lambda_i,\ i = 1, 2, \ldots, r$) which are an adequate approximation of the eigenvalues of $\mathbf{A}$. The eigenvectors of $\mathbf{A}$, approximated by the

columns of $\mathbf{\Phi}$, are also known as the *exact* DMD modes [39, 40],

$$\mathbf{\Phi} = \mathbf{X}_k \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \mathbf{W}. \tag{2.22}$$

The continuous time DMD reconstruction ($\hat{\mathbf{x}}$) at any time $t \geq t_0$ ($= n_{k-1}\Delta t$) is given by

$$\mathbf{x}(t) \approx \hat{\mathbf{x}}(t) = \sum_{i=1}^{r} \vartheta_i \boldsymbol{\phi}_i e^{\widehat{\omega}_i (t-t_0)}, \tag{2.23}$$

where $\boldsymbol{\phi}_i$ are the columns of $\mathbf{\Phi}$, also known as the (*exact*) DMD modes, $\widehat{\omega}_i$ are the DMD frequencies and ϑ_i are the DMD modal amplitudes. DMD frequencies are related to DMD eigenvalues by $\widehat{\omega}_i = \ln(\lambda_i)/\Delta_t$, where Δ_t is the time interval between two consecutive snapshots ($\Delta_t = \Delta_n \Delta t$). The ϑ_i can be calculated by solving an optimization problem as described in [54]. Note that for real data, the triplets $\{\boldsymbol{\phi}_i, \widehat{\omega}_i, \vartheta_i\}$ ($i = 1, 2, \ldots, r$) appear in complex-conjugate pairs. So, for real data (2.23) can be written as

$$\widehat{\mathbf{x}}(t) = \sum_{\widehat{m}=1}^{\widehat{M}} (\vartheta_{\widehat{m}} \boldsymbol{\phi}_{\widehat{m}} e^{\widehat{\omega}_{\widehat{m}}(t-t_0)} + \vartheta^*_{\widehat{m}} \boldsymbol{\phi}^*_{\widehat{m}} e^{\widehat{\omega}^*_{\widehat{m}}(t-t_0)}), \tag{2.24}$$

where the superscript '$*$' denotes the complex-conjugate. For purely real modes, two terms in (2.24) can be combined to a single term ($2\widehat{M} \geq r$). Since we are dealing with the real data, from this point onward by i^{th} DMD mode, we will essentially refer to the i^{th} complex-conjugate pair of DMD modes. The schematic illustration of the DMD process is shown in Fig. 2.2.

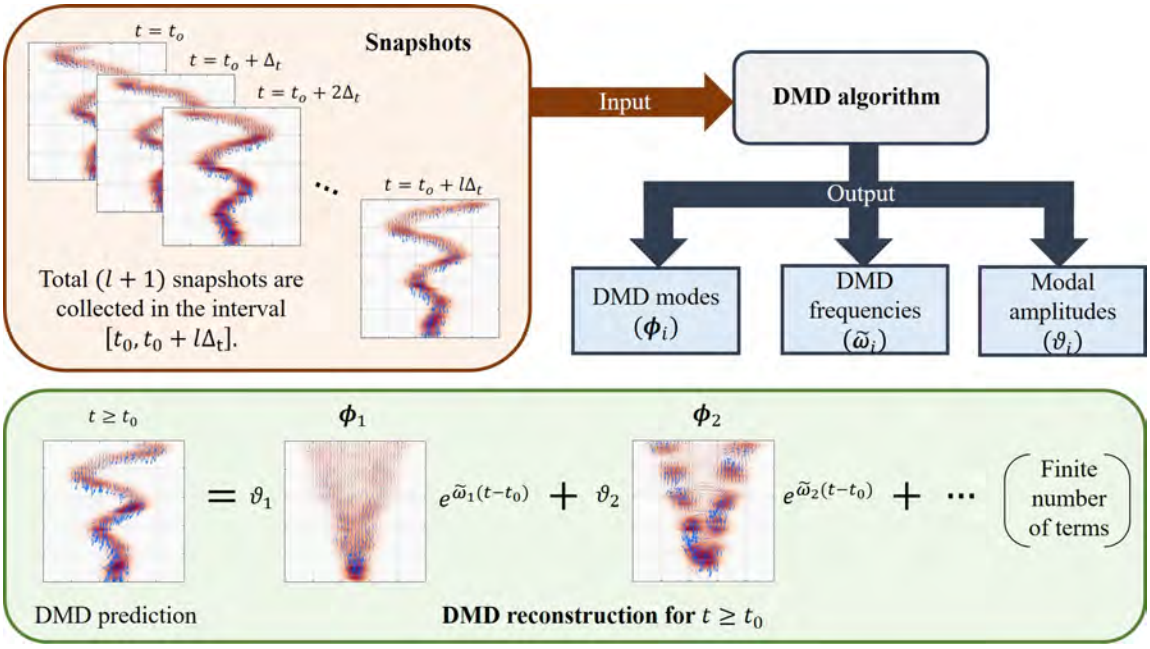


Figure 2.2: Schematic illustration of the DMD method for modeling current density of an oscillating electron beam.

2.2.1 Hankel-DMD

The usual DMD fails to resolve the dynamics when the spectral complexity M_s is larger than the spatial complexity N of the observed system [55, 56]. The spectral complexity M_s depends on the frequency content of the system and essentially equals to the number of terms required to accurately describe the dynamics as summation of complex exponentials (2.23). This situation typically arises when $N < l$, where the SVD matrices $\mathbf{U}_{N\times l}, \mathbf{\Sigma}_{N\times N}$ and $\mathbf{V}_{l\times N}$ can have maximum rank of N, restricting $r \leq N$. Since $M_s > N \geq r$, r number of DMD frequencies are not sufficient to capture

the system's temporal evolution. There can be other scenarios [56] as well where the usual DMD fails. We will discuss those issues in details in the next section in the context of electromagnetic resonances. These issues can be addressed by assuming a higher order linear system of the form

$$\mathbf{X}_k \approx \mathbf{A}_1\mathbf{X}_{k-1} + \mathbf{A}_2\mathbf{X}_{k-2} + \ldots + \mathbf{A}_d\mathbf{X}_{k-d}. \tag{2.25}$$

The equivalent first-order problem of (2.25) can be written as,

$$[\![\mathbf{X}]\!]_k = [\![\mathbf{A}]\!][\![\mathbf{X}]\!]_{k-1} \tag{2.26}$$

$$\Rightarrow \begin{bmatrix} \mathbf{X}_{k-d+1} \\ \mathbf{X}_{k-d+2} \\ \vdots \\ \mathbf{X}_{k-1} \\ \mathbf{X}_k \end{bmatrix} \approx \begin{bmatrix} \mathbf{0}_N & \mathbf{I}_N & \ldots & \mathbf{0}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{0}_N & \ldots & \mathbf{0}_N & \mathbf{0}_N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_N & \mathbf{0}_N & \ldots & \mathbf{0}_N & \mathbf{I}_N \\ \mathbf{A}_d & \mathbf{A}_{d-1} & \ldots & \mathbf{A}_2 & \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_{k-d} \\ \mathbf{X}_{k-d+1} \\ \vdots \\ \mathbf{X}_{k-2} \\ \mathbf{X}_{k-1} \end{bmatrix}, \tag{2.27}$$

where $\mathbf{0}_N$ and $\mathbf{I}_N$ are respectively zero and identity matrices of dimension $N \times N$ with $[\![\cdot]\!]$ indicating the stacked quantities. The Hankel-DMD follows the same steps of usual DMD, but with following exceptions:

- Instead of $\mathbf{X}_k$ and $\mathbf{X}_{k-1}$, $[\![X]\!]_k$ and $[\![X]\!]_{k-1}$ are the new snapshot matrices.
- First N rows of $\mathbf{\Phi}$ in (2.22) are retained to obtain the DMD modes in original state-space dimension, i.e. $\mathbf{\Phi}_H = \mathbf{\Phi}_{[1:N,:]}$. The final reconstruction is same as (2.23) where $\boldsymbol{\phi}_i$ are the columns of $\mathbf{\Phi}_H$.

For a dynamical system exactly satisfying $\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1}$, $\mathbf{A}_1 = \mathbf{A}$, $\mathbf{A}_2 = \mathbf{A}^2$, $\ldots$ $\mathbf{A}_d = \mathbf{A}^d$; both the usual and the Hankel-DMD produce same set of DMD

eigenvalues and modes (after truncation). Note that we use Hankel-DMD for both on-the-fly and offline application.

2.3 Koopman Autoencoders (KAEs)

In what follows, we assume that the reader is familiar with basic concepts of neural networks and we keep the associated discussion very brief. In a nutshell, given a collection of input $\mathbf{x}$ and output $\mathbf{y}$ dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{s}$, a neural network "learns" the mapping from input to output through an optimization process and predicts $\mathbf{y}_i$ for a given $\mathbf{x}_i$, outside the training dataset, i.e. for $i > s$. Autoencoders [57, 58] can be seen as a special type of feed-forward fully-connected neural networks which have essentially two separate components: an *i)* encoder and a *ii)* decoder. The encoder component maps (encodes) the input to a lower dimensional space and the decoder decodes it back to the original high-dimensional space. Autoencoders are primarily used in data compression to facilitate machine learning methods and to make efficient use of memory. As a result of its ability to compress data while retaining relevant information, autoencoders are a natural choice for nonlinear model order reduction.

In the context of Section 2.1, Koopman autoencoders (KAEs) seek to obtain a suitable vector-valued observable $\mathbf{z} = \mathbf{g}(\mathbf{x})$, such that $\mathbf{z}^{(n+1)} = \mathbf{K} \cdot \mathbf{z}^{(n)}$ as in (2.9). KAE models have been successfully implemented in modeling of fluids, sea-surface temperature [44, 43, 59] and many other complex nonlinear phenomena. KAE models have also been recently applied to plasma systems, in particular to model the evolution of plasma currents in EMPIC simulations [2]. In what follows, we will

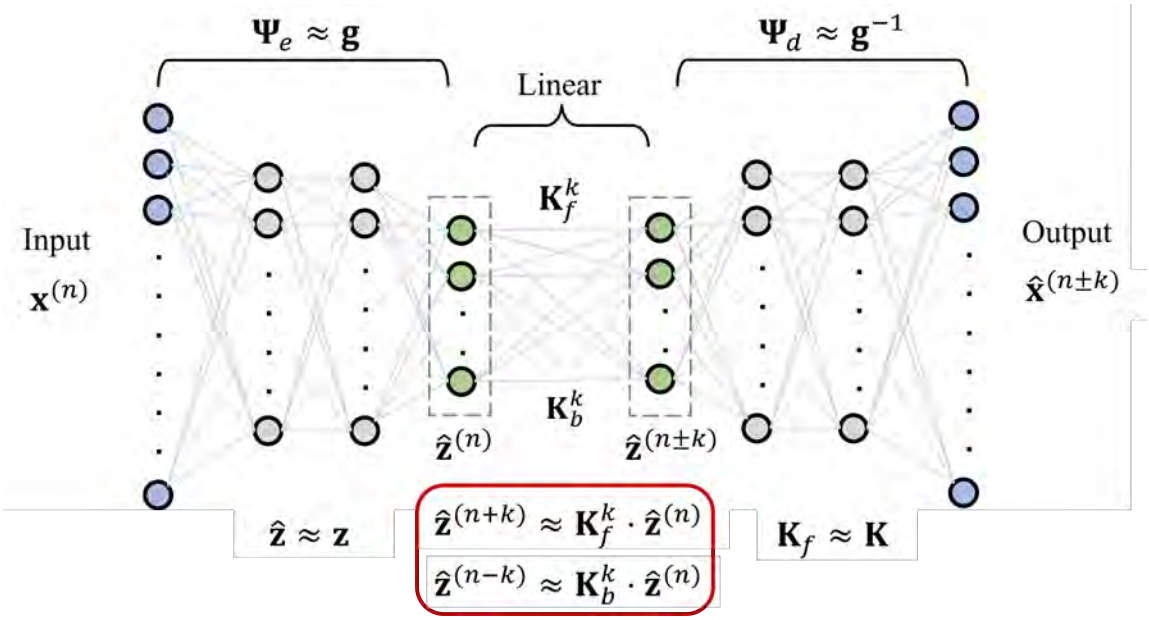


Figure 2.3: Schematic diagram of a consistent Koopman autoencoder architecture. The input and output layers are shown by the blue neurons (nodes) whereas the hidden layers are shown by gray neurons. The green neurons denote the encoded layer where it learns the (reduced-order) linear dynamics.

describe the basic aspects of an KAE architecture and provide some illustrative results of its application to the reduced order modeling of kinetic plasma problems.

Similar to an generic autoencoder, a KAE has an encoding $\mathbf{\Psi}_e$ and a decoding $\mathbf{\Psi}_d$ layer. In between those two, there is an extra linear $\mathbf{K}_f$ layer which approximates $\mathbf{K}$. In other words, the encoder mapping $\mathbf{\Psi}_e$ approximates the transformation $\mathbf{g}(\cdot)$ i.e., $\mathbf{\Psi}_e(\mathbf{x}) \approx \mathbf{g}(\mathbf{x})$.

Fig. 2.3 shows the KAE architecture. It should be noted that, in typical KAE architecture, the linear layer in the middle only approximates the forward dynamics $\mathbf{K}_f$ ($\approx \mathbf{K}$) [43]. However, we will focus here on an improved type of KAE architecture [44] where the backward dynamics $\mathbf{K}_b$ is also taken into consideration, such that $\mathbf{K}_f \cdot \mathbf{K}_b \approx \mathbf{I}$ (identity matrix). This consideration improves the stability of the solution and is referred to as consistent KAE [44] to distinguish it from the earlier, more traditional KAE. However, for brevity we will refer to it simply as KAE. The input and output layers (white) consist of $N_{in} = N_{out} = N$ neurons which are typically equal to the dimension of the state which we are attempting to model. Both the encoding and decoding layers have N_h neurons in the hidden (gray) layer. The "bottleneck" layer (or encoded layer, in black) has N_b neurons with $N_b \ll N$ (order-reduction). The linear layers are followed by a hyperbolic tangent activation layer except the bottleneck layer. The state at the n^{th} time step $\mathbf{x}^{(n)}$ is fed to the input layer, which is then encoded to $\mathbf{g}(\mathbf{x}^{(n)})$. The linear layer in the middle section of the network advances the dynamics in the forward (or backward) directions by k time steps to generate $\mathbf{g}(\mathbf{x}^{(n\pm k)})$, which is then decoded back to the original state space giving us $\hat{\mathbf{x}}^{(n\pm k)}$:

$$\mathbf{x}^{(n+k)} \approx \hat{\mathbf{x}}^{(n+k)} = \mathbf{\Psi}_d \circ \mathbf{K}_f^k \circ \mathbf{\Psi}_e(\mathbf{x}^{(n)}), \tag{2.28a}$$

$$\mathbf{x}^{(n-k)} \approx \hat{\mathbf{x}}^{(n-k)} = \mathbf{\Psi}_d \circ \mathbf{K}_b^k \circ \mathbf{\Psi}_e(\mathbf{x}^{(n)}). \tag{2.28b}$$

Here, "$\circ$" denotes the composition of two operators. The $\mathbf{\Psi}_e$ and $\mathbf{\Psi}_d$ are not exactly symmetric because $\mathbf{\Psi}_d$ has an extra nonlinear activation layer at the end. Before

feeding the autoencoder, the input data are scaled in the range $[-1, 1]$. Since the hyperbolic tangent activation function also varies from -1 to 1, the final activation layer for $\mathbf{\Psi}_d$ ensures that the output is in the desired range. Note that typically, data are sampled regularly across certain time steps Δ_n. In that case, $\mathbf{K}_f^k$ advances the dynamics by k time samples or $k\Delta_n$ time steps. In order to facilitate a concise explanation of the functionality of KAE, we assume $\Delta_n = 1$.

The autoencoder is trained to minimize the total loss function $\mathcal{L}_{tot}$ which consists of four separate components denoted as (i) identity loss $\mathcal{L}_{id}$, (ii) forward loss $\mathcal{L}_{fwd}$, (iii) backward loss $\mathcal{L}_{bwd}$, and (iv) consistency loss $\mathcal{L}_{con}$:

$$\mathcal{L}_{tot} = \gamma_{id}\mathcal{L}_{id} + \gamma_{fwd}\mathcal{L}_{fwd} + \gamma_{bwd}\mathcal{L}_{bwd} + \gamma_{con}\mathcal{L}_{con}, \tag{2.29}$$

where $\gamma_{id}, \gamma_{fwd}, \gamma_{bwd}$ and γ_{con} are the user defined weights. The identity loss $\mathcal{L}_{id}$ measures the autoencoder's ability to reconstruct the state as it is at a particular time step n by first encoding it to a lower dimensional subspace and then by decoding it back to the original state space. $\mathcal{L}_{id}$ can be defined as

$$\mathcal{L}_{id} = \frac{1}{2l}\sum_{n=1}^{l} ||\hat{\mathbf{x}}^{(n)} - \mathbf{x}^{(n)}||_2, \tag{2.30}$$

where l is number of time samples in the training data. The consistency loss $\mathcal{L}_{con}$ measures consistency of the matrices $\mathbf{K}_f$ and $\mathbf{K}_b$, i.e. how closely they follow the relation $\mathbf{K}_f \cdot \mathbf{K}_b \approx \mathbf{I}$. It can be expressed as

$$\mathcal{L}_{con} = \sum_{i=1}^{N_b} \frac{1}{2i}||\mathbf{K}_{b_{i\star}}\mathbf{K}_{f_{\star i}} - \mathbf{I}_{N_b}||_F + \frac{1}{2i}||\mathbf{K}_{f_{\star i}}\mathbf{K}_{b_{i\star}} - \mathbf{I}_{N_b}||_F, \tag{2.31}$$

where $\mathbf{K}_{b_{i\star}}$ is the upper i rows of $\mathbf{K}_b$, $\mathbf{K}_{f_{\star i}}$ is the i left most columns of $\mathbf{K}_f$, $||\cdot||_F$ denotes the Frobenius norm, and $\mathbf{I}_{N_b}$ is the identity matrix of dimension $N_b \times N_b$. The forward loss $\mathcal{L}_{fwd}$ measures the autoencoder's ability to reconstruct $\mathbf{x}^{(n+k)}$ by encoding $\mathbf{x}^{(n)}$, forwarding the dynamics by k time steps (or time samples) in encoded space and then decoding it back to the original state space. Mathematically, $\mathcal{L}_{fwd}$ can be expressed as

$$\mathcal{L}_{fwd} = \frac{1}{2k_m l} \sum_{k=1}^{k_m} \sum_{n=1}^{l} ||\hat{\mathbf{x}}^{(n+k)} - \mathbf{x}^{(n+k)}||_2^2, \tag{2.32}$$

where k_m is the maximum number of time steps utilized in the forward direction. Finally, the backward loss $\mathcal{L}_{bwd}$ plays a role similar to $\mathcal{L}_{fwd}$ except it deals with reconstruction in the backward direction, that is $\mathbf{x}^{(n-k)}$. As such, the backward loss is expressed as

$$\mathcal{L}_{bwd} = \frac{1}{2k_m l} \sum_{k=1}^{k_m} \sum_{n=1}^{l} ||\hat{\mathbf{x}}^{(n-k)} - \mathbf{x}^{(n-k)}||_2^2, \tag{2.33}$$

where k_m now represents the maximum number of time steps utilized in the backward direction. Note that for notational convenience we have explained the KAE operation in the above by assuming that each time step data corresponds to one sample in training data. However, as mentioned earlier, the input data does not need to correspond to the sampling of every consecutive time step. Instead, there might be regular or irregular intervals comprising several time steps between consecutive input data samples. Apart from the usual neural network training hyperparameters such as learning rate, training epochs etc., the tunable parameters $\gamma_{id}, \gamma_{fwd}, \gamma_{bwd}, \gamma_{con}, N_h$,

and N_b play a crucial role in determining the extrapolation accuracy of the KAE model.

2.4 Summary

In this chapter, we have described two Koopman ROMs: DMD and KAE. DMD is a linear model, whereas KAE is nonlinear. In subsequent chapters, we will explore how both methods, particularly the Hankel variant of DMD, can be applied to model linear and nonlinear electromagnetic systems. While DMD will be utilized for modeling both electromagnetic cavities and kinetic plasmas, we will explore KAE only for kinetic plasmas.

Chapter 3: On-the-fly Dynamic Mode Decomposition for Real-Time Detection of End of Transience

One of the key components of our approach (Fig. 1.1) is a real-time stopping criterion for the high-fidelity time-domain simulations. In order to exploit the time extrapolation ability of DMD for reducing computation cost of high-fidelity simulations, it is important to identify the end of the transient state on-the-fly (online or in-line) i.e. in real-time with the ongoing high-fidelity simulation. Accurate long term prediction of self-repeating behavior requires the DMD harvesting region to include the time window showcasing such behavior. Such self-repeating nature can be broadly characterized by steady state (constant), periodic and *quasi* (or *pseudo*) periodic behavior. For simplicity, we will refer to steady-state and periodic behavior with constant amplitude as "equilibrium," and periodic behavior with growing or decaying amplitude as *quasi* equilibrium, despite a slight abuse of the terms. One may terminate high-fidelity simulations once the system has started showing such behavior, ensuring enough quality data for the DMD to work with. Several past works [60, 61, 62, 63] deal with identification of state transition in high-dimensional

physical systems. Some recent publications [64, 65] highlight the importance of DMD in identifying such regime transitions. The authors in [64] rely on the DMD reconstruction error difference between transient and equilibrium states of a dynamical system to identify such transitions. However, one of the key assumptions in [64] is the fast relaxation of the dynamical system in transience, i.e. a faster time scale of the transient dynamics compared to equilibrium dynamics. The present work does not rely on the fast relaxation assumption since the transience is characterized by temporal variations in the amplitude and changing frequency content. While [65] also performs identification of regime transition, it does so by observing the variation of a DMD-based least-squares residual term as the DMD window is gradually increased to span the spatial domain. In contrast, the residual term in this work is based on the loci of DMD eigenvalues in the complex plane and consistency between different DMD predictions. This chapter focuses on two different approaches for real-time identification of onset of self-repeating (constant, periodic or *quasi* periodic) behavior charcterized by steady state, equilibrium, *quasi* (or *pseudo*) equilibrium, or limit cycle behavior:

1. The first approach focuses on systems exhibiting equilibrium behavior. This approach involves tracking DMD modes with the sliding DMD time windows during high-fidelity simulations (Fig. 3.1). We monitor the positions of DMD eigenvalues relative to the unit circle on the complex plane to detect the equilibrium state. This analysis provides insight into the convergence of DMD

mode shapes and the shifting of DMD eigenvalues as the DMD window transitions from the transient to the equilibrium state. Such analysis can reveal how hidden features in the transient state manifest as the system approaches equilibrium.

2. The second approach extends the first by also identifying the *quasi* equilibrium state. Such scenarios can be accurately modelled and predicted by DMD, thus real-time identification of onset of *quasi* equilibrium can reduce the high-fidelity simulation time. This approach utilizes the difference between DMD predictions from consecutive DMD windows. It is more general, capable of detecting the onset of any self-repeating behavior, whether oscillation amplitudes remain constant, decay, or grow over time. However, since it does not involve DMD mode tracking, it does not provide insights into how transient DMD features evolve into equilibrium features.

3.1 Real-Time Detection of Equilibrium by DMD Mode Tracking

We slide the DMD time window in forward time as the new data becomes available from time-domain high-fidelity simulation, and track the changes in the extracted DMD features. This is particularly useful while characterizing highly nonlinear physical systems [66], as the sliding-window DMD approximates the evolution of a nonlinear system through piecewise linear dynamic systems supported by the windowed

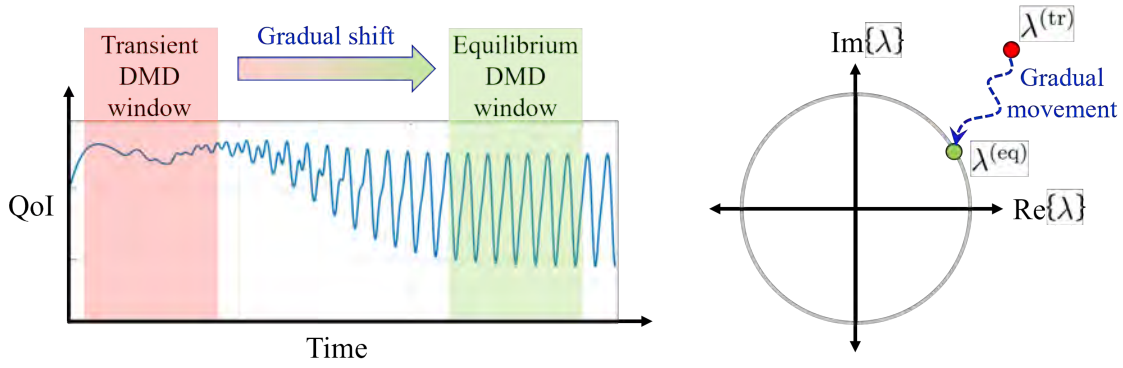


Figure 3.1: Illustrative cartoon of on-the-fly DMD approach for detecting state transition based on tracking of dominant DMD eigenvalues from transient ($\lambda^{(\mathrm{tr})}$) to equilibrium ($\lambda^{(\mathrm{eq})}$) on the complex plane. Here QoI represents the quantity of interest.

data [67, 68, 69, 70, 71, 72]. Next, we present the algorithm to track DMD modes across sliding DMD windows, followed by detection of the equilibrium phase.

3.1.1 Tracking DMD Modes

DMD captures key features of a dynamical system within the data-harvesting time window. For a sufficiently "well-behaved" dynamic system, an infinitesimal shift in the DMD window is not expected to produce a drastic change in its constituent spatio-temporal features. We aim to track each DMD eigenvalue-mode pair $(\lambda, \boldsymbol{\phi})$ from one DMD window to next, because doing so provides insights into how constitutive features of the dynamic system evolve. More importantly, it also helps identify if certain $(\lambda, \boldsymbol{\phi})$ pairs become "sufficiently" stationary over several windows,

indicating the onset of equilibrium. Mode tracking is an evolving field of study [73, 74, 75, 76]. Generally, the tracking of eigenvectors is preferred over eigenvalues due to convergence issues caused by repeated (or nearly equal) eigenvalues [73]. However, in DMD theoretical framework we work with the assumption that DMD eigenvalues are distinct [38, 39, 77].

In DMD, the effect of a sliding window can be viewed as a perturbation in the snapshot matrices. Let $\mathbf{X}_{k-1}^{(w)}, \mathbf{X}_{k}^{(w)}$ (from (2.17), (2.18)) be the snapshot matrices for the w^{th} window and $\mathbf{X}_{k'-1}^{(w+1)}, \mathbf{X}_{k'}^{(w+1)}$ for the $(w+1)^{\text{th}}$ window, with w^{th} and $(w+1)^{\text{th}}$ window usually multiple snapshots apart. Note that we use k' subscript for $(w+1)^{\text{th}}$ window since as we move to the next window, the snapshot matrix starts from a different time snapshot k', where $(k'-k)$ is the amount of shift in terms of time samples or snapshots. One can write $\mathbf{X}_{k'-1}^{(w+1)} = \mathbf{X}_{k-1}^{(w)} + \delta_1$ and $\mathbf{X}_{k'}^{(w+1)} = \mathbf{X}_{k}^{(w)} + \delta_2$. The amount of perturbation (δ_1, δ_2) depends on how fast the system changes between two consecutive DMD windows. Through the arguments presented below, we first point out that infinitesimal perturbations in the snapshot matrix will result in only infinitesimal changes in DMD modes and eigenvalues. The following arguments concerning (2.17)-(2.23) support this claim:

1. DMD elements $\mathbf{A}$, $\tilde{\mathbf{A}}$, $\mathbf{\Phi}$, as well as the reconstruction in (2.23) are linear transformations whose continuity ensures small change in output with small change in input.

2. Continuity is less obvious for (2.19) and the spectral decomposition of $\tilde{\mathbf{A}}$. However, the perturbation bounds for singular values and singular vectors are well documented [78, 79, 80, 81], ensuring infinitesimal change in output given infinitesimal change in input for (2.19). Regarding the eigendecomposition step, continuity of the roots of a polynomial ensures that eigenvalues of $\tilde{\mathbf{A}}$ (roots of its characteristic polynomial) do not experience discontinuities under small perturbations. Similarly, perturbation bounds for eigenvectors of simple eigenvalues [82] assures an infinitesimal change in $\mathbf{W}$, thus an infinitesimal change in DMD modes with infinitesimal change in $\tilde{\mathbf{A}}$.

Following the above arguments, a gradual shift in the DMD window is expected to lead to a gradual change in DMD eigenvalues and mode shapes. An exception arises at bifurcation points, which we address in the tracking algorithm described below.

The tracking algorithm refers to each DMD mode and corresponding eigenvalue as the pair $(\lambda, \boldsymbol{\phi})$. In other words, both the position of λ in the complex plane as well as information on the spatial distribution of $\boldsymbol{\phi}$ are employed for mode tracking. Define $(\lambda_i^{(w)}, \boldsymbol{\phi}_i^{(w)})$ as the DMD eigenvalue-mode pair in the w^{th} window, where $1 \leq i \leq p$. The aim of the tracking algorithm is to assign $(\lambda_j^{(w+1)}, \boldsymbol{\phi}_j^{(w+1)})$, $(1 \leq j \leq q)$ from $(w+1)^{th}$ window to $(\lambda_i^{(w)}, \boldsymbol{\phi}_i^{(w)})$ as its successor. Assuming p and q to be the number of DMD modes in the w^{th} and $(w+1)^{th}$ window respectively, there can be broadly three scenarios,

1. $p = q$: In this case, each DMD eigenvalue-mode pair in the w^{th} window is associated with exactly one pair in the $(w+1)^{th}$ window.

2. $p > q$: The algorithm must terminate the tracking of some pairs $(\lambda_i^{(w)}, \boldsymbol{\phi}_i^{(w)})$ to which no successors can be assigned.

3. $p < q$: The algorithm must initiate the tracking of newly identified pairs starting at the $(w+1)^{th}$ window after all pairs from the w^{th} window have been assigned unique successors.

The primary condition for successor assignment is given in terms of the placement of DMD eigenvalues. In other words, the first candidate for mode-matching of $(\lambda_i^{(w)}, \boldsymbol{\phi}_i^{(w)})$ is

$$\tilde{j} = \underset{j=1,\ldots q}{\arg\min} \|\lambda_i^{(w)} - \lambda_j^{(w+1)}\| \tag{3.1}$$

If a conflict arises, resulting in assignment of the same $\tilde{j}$ for multiple i, mode-shape matching is invoked as the secondary criterion for tracking. The modal assurance criterion (MAC) is a popular metric used for comparing mode shapes [74], given by,

$$\mathrm{MAC}(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \frac{\left|\boldsymbol{\phi}_i^{\mathrm{T}}\ \overline{\boldsymbol{\phi}}_j\right|^2}{(\boldsymbol{\phi}_i^{\mathrm{T}}\ \overline{\boldsymbol{\phi}}_i) \cdot (\boldsymbol{\phi}_j^{\mathrm{T}}\ \overline{\boldsymbol{\phi}}_j)}. \tag{3.2}$$

This work uses the absolute value of MAC, defined as $\rho(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = |\mathrm{MAC}(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)|$. The maximum value ρ can attain is 1, denoting an exact configuration match, while $\rho = 0$ indicates no match at all. The tracking algorithm is described in Algorithm 1.

At the bifurcation point, broadly two scenarios are possible. First, one complex conjugate pair of DMD eigenvalues generates two real eigenvalues after encountering

Algorithm 3.1: Algorithm for tracking DMD eigenvalue-mode pair $(\lambda, \boldsymbol{\phi})$.

Input: DMD eigenvalue-mode pair $(\lambda_i^{(w)}, \boldsymbol{\phi}_i^{(w)})$ from w^{th} window, $i = 1, 2, \ldots, p$ and $(\lambda_j^{(w+1)}, \boldsymbol{\phi}_j^{(w+1)})$ from $(w+1)^{th}$ window, $j = 1, 2, \ldots, q$.

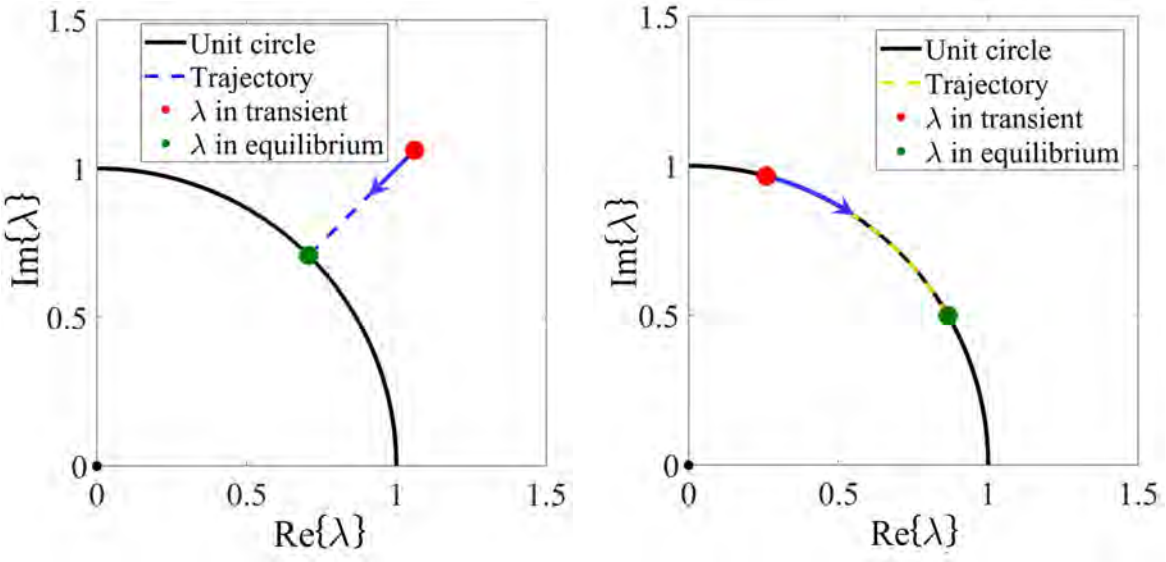
Output: Successor of $(\lambda_i^{(w)}, \boldsymbol{\phi}_i^{(w)})$.

1: **for** i = 1 to p **do**
2: Find $\tilde{j} = \arg\min_{j=1,\ldots q} d_\lambda(i,j) = \arg\min_{j=1,\ldots q} \|\lambda_i^{(w)} - \lambda_j^{(w+1)}\|$.
3: **end for**
4: **if** All i are associated with distinct $\tilde{j}$ **then**
5: **return** $(\lambda_{\tilde{j}}^{(w+1)}, \boldsymbol{\phi}_{\tilde{j}}^{(w+1)})$ as successor of respective $(\lambda_i^{(w)}, \boldsymbol{\phi}_i^{(w)})$.
6: **else**
7: Identify the set of indices i which share a common $\tilde{j}$. Let I be the set of i $(i \in I)$ which have common $\tilde{j} = \tilde{j}_I$.
8: Identify $\hat{i} = \arg\max_{i \in I} \rho(\boldsymbol{\phi}_i^{(w)}, \boldsymbol{\phi}_{\tilde{j}_I}^{(w+1)})$
9: Identify the second closest eigenvalue to $\lambda_{\hat{i}}^{(w)}$ from $(w+1)^{th}$ window after $\lambda_{\tilde{j}_I}^{(w+1)}$. Let the index of second closest eigenvalue be $\tilde{j}_{2I}$.
10: **if** $\big(\rho(\boldsymbol{\phi}_{\hat{i}}^{(w)}, \boldsymbol{\phi}_{\tilde{j}_I}^{(w+1)}) \geq \rho(\boldsymbol{\phi}_{\hat{i}}^{(w)}, \boldsymbol{\phi}_{\tilde{j}_{2I}}^{(w+1)})\big)$ **then**
11: **return** $(\lambda_{\tilde{j}_I}^{(w+1)}, \boldsymbol{\phi}_{\tilde{j}_I}^{(w+1)})$ as successor of $(\lambda_{\hat{i}}^{(w)}, \boldsymbol{\phi}_{\hat{i}}^{(w)})$.
12: For, $\forall i \in I - \{\hat{i}\}$, replace the closest eigenvalue index $\tilde{j}_I$ by next closest eigenvalue index $\tilde{j}_{2I}$ from $(w+1)^{th}$ window. If there is no next closest eigenvalue, the eigenvalue-mode pair corresponding to $i \in I - \{\hat{i}\}$ is not tracked further. Repeat from step 4 for rest of the eigenvalues.
13: **else**
14: Delete $\hat{i}$ from I, so that $\hat{i} \notin I$. Then repeat from step 8.
15: **end if**
16: **if** $\big(\rho(\boldsymbol{\phi}_i^{(w)}, \boldsymbol{\phi}_{\tilde{j}_I}^{(w+1)}) < \rho(\boldsymbol{\phi}_i^{(w)}, \boldsymbol{\phi}_{\tilde{j}_{2I}}^{(w+1)})\big)$, $\forall i \in I$ **then**
17: Identify the $\tilde{i} = \arg\max_{i \in I} |d_\lambda(i, \tilde{j}_I) - d_\lambda(i, \tilde{j}_{2I})|$.
18: **return** $(\lambda_{\tilde{j}_I}^{(w+1)}, \boldsymbol{\phi}_{\tilde{j}_I}^{(w+1)})$ as successor of $(\lambda_{\tilde{i}}^{(w)}, \boldsymbol{\phi}_{\tilde{i}}^{(w)})$.
19: For, $\forall i \in I - \{\tilde{i}\}$, replace the closest eigenvalue index $\tilde{j}_I$ by next closest eigenvalue index $\tilde{j}_{2I}$ from $(w+1)^{th}$ window. If there is no next closest eigenvalue, the eigenvalue-mode pair corresponding to $i \in I - \{\tilde{i}\}$ is not tracked further. Repeat from step 4 for rest of the eigenvalues.
20: **end if**
21: **end if**

the real axis. Second, two real DMD eigenvalues merge and become a complex conjugate pair of eigenvalues. Since for real data the DMD eigenvalues are mirrored with respect to the real axis, the tracking algorithm concentrates only on the upper half complex plane including the real axis. The first scenario leads to $p < q$, where the algorithm starts tracking the newly generated DMD eigenvalues from that particular window. For the second case $p > q$, the algorithm stops tracking some eigenvalues from the previous window.

3.1.2 State Transition: From Transient to Equilibrium

DMD analysis of a dynamical system focuses on low-dimensional modeling of the equilibrium state, usually ignoring transient phenomena [83, 84, 85]. Regardless of the ROM employed, knowing when the transient phase comes to an end is useful for terminating the high-fidelity simulation in a timely fashion so that the future solution can be predicted with predictive models like Koopman ROMs. In the literature, several methods are available for detecting state transition of high-dimensional dynamical systems [60, 61, 62, 63]. The authors in [64] have presented a method exploiting DMD reconstruction error for identifying such transitions. The current paper takes advantage of the temporal variation in position of the DMD eigenvalues in the complex plane with respect to the unit circle. The algorithm presented here can be exploited for on-the-fly applications, given some a priori knowledge about the timescale of the problem. This will be discussed in detail later. A preliminary version of this state transition algorithm was described in our work [86].



(a) Radial movement.

(b) Azimuthal movement.

Figure 3.2: Eigenvalue movements: (a)Radial movement of eigenvalue towards unit circle as the DMD window moves from transient to equilibrium state; (b) Eigenvalue movement along unit circle to the equilibrium position (green) as the DMD window moves towards equilibrium from transient.

The fundamental idea behind the proposed approach is that given a sufficiently wide data harvesting region within the equilibrium state, it follows that (a) the dominant DMD eigenvalues lie on the unit circle [38, 39, 87] and (b) the mode shapes and corresponding frequencies associated with the dominant DMD modes remain invariant. Intuitively, the latter makes sense because in equilibrium the dynamics of the system remain unchanged irrespective of the observation window as long as that window is sufficiently wide. The Maxwell-Vlasov equations (governing equations in

kinetic plasma simulations) are autonomous in nature. In the equilibrium state, the number of particles entering the solution domain remains same as the number of particles leaving, ensuring that the governing dynamics are autonomous. Note that the solution of the well-posed DMD is unique [77], whereby the extracted dominant DMD modes and corresponding eigenvalues remain unchanged as we slide the window within the equilibrium region. Of course, conditions (a) and (b) are not necessarily true in the transient state as indicated by the presence of dominant DMD eigenvalues away from the unit circle and continuously changing dynamics.

In practical scenarios, the data obtained is not free from noise (in a general sense, either from finite machine precision and discretization errors present in a simulation or from ambient and instrument noise present in a measurement). As a result, conditions (a) and (b) are not exactly satisfied. Therefore, we emphasize that invariance of characteristics applies to only dominant DMD modes (i.e. with physically meaningful character) in the equilibrium stage. DMD modes corresponding to the noise space of the data do not follow such observations. Here, we adopt a 5% error criterion, so the first few high energy DMD modes corresponding to $\geq$ 95% of the reconstructed amplitude [1] are defined as the dominant modes. As the harvesting window approaches the equilibrium state, two key parameters are tracked. The α parameter measures the relative error in the reconstructed data, assuming exponential growth or decay to be the only source of error due to non-zero distance of the

[1]Note that the modal amplitude, defined as $A_m(t) = \|\vartheta_m \boldsymbol{\phi}_m e^{\widehat{\omega}_m(t-t_0)} + \overline{\vartheta}_m \overline{\boldsymbol{\phi}}_m e^{\overline{\widehat{\omega}}_m(t-t_0)}\|_2$ ($\|.\|_2$ denotes the Frobenius norm) varies with time, so the measurements are performed at the end of the DMD harvesting window.

dominant DMD eigenvalues from the unit circle. The β parameter represents the error in reconstructed data considering the error only due to fluctuation in phase of dominant DMD eigenvalues. The expressions for α and β are derived next. Recall the DMD reconstruction formula,

$$\widehat{\mathbf{x}}(t) \approx \sum_{m=1}^{M_d} (\vartheta_m \boldsymbol{\phi}_m e^{\widehat{\omega}_m (t-t_0)} + \overline{\vartheta}_m \overline{\boldsymbol{\phi}}_m e^{\overline{\widehat{\omega}}_m (t-t_0)}) \tag{3.3a}$$

$$= \sum_{m=1}^{M_d} e^{\widehat{\omega}_{mR}(t-t_0)} (\vartheta_m \boldsymbol{\phi}_m e^{j\widehat{\omega}_{mI}(t-t_0)} + \overline{\vartheta}_m \overline{\boldsymbol{\phi}}_m e^{-j\widehat{\omega}_{mI}(t-t_0)})$$

$$= \sum_{m=1}^{M_d} e^{\widehat{\omega}_{mR}(t-t_0)} \psi_m, \tag{3.3b}$$

where complex frequency $\widehat{\omega}_m = \widehat{\omega}_{mR} + j\widehat{\omega}_{mI}$. $\psi_m = (\vartheta_m \boldsymbol{\phi}_m e^{j\widehat{\omega}_{mI}(t-t_0)} + \overline{\vartheta}_m \overline{\boldsymbol{\phi}}_m e^{-j\widehat{\omega}_{mI}(t-t_0)})$ is the oscillating part of the solution and M_d is the number of dominant DMD modes. In equilibrium, the DMD solution must not include exponentially growing or decaying factors, i.e. $\widehat{\omega}_{mR} = 0$, $m = 1, 2, \ldots M_d$. Assuming non-zero $\widehat{\omega}_{mR}$ to be the only source of error, the ideal solution $\mathbf{x}(t)$ must be

$$\mathbf{x}(t) \approx \sum_{m=1}^{M_d} \psi_m. \tag{3.4}$$

Using $\widehat{\omega}_m = \ln(\lambda_m)/\Delta_t$, $\lambda_m = |\lambda_m| e^{j\theta_m}$, one can write $\widehat{\omega}_m = \frac{\ln|\lambda_m|}{\Delta_t} + \frac{j\theta_m}{\Delta_t} = \widehat{\omega}_{mR} + j\widehat{\omega}_{mI}$, giving $\widehat{\omega}_{mR} = \frac{\ln|\lambda_m|}{\Delta_t}$. From (3.3b),

$$\hat{\mathbf{x}}(t) \approx \sum_{m=1}^{M_d} |\lambda_m|^{\frac{(t-t_0)}{\Delta_t}} \psi_m. \tag{3.5}$$

The relative 2-norm error, $\delta(t)$, is given below, under the assumption that error is only due to exponential growth/decay. This results in the definition of the parameter

α:

$$\delta(t) = \frac{||\hat{\mathbf{x}}(t) - \mathbf{x}(t)||_2}{||\mathbf{x}(t)||_2} = \frac{||\sum_{m=1}^{M_d} |\lambda_m|^{\frac{(t-t_0)}{\Delta_t}} \psi_m - \sum_{m=1}^{M_d} \psi_m||_2}{||\sum_{m=1}^{M_d} \psi_m||_2} \tag{3.6}$$

$$= \frac{||\sum_{m=1}^{M_d} (|\lambda_m|^{\frac{(t-t_0)}{\Delta_t}} - 1)\psi_m||_2}{||\sum_{m=1}^{M_d} \psi_m||_2} \tag{3.7}$$

and hence, from the triangle inequality, it follows

$$\delta(t) \leq \frac{\sum_{m=1}^{M_d} ||(|\lambda_m|^{\frac{(t-t_0)}{\Delta_t}} - 1)\psi_m||_2}{||\sum_{m=1}^{M_d} \psi_m||_2} = \alpha(t - t_0) = \alpha(\tilde{t}). \tag{3.8}$$

Note that ψ_m is a function of the time difference between target time t and the reference initial time of that particular DMD window t_0, which we denote as $\tilde{t}$. As a result, we can write $\alpha(t - t_0)$=$\alpha(\tilde{t})$ in the above. As the DMD window moves towards equilibrium, the dominant DMD eigenvalues move closer towards unit circle (Fig. 3.2), thus decreasing $\alpha(\tilde{t})$, for a fixed $\tilde{t}$. We define the convergence in α as the termination of its secular decay. However, a dynamical system can continue to be in transience even after all DMD eigenvalues have moved to the unit circle (Fig. 3.2b). This happens when the transient state involves variation of frequency content instead of amplitude. Thus the presented approach also validates that the dominant DMD eigenvalues do not move along the unit circle. Doing so ensures that the error due to shift in phase ($\Delta\theta_m$) over successive windows is less than some predetermined

threshold β_{thr}.

$$\begin{aligned}\hat{\mathbf{x}}(t) &\approx \sum_{m=1}^{M_d}(\vartheta_m \boldsymbol{\phi}_m e^{\tilde{\omega}_m(t-t_0)} + \overline{\vartheta}_m \overline{\boldsymbol{\phi}}_m e^{\overline{\tilde{\omega}}_m(t-t_0)}) \\ &= \sum_{m=1}^{M_d}(\vartheta_m \boldsymbol{\phi}_m e^{(\widehat{\omega}_{mR}+j(\widehat{\omega}_{mI}+\Delta\widehat{\omega}_{mI}))(t-t_0)} + \overline{\vartheta}_m \overline{\boldsymbol{\phi}}_m e^{(\widehat{\omega}_{mR}-j(\widehat{\omega}_{mI}+\Delta\widehat{\omega}_{mI}))(t-t_0)}) \\ &= \sum_{m=1}^{M_d}(\chi_m e^{j\Delta\widehat{\omega}_{mI}(t-t_0)} + \overline{\chi}_m e^{-j\Delta\widehat{\omega}_{mI}(t-t_0)})\end{aligned} \tag{3.9}$$

$$\begin{aligned}&= \sum_{m=1}^{M_d}(\chi_m e^{j\frac{\Delta\theta_m}{\Delta_t}(t-t_0)} + \overline{\chi}_m e^{-j\frac{\Delta\theta_m}{\Delta_t}(t-t_0)}) \\ &= \sum_{m=1}^{M_d} 2\mathrm{Re}\{\chi_m e^{j\frac{\Delta\theta_m}{\Delta_t}(t-t_0)}\},\end{aligned} \tag{3.10}$$

where, $\tilde{\omega}_m = \widehat{\omega}_m + j\Delta\widehat{\omega}_{mI}$ and $\chi_m = \vartheta_m \boldsymbol{\phi}_m e^{\widehat{\omega}_m(t-t_0)}$, a function of $(t-t_0) = \tilde{t}$. As above, we compute the relative 2-norm error $\delta(t)$ under the assumption that error is only due to $\Delta\widehat{\omega}_{mI}$ with $\mathbf{x}(t) \approx \sum_{m=1}^{M_d} 2\mathrm{Re}\{\chi_m\}$ as the ideal solution. This results in the definition of the parameter β:

$$\begin{aligned}\delta(t) = \frac{||\hat{\mathbf{x}}(t) - \mathbf{x}(t)||_2}{||\mathbf{x}(t)||_2} &= \frac{||\sum_{m=1}^{M_d} 2\mathrm{Re}\{\chi_m e^{j\frac{\Delta\theta_m}{\Delta_t}(t-t_0)}\} - \sum_{m=1}^{M_d} 2\mathrm{Re}\{\chi_m\}||_2}{||\sum_{m=1}^{M_d} 2\mathrm{Re}\{\chi_m\}||_2} \\ &= \frac{||\sum_{m=1}^{M_d} 2\mathrm{Re}\{\chi_m(e^{j\frac{\Delta\theta_m}{\Delta_t}(t-t_0)} - 1)\}||_2}{||\sum_{m=1}^{M_d} 2\mathrm{Re}\{\chi_m\}||_2}\end{aligned} \tag{3.11}$$

and hence

$$\delta(t) \leq \frac{\sum_{m=1}^{M_d} ||\mathrm{Re}\{\chi_m(e^{j\frac{\Delta\theta_m}{\Delta_t}(t-t_0)} - 1)\}||_2}{||\sum_{m=1}^{M_d} \mathrm{Re}\{\chi_m\}||_2} = \beta(t-t_0) = \beta(\tilde{t}) \tag{3.12}$$

Parameters α and β are computed for dominant DMD modes only. Our goal is to detect the "knee" or "elbow" region in the graph of α against w (window index), capturing transition to equilibrium state. However, on-the-fly detection of the knee region is challenging, especially when data is noisy. We thus examine the rolling average of α over W successive windows and search for a non-negative slope in the averaged graph, hinting convergence in α. Once convergence in α is detected, the focus shifts to the parameter β to ensure that the error due to phase shift of the dominant eigenvalues over W windows is within an acceptable bound ($\leq \beta_{thr}$). We will illustrate the on-the-fly algorithm (Algorithm 2) assuming we have some a priori knowledge about the timescale of the problem to inform the selection of appropriate window width Δt_w (in time) or equivalently Δn_w (in time steps).

It is important to note that the performance of Algorithm 2 depends on the choice of parameters Δt_w, W, β_{thr} and δn_w, where δn_w is the shift between two successive sliding DMD windows in terms of time steps. In terms of continuous time, this shift is represented by $\delta t_w = \delta n_w \Delta t$. These parameters must be selected beforehand and do not adapt during the run. We make the following observations:

- The parameter Δt_w denotes the width of each sliding window. Prior knowledge about the time-scale of the problem helps to make sure that Δt_w covers multiple oscillation cycles (if any) in the equilibrium state. If the window width is not sufficient to capture the dynamics of the equilibrium, temporal variation in the parameter α might not elicit convergence even as the window slides towards

Algorithm 3.2: Algorithm for detecting onset of equilibrium state

Input: Data from high-fidelity simulation.

Output: Window index indicating onset of equilibrium.

Initialization : For first window, calculate r as using optimal hard thresholding and use it for rest of the algorithm.

1: At current (w^{th}) window, say D denotes the set of dominant DMD eigenvalues. Identify the i for which $\lambda_i^{(w)} \in D$, where $1 \leq i \leq p$, and p is number of DMD modes (M from (2.24)) in the w^{th} window.

2: Calculate $\alpha(\tilde{t})$ for w^{th} window at $\tilde{t} = \Delta t_w$ denoted by $\alpha(\Delta t_w)^{(w)}$, where Δt_w is the DMD window width.

3: For $w \geq Wh$, perform averaging of α over W windows to get $< \alpha >_W^{(h)} = [\alpha(\Delta t_w)^{(s+1)} + \alpha(\Delta t_w)^{(s+2)} + \ldots + \alpha(\Delta t_w)^{(s+W)}]/W$, where $s = W(h-1)$, $h = 1, 2, \ldots$.

4: **if** $(\log(< \alpha >_W^{(h)}) \geq \log(< \alpha >_W^{(h-1)}))$ **then**

5: From the tracking Algorithm 1, identify the predecessors of $\lambda_i^{(w)} \in D$ for previous W windows, $\lambda_{(i)}^{(w-1)}, \lambda_{(i)}^{(w-2)}, \ldots, \lambda_{(i)}^{(w-W)}$, calculate $\Delta\theta_{a,i}^{(w)} = |\text{Arg}(\lambda_{(i)}^{(w-a)}) - \text{Arg}(\lambda_i^{(w)})|$, where $a = 1, 2, \ldots, W$.

6: Calculate $\beta(\Delta t_w)$ wrt. $\Delta\theta_{a,i}^{(w)}$ at w^{th} window for W predecessors, $\beta(\Delta t_w)_1^{(w)}, \beta(\Delta t_w)_2^{(w)}, \ldots, \beta(\Delta t_w)_W^{(w)}$.

7: **if** $(\beta(\Delta t_w)_1^{(w)}, \beta(\Delta t_w)_2^{(w)}, \ldots, \beta(\Delta t_w)_W^{(w)} \leq \beta_{thr})$ **then**

8: Stop harvesting.

9: **return** w

10: **else**

11: Continue harvesting, move to the $(w+1)^{th}$ window and return to the step 1.

12: **end if**

13: **else**

14: Continue harvesting, move to the $(w+1)^{th}$ window and return to the step 1.

15: **end if**

equilibrium. For offline applications, one of many simple algorithms such as zero crossing detection or peak detection can be used to approximate the period of limit cycle oscillations.

- The shift δn_w generally spans an integer multiple of snapshots. For on-the-fly processes, the natural choice is to shift by one snapshot, in which case DMD is performed when a new snapshot becomes available. In our test cases, we shift by two snapshots as it provides enough flexibility to experiment with varying snapshot intervals, keeping the shift $\delta n_w (\equiv \delta t_w)$ constant. Ideally, overlap between successive windows must be avoided to minimize the computation cost. In practice however, intersection between consecutive sliding windows is employed for the following reasons: (i.) window overlap implies smaller perturbation in the snapshot matrix, which helps with tracking DMD eigenvalue-mode pair, and, (ii.) overlap helps determine W, the number of windows over which α is averaged.

- W is chosen such that it is the minimum number of shifts for there to be no overlap between w^{th} and $(w+W)^{th}$ window. In other words, $W = \lfloor \Delta t_w / \delta t_w \rfloor$. Too small a value for W can result in premature, erroneous detection of equilibrium, especially for highly noisy α variation. While a large W overcomes this difficulty, it is at the cost of delayed detection of equilibrium. Delayed detection of equilibrium does not pose risk of increased error in DMD extrapolation, only inefficiency.

- The threshold β_{thr} is based on the acceptable error limit for each application. In this work we follow a 5%. From experience, the error due to fluctuations in phase is much smaller than errors due to exponential growth/decay. Therefore, we set $\beta_{thr} = 0.01$ (1% error). As shown in (3.12), β is calculated based on the difference in phase of dominant eigenvalue at the w^{th} window and it's predecessors at previous windows. It aims to detect slow unidirectional movement (Fig. 3.2b) of eigenvalues along the unit circle. As a rule of thumb, we check for the error due to shift in phase over W previous windows. If there is extremely slow movement of DMD eigenvalues along the unit circle in the transient state, it might go undetected for small W value. Of course, very slow phase variation might not be of interest, as long as the reconstruction accuracy stays within acceptable limits.

3.1.3 Toy Example: Lorenz'96 Model

The proposed on-the-fly equilibrium detection algorithm is discussed in chapter 5 in the context of EMPIC plasma simulations. In this chapter, we will showcase its effectiveness for a toy example, the Lorenz'96 model. The Lorenz'96 model was introduced by Edward Lorenz in 1996 [88] as a simplified model for predicting atmospheric phenomena. It has been widely used with data assimilation and ensemble forecasting techniques [89, 90, 91, 92]. Here, we use the Lorenz'96 model to demonstrate the effectiveness of Algorithm 2 in detecting the onset of the equilibrium state in a perhaps more familiar setting. For N states, the equations governing the dynamics

are

$$\dot{y_i} = (y_{i+1} - y_{i-2})y_{i-1} - y_i + F_e \tag{3.13}$$

$$y_{-1} = y_{N-1} \tag{3.14}$$

$$y_0 = y_N \tag{3.15}$$

$$y_{N+1} = y_1 \tag{3.16}$$

where, y_i $(i = 1, 2, \ldots, N)$ represents the state of the dynamical system and F_e the external force. We set $N = 4$ and $F_e = 10$, leading to stable limit cycle behavior (Fig. 3.4f). 200 random realizations of the initial state are achieved in the following form: $(y_1^{(0)}, y_2^{(0)}, y_3^{(0)}, y_4^{(0)}) = (F_e + \hat{\delta}_j, F_e, F_e, F_e),\ j = 1, 2, \ldots, 200,$ where $\hat{\delta}_j$ is randomly generated with uniform distribution in $[-2, 2]$. The system is solved until $t = 120$ units, with a total of $n = 6001$ time steps. Each snapshot is a 800×1 vector formed by stacking the instantaneous values of each of the 200 realizations of the four states.

Algorithm 2 is used to identify the onset of equilibrium with $\beta_{thr} = 0.01$ and $\delta t_w = 0.16$. Using $\Delta t_w = 4.8$ ns allows it to cover multiple cycles of the limit cycle oscillations with interval between two consecutive snapshots being $\Delta_t = 0.08$. It is seen that $\alpha(\Delta t_w)$ decreases with increasing k initially but eventually converges (Fig. 3.3), with the knee/elbow region marking the transition from transient to steady-state. In this figure, the rolling average of $\alpha(\Delta t_w)$ over $\lfloor \Delta t_w / \delta t_w \rfloor$ data points is shown (in this case, 30 datapoints). The knee/elbow region for the averaged graph is clearly visible around $w = 105$, indicating state transition around $n_{st}(105) = 833$

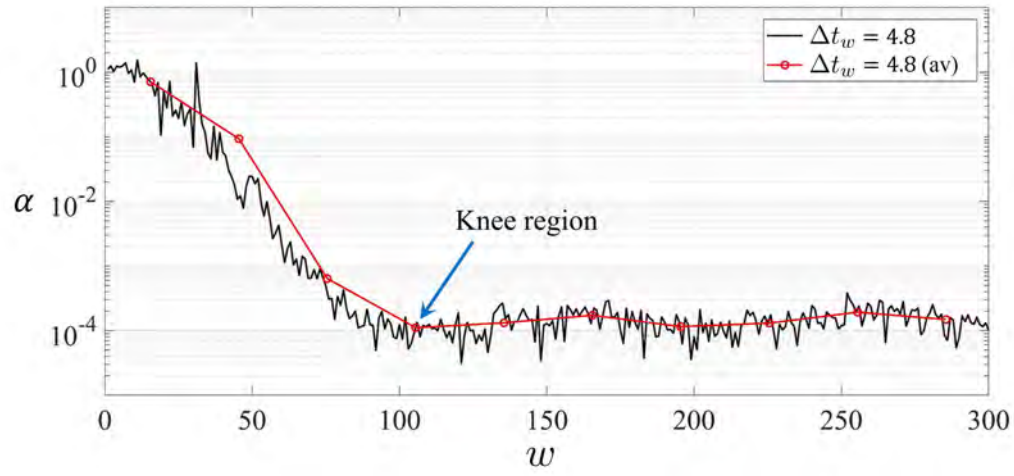
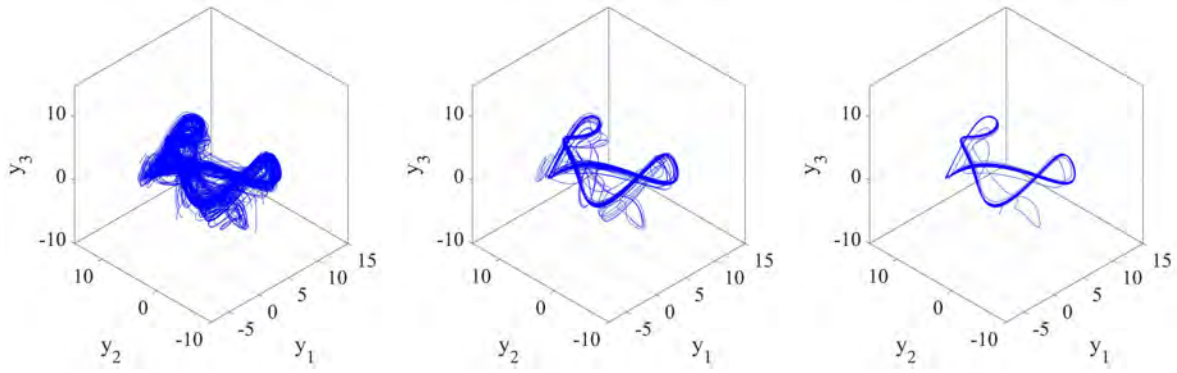


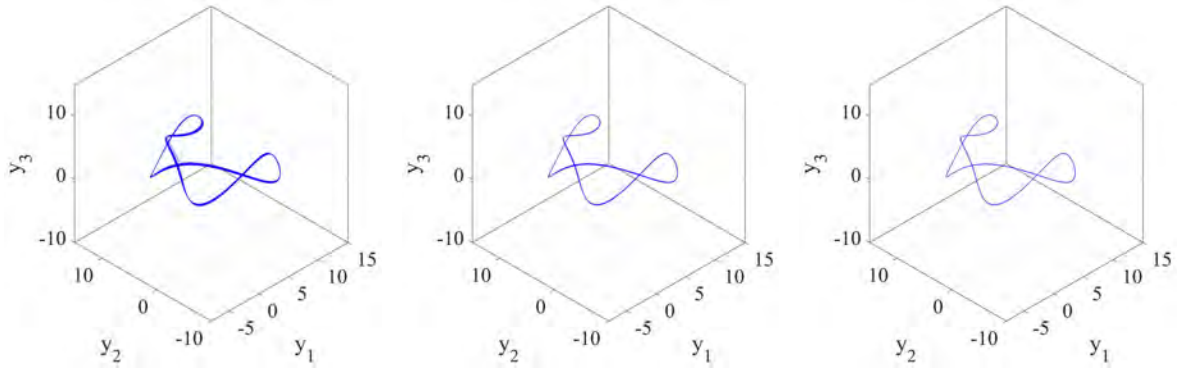
Figure 3.3: Variation in $\alpha(\Delta t_w)$ as the window slides towards equilibrium for $\Delta_t = 0.08$. The knee region of the averaged (red) graph is at around $w = 105$.

to $n_{en}(105) = 1073$, where $n_{st}(k)$ and $n_{en}(k)$ are respectively the starting and ending time step of k^{th} window.

This result is validated by plotting the state-space trajectories, see Fig. 3.4. For $n_{beg} = 900$, we observe formation of a periodic orbit (Fig. 3.4e) indicating the onset of equilibrium. This is in agreement with Algorithm 2, which suggests state transition in the range $n = 833$ to $n = 1073$. The sensitivity of $\alpha(\Delta t_w)$ towards variation in Δt_w and Δt is shown in Fig. 3.5 (Δ_k equivalent to 8 time steps for each case). The early detection of equilibrium region for large window widths (Fig. 3.5a) can be attributed to the "look-ahead" artifact due to the finite width of DMD time window. As in other examples in this work, on-the-fly detection of the knee is adversely impacted by the use of the non-negative slope criterion. As seen in



(a) $n_{beg} = 100$ (b) $n_{beg} = 300$ (c) $n_{beg} = 500$



(d) $n_{beg} = 700$ (e) $n_{beg} = 900$ (f) $n_{beg} = 1100$

Figure 3.4: Trajectory of state-space (y_1, y_2, y_3) for 200 different initial conditions, from $n = n_{beg}$ to $n = 6001$. The noisy trajectories for early values of n_{beg} suggest that the states are not yet evolving on a periodic orbit, thus still in transience. From $n_{beg} = 900$ we see a clear periodic orbit, indicating the onset of equilibrium.

Fig. 3.5b, the knee region appears first, visually speaking, around $w = 105$, but the first non-negative slope is encountered much later, around $w = 135$ for $\Delta t = 0.04$. The algorithm detects it at $w = 180$ and terminates. The delayed detection is not necessarily a drawback in the sense that it provides a conservative estimate. When

the graph of $\alpha(\Delta t_w)$ is "noisy", there is potential for false positives in an on-the-fly approach. The non-negative slope criterion provides robustness against such false detection, but at the cost of precision since detection of equilibrium is delayed. As mentioned before, delayed equilibrium detection does not affect prediction accuracy, but it does affect its computational efficiency. More work is needed for building better methods for on-the-fly detection of the knee region in noisy datasets. Discussion of additional textbook example can be found in [86].

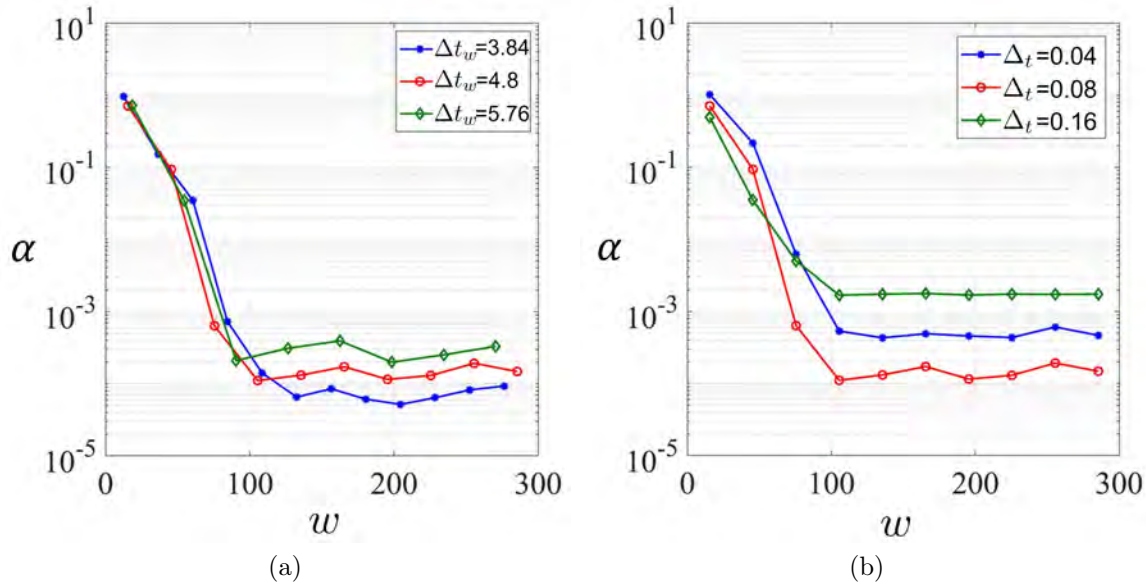


Figure 3.5: (a) Sensitivity of Algorithm 2 towards window width Δt_w ($\pm 20\%$), keeping fixed $\Delta_t = 0.08$. Equilibrium detected at $w = 168, 150$ and 144 for $\Delta t_w = 3.84, 4.8$ and 5.76 respectively. (b) Sensitivity of Algorithm 2 towards sampling interval Δ_t , keeping fixed $\Delta t_w = 4.8$. Equilibrium detected at $w = 180, 150$ and 150 for $\Delta_t = 0.04, 0.08$ and 0.16 respectively.

3.2 Real-Time Detection of *Quasi* Equilibrium using Consistency among DMD Predictions

A sliding-window DMD algorithm for online identification of equilibrium was discussed in Section 3.1 [47, 86]. Recently a similar approach of observing convergence in DMD eigenvalues and modal amplitudes was proposed in [93] as well. However, our proposed method is much simpler and focuses only on convergence of the DMD solution at some future time-window. It is also important to mention that the method in Section 3.1 is tailored towards online detection of only the equilibrium state. However, since we are interested in detecting the pseudo-equilibrium as well, we propose a sliding-window DMD algorithm based on convergence of the DMD solutions inside a fixed future time window. Below we provide our rationale for using the convergence in DMD solutions as our criterion,

- The underlying features of a dynamical system (e.g. frequency content) remain invariant in equilibrium or *quasi* equilibrium, indicating invariance in the extracted DMD modes and frequencies, given that the observation (DMD) window is large enough.

- A well-posed DMD has unique solution i.e. a unique set of DMD modes, frequencies and modal amplitudes [94], ensuring a unique DMD reconstruction for some future time-window. The Hankel stacking essentially helps make an ill-posed DMD problem well-posed [95, 56].

Algorithm 3.3: On-the-fly sliding-window DMD algorithm for detecting t_f for real-time termination of FDTD/FETD.

Input:
- The approximate time instant t_{qry} up to which we wish to extrapolate for any specific application.
- Approximate idea regarding the maximum dimension (L) and material properties of the cavity.
- High-fidelity data from the ongoing FDTD/FETD simulation.

Output: FDTD/FETD termination flag.

1: Determine the DMD window width Δt_w or equivalently Δn_w, where $\Delta n_w = \lceil \Delta t_w / \Delta t \rceil$ following Section 4.3.1.
2: After the FDTD simulation reaches $n = \Delta n_w$, determine the DMD sampling interval Δ_t and appropriate number of stacks d as per Section 4.3.1.
3: As the FDTD/FETD simulation progresses, shift the DMD window by δn_w time steps (represented by increasing window index w). Let us denote the starting and endpoint of the w^{th} window by $t_{st,w} (\equiv n_{st,w})$ and $t_{en,w} (\equiv n_{en,w})$ respectively.
4: **for** Current (w^{th}) DMD window **do**
5: **if** $t_{en,w} > t_{qry}$ **then**
6: Failed to detect FDTD/FETD before desired the query time.
7: **return**
8: **else**
9: Perform Hankel-DMD for w^{th} window and obtain the DMD prediction from $t_{qry} (\equiv n_{qry})$ to $t_{qry} + \Delta t_w (\equiv n_{qry} + \Delta n_w)$ at $(n_w + 1)$ time steps. Let us denote this DMD prediction corresponding to w^{th} DMD window as $\widehat{\mathbf{x}}_w^{(n)}$.
10: Get the average relative 2-norm error between $\widehat{\mathbf{x}}_w^{(n)}$ and $\widehat{\mathbf{x}}_{w-1}^{(n)}$ over $(n_w + 1)$ time steps, denoted by $\langle \delta^{(n)} \rangle_w$, where

$$\langle \delta^{(n)} \rangle_w = \frac{1}{(n_w + 1)} \sum_{n=n_{qry}}^{n_{qry}+\Delta n_w} \frac{||\widehat{\mathbf{x}}_w^{(n)} - \widehat{\mathbf{x}}_{w-1}^{(n)}||_2}{||\widehat{\mathbf{x}}_{w-1}^{(n)}||_2} \tag{3.17}$$

11: **if** $(\langle \delta^{(n)} \rangle_w, \langle \delta^{(n)} \rangle_{w-1}, \ldots, \langle \delta^{(n)} \rangle_{w-q} < \delta_0)$ **then**
12: Raise the FDTD/FETD termination flag.
13: **return**
14: **else**
15: Move to the next DMD window, $w = w + 1$.
16: **end if**
17: **end if**
18: **end for**

Following this reasoning, we can state that as we slide the DMD window from transient to *quasi* equilibrium, the extracted DMD modes, frequencies, and modal amplitudes $\{\boldsymbol{\phi}_{\hat{m}}, \widehat{\omega}_{\hat{m}}, \vartheta_{\hat{m}}\}$ keep varying within the transient phase. Consequently, the DMD prediction over some fixed future time window also keeps varying. However, as we enter pseudo-equilibrium, the extracted $\{\boldsymbol{\phi}_{\hat{m}}, \widehat{\omega}_{\hat{m}}, \vartheta_{\hat{m}}\}$ become invariant to the DMD window shift, resulting in convergence of the DMD prediction. Upon satisfying a predetermined convergence criterion (δ_0), a flag is raised to terminate the FDTD/FETD simulation. The algorithm is described in 3.

The success of algorithm 3 depends on the careful selection of the parameters Δt_w, Δ_t, d, δn_w, δ_0, and q. Selection of Δt_w, Δ_t, d is problem dependent, and depends on approximate idea about the time scale of the problem. The shift δn_w can be chosen as some fraction of Δn_w. We choose $\delta n_w = \lceil \Delta n_w/20 \rceil$. Other parameters of importance are the convergence threshold δ_0 and the number of windows q for checking the convergence in line 11. A large value of q (small value of δ_0) ensures greater confidence in the convergence, but delayed detection of the equilibrium or pseudo-equilibrium. Delayed detection does not affect the extrapolation accuracy but reduces overall efficiency. We select $\delta_0 = 0.1$ and $q = 2$ for all our test cases. The test cases are discussed in chapter 4 and 6.

3.3 Summary

In this chapter, we have explored the critical role of real-time termination in high-fidelity time-domain simulations to effectively utilize time extrapolation models

for acceleration. We introduced two on-the-fly DMD algorithms capable of detecting when a system reaches steady state, equilibrium, or *quasi*-equilibrium, thereby signaling the onset of self-repeating behavior. We discussed how the DMD mode tracking algorithm provides insights into the transition from transient to equilibrium states, and its potential to identify equilibrium behavior from transient data alone. In subsequent chapters, we will apply these on-the-fly algorithms, along with offline DMD methods, to enhance the efficiency of electromagnetic cavity and kinetic plasma simulations.

Chapter 4: Modeling Fields Inside Electromagnetic Cavities using Dynamic Mode Decomposition

Electromagnetic (EM) cavities are essential components in many radio-frequency and microwave systems. Application of cavity resonators includes shielded enclosures, cavity-backed antennas, power sources, filters, amplifiers, oscillators, etc. Time-domain numerical methods such as finite-difference time-domain (FDTD) and the finite-element time-domain (FETD) [3, 4, 96, 97] are often used for simulating EM fields inside cavities and shielded enclosures because of their ability to accurately capture transient, wideband, and nonlinear effects. For wideband applications, frequency-domain solvers can be computationally cumbersome, especially for high$-Q$ resonating structures where it is necessary to solve for a wideband of frequencies with very sharp frequency resolution. In addition, handling of nonlinear and time-varying media is far more challenging in the frequency-domain than in the time-domain [98]. However, the time-domain methods have their own set of challenges, with the primary limitation arising from the sequential nature of the solver [99]. Time-domain

solvers typically use a marching-on-time scheme in a sequential manner, resulting in a long simulation time for querying late-time fields. Also, the well-known Courant–Friedrichs–Lewy (CFL) condition, which poses a restriction on the maximum time step size based on the spatial dimension of mesh elements, prevents an arbitrarily large selection of the time step [4, 3, 99].

Several methods have been proposed to overcome these deficiencies. Although unconditionally stable FDTD algorithms [100, 101, 102, 103, 104] are not limited by the CFL condition, the accuracy for these implicit methods tends to degrade with time step larger than the CFL limit. This happens due to splitting errors and additional numerical dispersion errors. A series of stable and explicit FDTD schemes based on reduced-order models (ROMs) were recently proposed [105, 106, 107]. Reduced-order model based in proper orthogonal decomposition (POD) was also proposed in [108]. Although these methods address several issues related to the time-update scheme, they require knowledge of the state vector, boundary conditions, and the system matrix, which in many cases may be inefficient to store. Also, the electromagnetic response in practice may be required only for certain components of the field, at certain points in space or within a limited region of the cavity, rather than throughout the entire cavity domain. In this respect, purely data-driven approaches for extrapolating the electromagnetic response fare better because they are agnostic to the implementation details of the high-fidelity time-domain solver. Over the years, several

data-driven approaches such as autoregression (AR) [109], autoregressive moving average (ARMA) [110], Prony's method [111, 112], matrix pencil [98, 113], estimation of signal parameters using rotational invariance techniques (ESPIRIT) [114, 115], and state-space based methods [116, 117, 118], have been developed for the time extrapolation of electromagnetic responses. However, most of these methods assume the time-series to be a finite sum of scaled complex exponentials and proceed to find the model parameters that best fit the data, without providing much physical insight about the parameters, especially the scaling factors. State-space based methods [116, 117] go a bit further by assuming a first-order linear time-invariant (LTI) system representation for the underlying dynamics and proceed to solve for the unknown system matrices from available data. However, these methods provide little insight into how the eigenvectors and eigenvalues of extracted system matrices are related to the electromagnetic (physical) resonances. This is not surprising since these methods are primarily developed for modeling single or multi-channel time-series data only and not tailored towards extracting spatial features.

In this chapter, we will first give brief overview of the FDTD algorithm which is used to generate the high-fidelity data for cavity simulations. We will then explore how DMD can be used to extract the resonant modes and frequencies of cavity fields which can be used for analysis, time extrapolation and extraction of the Q-factor. In this work, we use Hankel-DMD (Section 2.2.1) which utilizes the time-delayed *embedding* of the observed state to gain generality [56, 119, 55, 95]. We

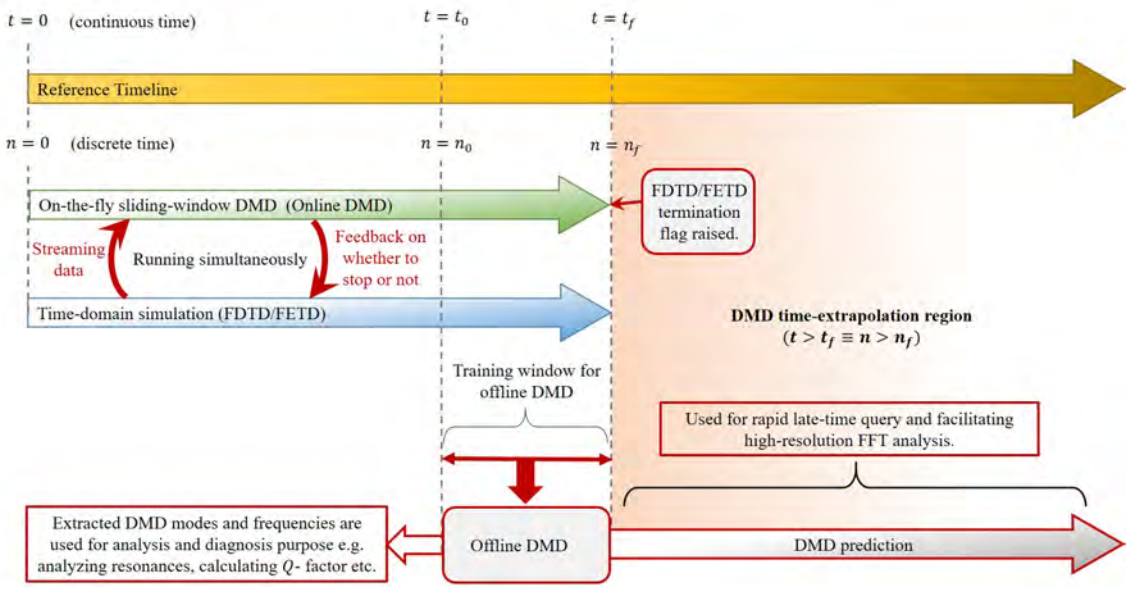


Figure 4.1: DMD assisted FDTD/FETD (DMD-FDTD or DMD-FETD) scheme for analysis and acceleration of high-fidelity time-domain simulations.

show that under certain conditions, the extracted DMD modes exhibit a one-to-one correspondence with the cavity modes. Although DMD has been recently applied for certain radiation problems [45], insights into the physical significance of the DMD modes is still lacking. We address this issue in detail and justify the use of the Hankel matrix approach for modeling electromagnetic resonances especially based on restricted data points. It is important to note that state-space based methods such as eigensystem realization algorithm (ERA) [118] and multi-channel matrix pencil method [113] are related to DMD through similarity transforms [39, 120]. However, DMD is specifically tailored towards the extraction of *spatial* (in addition to

temporal) features, whereas those other methods are not. DMD also exhibits better stability and robustness to noise compared to its Arnoldi-type counterparts [48, 121]. The overall approach is summarized in Fig. 4.1. This chapter is primarily based on [50].

4.1 Finite Difference Time Domain (FDTD) Overview

The Finite-Difference Time-Domain (FDTD) method is a numerical technique for solving Maxwell's equations directly in time and space. It was first introduced by Kane Yee in 1966 [122] and has since become a widely used method for modeling electromagnetic wave interactions with materials. The high-fidelity data used for DMD is obtained from cavity simulations using in-house Cartesian and cylindrical FDTD algorithms with split-field perfectly matched layer (PML) absorbing boundary conditions (ABCs) [123, 124, 125, 97]. We also implement DMD on time-domain data obtained from commercial software such as CST Studio Suite which also uses FDTD algorithm.

4.1.1 Cartesian FDTD

Maxwell's time-dependent curl equations in three dimensions for general materials, including conductivity, are given by [126]:

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu}\nabla \times \mathbf{E} - \mathbf{M} \tag{4.1}$$

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon}\left(\nabla \times \mathbf{H} - \sigma \mathbf{E}\right) + \mathbf{J} \tag{4.2}$$

where $\mathbf{E}$ is the electric field intensity (V/m), $\mathbf{H}$ is the magnetic field intensity (A/m), $\mathbf{J}$ is the electric current density (A/m^2), $\mathbf{M}$ is the equivalent magnetic current density (V/m^2), ϵ is the permittivity, μ is the permeability, and σ is the electrical conductivity of the medium.

The constitutive relations are:

$$\mathbf{D} = \epsilon \mathbf{E}, \quad \mathbf{B} = \mu \mathbf{H},$$

where $\mathbf{D}$ and $\mathbf{B}$ are electric and magnetic flux densities respectively. Yee [122] introduced a system of finite-difference approximations of Maxwell's curl equations. The electric and magnetic field components are staggered in both space and time to achieve second-order accuracy. The spatial derivatives are approximated using central differences, and the time derivatives are approximated using a leapfrog scheme. For example, the finite-difference approximation of the spatial derivative $\frac{\partial H_z}{\partial y}$ at point $(i\Delta x, (j+1/2)\Delta y, k\Delta z)$ and time step n is:

$$\left.\frac{\partial H_z}{\partial y}\right|_{i,j+1/2,k}^{n+1/2} \approx \frac{H_z^{n+1/2}(i,j+1,k) - H_z^{n+1/2}(i,j,k)}{\Delta y} \tag{4.3}$$

The time stepping scheme updates the magnetic field components at half-time steps and the electric field components at integer time steps. The FDTD method

involves updating the magnetic field components using the electric field components and vice versa.

The update equations for the magnetic field components in Cartesian coordinates are:

$$\begin{aligned} H_x^{n+1/2}(i+1/2,j,k) &= H_x^{n-1/2}(i+1/2,j,k) \\ &- \frac{\Delta t}{\mu}\left(\frac{E_z^n(i+1/2,j+1/2,k)-E_z^n(i+1/2,j-1/2,k)}{\Delta y}\right. \\ &\left.-\frac{E_y^n(i+1/2,j,k+1/2)-E_y^n(i+1/2,j,k-1/2)}{\Delta z}\right) \\ &- \Delta t M_x^{n+1/2}(i+1/2,j,k) \end{aligned} \tag{4.4}$$

$$\begin{aligned} H_y^{n+1/2}(i,j+1/2,k) &= H_y^{n-1/2}(i,j+1/2,k) \\ &- \frac{\Delta t}{\mu}\left(\frac{E_x^n(i,j+1/2,k+1/2)-E_x^n(i,j+1/2,k-1/2)}{\Delta z}\right. \\ &\left.-\frac{E_z^n(i+1/2,j+1/2,k)-E_z^n(i-1/2,j+1/2,k)}{\Delta x}\right) \\ &- \Delta t M_y^{n+1/2}(i,j+1/2,k) \end{aligned} \tag{4.5}$$

$$\begin{aligned} H_z^{n+1/2}(i,j,k+1/2) &= H_z^{n-1/2}(i,j,k+1/2) \\ &- \frac{\Delta t}{\mu}\left(\frac{E_y^n(i+1/2,j,k+1/2)-E_y^n(i-1/2,j,k+1/2)}{\Delta x}\right. \\ &\left.-\frac{E_x^n(i,j+1/2,k+1/2)-E_x^n(i,j-1/2,k+1/2)}{\Delta y}\right) \\ &- \Delta t M_z^{n+1/2}(i,j,k+1/2) \end{aligned} \tag{4.6}$$

The update equations for the electric field components are:

$$
\begin{aligned}
&E_x^{n+1}(i, j+1/2, k+1/2) = \\
&\left(\frac{\epsilon}{\Delta t}+\frac{\sigma}{2}\right)^{-1}\left[\left(\frac{H_z^{n+1/2}(i, j+1, k+1/2)-H_z^{n+1/2}(i, j, k+1/2)}{\Delta y}\right.\right. \\
&\left.-\frac{H_y^{n+1/2}(i, j+1/2, k+1)-H_y^{n+1/2}(i, j+1/2, k)}{\Delta z}\right) \\
&\left.+\left(\frac{\epsilon}{\Delta t}-\frac{\sigma}{2}\right) E_x^n(i, j+1/2, k+1/2)\right]+\Delta t J_x^n(i, j+1/2, k+1/2)
\end{aligned} \tag{4.7}
$$

$$
\begin{aligned}
&E_y^{n+1}(i+1/2, j, k+1/2) = \\
&\left(\frac{\epsilon}{\Delta t}+\frac{\sigma}{2}\right)^{-1}\left[\left(\frac{H_x^{n+1/2}(i+1/2, j, k+1)-H_x^{n+1/2}(i+1/2, j, k)}{\Delta z}\right.\right. \\
&\left.-\frac{H_z^{n+1/2}(i+1, j, k+1/2)-H_z^{n+1/2}(i, j, k+1/2)}{\Delta x}\right) \\
&\left.+\left(\frac{\epsilon}{\Delta t}-\frac{\sigma}{2}\right) E_y^n(i+1/2, j, k+1/2)\right]+\Delta t J_y^n(i+1/2, j, k+1/2)
\end{aligned} \tag{4.8}
$$

$$
\begin{aligned}
&E_z^{n+1}(i+1/2, j+1/2, k) = \\
&\left(\frac{\epsilon}{\Delta t}+\frac{\sigma}{2}\right)^{-1}\left[\left(\frac{H_y^{n+1/2}(i+1/2, j+1, k)-H_y^{n+1/2}(i+1/2, j, k)}{\Delta x}\right.\right. \\
&\left.-\frac{H_x^{n+1/2}(i+1, j+1/2, k)-H_x^{n+1/2}(i, j+1/2, k)}{\Delta y}\right) \\
&\left.+\left(\frac{\epsilon}{\Delta t}-\frac{\sigma}{2}\right) E_z^n(i+1/2, j+1/2, k)\right]+\Delta t J_z^n(i+1/2, j+1/2, k)
\end{aligned} \tag{4.9}
$$

To ensure numerical stability, the time increment Δt must satisfy the Courant-Friedrichs-Lewy (CFL) condition:

$$
\Delta t \leq \frac{1}{c\sqrt{\left(\frac{1}{\Delta x}\right)^2+\left(\frac{1}{\Delta y}\right)^2+\left(\frac{1}{\Delta z}\right)^2}} \tag{4.10}
$$

where Δx, Δy, and Δz are the smallest spatial increments, and c is the speed of light in the material.

4.1.2 Cylindrical FDTD

The cylindrical Finite-Difference Time-Domain (FDTD) method is preferred for modeling cylindrical geometries, such as logging tools, cylindrical cavities etc. due to its conformity to cylindrical structures and reduced discretization errors [127]. The cylindrical coordinates are (ρ_c, ϕ_c, z), indicating radius, azimuthal angle and height. The subscript c is used to avoid confusion with correlation coefficient ρ and DMD modes $\boldsymbol{\phi}$. The cylindrical FDTD algorithm involves discretization of Maxwell's equations 4.1 in the cylindrical coordinates:

$$H_{\rho_c}^{n+1/2}(i,j+1/2,k+1/2) = H_{\rho_c}^{n-1/2}(i,j+1/2,k+1/2)+ \frac{\Delta t}{\mu}\left(\frac{E_{\phi_c}^n(i,j+1/2,k+1)-E_{\phi_c}^n(i,j+1/2,k)}{\Delta z} - \frac{E_z^n(i,j+1,k+1/2)-E_z^n(i,j,k+1/2)}{\rho_{ci}\Delta\phi_c}\right) - \Delta t M_{\rho_c}^n(i,j+1/2,k+1/2) \quad (4.11)$$

$$H_{\phi_c}^{n+1/2}(i+1/2,j,k+1/2) = H_{\phi_c}^{n-1/2}(i+1/2,j,k+1/2)+ \frac{\Delta t}{\mu}\left(\frac{E_z^n(i+1,j,k+1/2)-E_z^n(i,j,k+1/2)}{\Delta\rho_{ci}} - \frac{E_{\rho_c}^n(i+1/2,j,k+1)-E_{\rho_c}^n(i+1/2,j,k)}{\Delta z}\right) - \Delta t M_{\phi_c}^n(i+1/2,j,k+1/2) \quad (4.12)$$

$$H_z^{n+1/2}(i+1/2,j+1/2,k) = H_z^{n-1/2}(i+1/2,j+1/2,k)+ \frac{\Delta t}{\mu}\left(\frac{E_{\rho_c}^n(i+1/2,j+1,k)-E_{\rho_c}^n(i+1/2,j,k)}{\rho_{ci+1/2}\Delta\phi_c} - \frac{E_{\phi_c}^n(i+1,j+1/2,k)-E_{\phi_c}^n(i,j+1/2,k)}{2\Delta\rho_{ci}}\right) - \Delta t M_z^n(i+1/2,j+1/2,k) \quad (4.13)$$

The update equations for the electric field components are:

$$
\begin{aligned}
&E_{\rho_c}^{n+1}(i+1/2,j,k) = \\
&\left(\frac{\epsilon}{\Delta t}+\frac{\sigma}{2}\right)^{-1}\Bigg[\frac{H_z^{n+1/2}(i+1/2,j+1/2,k)-H_z^{n-1/2}(i+1/2,j-1/2,k)}{\rho_{ci+1/2}\Delta\phi_c} \\
&-\frac{H_{\phi_c}^{n+1/2}(i+1/2,j,k+1/2)-H_{\phi_c}^{n-1/2}(i+1/2,j,k-1/2)}{\Delta z} \\
&+\left(\frac{\epsilon}{\Delta t}-\frac{\sigma}{2}\right)E_{\rho_c}^{n}(i+1/2,j,k)\Bigg]+\Delta t J_{\rho_c}^{n}(i+1/2,j,k)
\end{aligned}
\tag{4.14}
$$

$$
\begin{aligned}
&E_{\phi_c}^{n+1}(i,j+1/2,k) = \\
&\left(\frac{\epsilon}{\Delta t}+\frac{\sigma}{2}\right)^{-1}\Bigg[\frac{H_{\rho_c}^{n+1/2}(i,j+1/2,k+1/2)-H_{\rho_c}^{n-1/2}(i,j+1/2,k-1/2)}{\Delta z} \\
&-\frac{H_z^{n+1/2}(i+1/2,j,k+1/2)-H_z^{n-1/2}(i-1/2,j,k+1/2)}{\Delta\rho_{ci}} \\
&+\left(\frac{\epsilon}{\Delta t}-\frac{\sigma}{2}\right)E_{\phi_c}^{n}(i,j+1/2,k)\Bigg]+\Delta t J_{\phi_c}^{n}(i,j+1/2,k)
\end{aligned}
\tag{4.15}
$$

$$
\begin{aligned}
&E_{z}^{n+1}(i,j,k+1/2) = \\
&\left(\frac{\epsilon}{\Delta t}+\frac{\sigma}{2}\right)^{-1}\Bigg[\frac{H_{\phi_c}^{n+1/2}(i+1/2,j,k+1/2)-H_{\phi_c}^{n-1/2}(i-1/2,j,k+1/2)}{\Delta\rho_{ci}} \\
&+\frac{H_{\phi_c}^{n+1/2}(i+1/2,j,k+1/2)-H_{\phi_c}^{n-1/2}(i-1/2,j,k+1/2)}{\rho_{ci}\Delta\phi_c} \\
&+\left(\frac{\epsilon}{\Delta t}-\frac{\sigma}{2}\right)E_{z}^{n}(i,j,k+1/2)\Bigg]+\Delta t J_{z}^{n}(i,j,k+1/2)
\end{aligned}
\tag{4.16}
$$

The spatial derivatives in cylindrical coordinates are approximated using central finite differences, and the time derivatives are approximated using a leapfrog scheme:

$$
\frac{\partial F}{\partial \rho_c} \approx \frac{F^{n}(i+1/2,j,k)-F^{n}(i-1/2,j,k)}{\Delta\rho_c} \tag{4.17}
$$

$$
\frac{\partial F}{\partial t} \approx \frac{F^{n+1/2}(i,j,k)-F^{n-1/2}(i,j,k)}{\Delta t} \tag{4.18}
$$

To ensure numerical stability, the time increment Δt must satisfy the Courant-Friedrichs-Lewy (CFL) condition:

$$\Delta t \leq \frac{1}{v\sqrt{\left(\frac{1}{\Delta\rho_{c\min}}\right)^2 + \left(\frac{1}{\rho_{c\min}\Delta\phi_c}\right)^2 + \left(\frac{1}{\Delta z_{\min}}\right)^2}} \tag{4.19}$$

where $\Delta\rho_{c\min}$, $\rho_{c\min}\Delta\phi_c$, and $\Delta z_{\min}$ are the smallest spatial increments, and v is the maximum phase speed.

ABCs such as the Perfectly Matched Layer (PML) are used to prevent reflections at the boundaries of the simulation domain. The cylindrical FDTD method provides a powerful tool for modeling electromagnetic wave propagation in cylindrical geometries, accommodating anisotropic materials, and ensuring numerical stability through appropriate discretization and boundary conditions.

4.1.3 Split-Field PML-FDTD

Split-field Perfectly Matched Layer - Finite-Difference Time-Domain (PML-FDTD) is a method used to simulate electromagnetic wave propagation in inhomogeneous media [123, 124, 125, 97]. This approach incorporates the PML absorbing boundary condition to effectively eliminate spurious reflections from the boundaries of the computational domain. In the split-field PML-FDTD formulation, the electromagnetic field components are split into subcomponents. This technique introduces matched artificial electric and magnetic conductivities within the PML region, which makes the PML implementation transparent to the constitutive properties of the medium.

Essentially, the PML can be considered an analytic continuation of Maxwell's equations into a complex spatial domain, facilitating a more accurate representation of boundary conditions in the simulation. This method is particularly useful for time-domain simulations involving complex media, such as layered microstrip antennas, biological tissues, or subsurface sensing applications. By ensuring zero reflection coefficients at the PML interface and exponential decay along the normal direction, split-field PML-FDTD significantly enhances the accuracy of simulations for scenarios involving inhomogeneous media.

4.2 Modeling Cavity Resonances

The phenomena of cavity resonance can be explained through superposition of several standing waves. Limitation of standard DMD in capturing standing waves and possible solution by stacking snapshot matrices had been earlier noted in [39]. However, not much attention has been given to why and how such stacking works. In this section, we analyze the challenges in spatio-temporal modeling of cavity resonance in the light of recent advances in the Hankel-DMD literature [56, 55].

4.2.1 Physical Modes

In the absence of sources, the oscillating electric field at any location inside a cavity can be expressed by summation of resonance modes as follows [128],

$$e(t) = \sum_{m} e_m(t), \tag{4.20}$$

where for perfect electric conductor (PEC) walls, the electric field $e_m(t)$ corresponding to the m^{th} resonance mode satisfies [128]

$$\frac{\partial^2 e_m}{\partial t^2} + \omega_m^2 e_m = 0. \tag{4.21}$$

The resonance frequency $\omega_m = k_m^2/\mu\epsilon$ corresponds to m^{th} cavity mode, where k_m is the corresponding wave number, with μ and ϵ being the material permeability and permittivity respectively. The resonance frequency ω_m can be complex for complex μ or ϵ. The solution to (4.21) is given by

$$e_m(t) = a_m \; e^{\tilde{\omega}_m t} + a_m^* \; e^{\tilde{\omega}_m^* t}, \tag{4.22}$$

where $\tilde{\omega}_m = j\omega_m$, with $\tilde{\omega}_m$, ω_m, $a_m \in \mathbb{C}$ and $j = \sqrt{-1}$. Due to bandlimited excitation, the summation in (4.20) can be truncated to finite number of terms for practical purpose. In light of (4.22), the electric field sampled at any N points in space inside the cavity can be represented in terms of a column vector as

$$\mathbf{e}(t) = \sum_{m=1}^{M} \mathbf{a}_m \; e^{\tilde{\omega}_m t} + \mathbf{a}_m^* \; e^{\tilde{\omega}_m^* t}, \tag{4.23}$$

where $\mathbf{e} \in \mathbb{R}^N, \mathbf{a}_m \in \mathbb{C}^N$. It is interesting to note the similarity between (4.23) and (2.24). This will be further discussed next.

4.2.2 DMD Modeling of Physical Modes

Assuming discrete-time evolution with sampling interval Δ_t, the expression in (4.23) can be rewritten as

$$\begin{aligned}\mathbf{e}(k\Delta_t) = \mathbf{e}_k &= \sum_{m=1}^{M} \mathbf{a}_m \; e^{\tilde{\omega}_m k\Delta_t} + \mathbf{a}_m^* \; e^{\tilde{\omega}_m^* k\Delta_t} \\ &= \sum_{p=1}^{2M} \mathbf{a}_p \; e^{\tilde{\omega}_p k\Delta_t} \\ \Rightarrow \mathbf{e}_k &= \sum_{p=1}^{2M} \mathbf{a}_p \; \tilde{\lambda}_p^k = \sum_{p=1}^{2M} \tilde{a}_p \mathbf{v}_p \; \tilde{\lambda}_p^k, \end{aligned} \tag{4.24}$$

where $\tilde{\lambda}_p = e^{\tilde{\omega}_p \Delta_t}$, and $\mathbf{a}_p = \tilde{a}_p \mathbf{v}_p$ for some $\tilde{a} \in \mathbb{C}$. Any quantity in the form of (4.24) can be expressed as the solution of a discrete-time first-order linear system with eigenvalues $\tilde{\lambda}_p \in \mathbb{C}$ and eigenvectors $\mathbf{v}_p \in \mathbb{C}^N$ as long as $\mathbf{v}_p$ or equivalently $\mathbf{a}_p$ are linearly independent for $p = 1, 2, \ldots, 2M$ [56]. In other words, DMD can model cavity resonance with good accuracy if $\mathbf{a}_p$ are linearly independent.

The first restriction arises from the spectral ($M_s = 2M$) and spatial (N) complexities. For $\mathbf{a}_p$ to be linearly independent, $N \geq 2M$. Even if the latter condition is satisfied, $\mathbf{a}_p$ i.e. the *partial* spatial pattern of the cavity modes may still be linearly dependent, e.g. TM_{110} and TM_{310} modes (indexing along x, y and z) when observed *restricted* to the $yz-$plane of a closed rectangular cavity. The stacks of time-delayed observables as in (2.27) have been used to overcome the issue regarding linear dependency [56]. The significance of time-delayed *embedding* for partial observations of

electromagnetic fields can be better understood in the context of FDTD time-update equations.

Without loss of generality, let us assume the number of integer and half-integer grid points be N in a three-dimensional (3-D) space. The explicit time-update of electric ($\mathbf{e}$) and magnetic fields ($\mathbf{h}$) in source free region on staggered FDTD grid can be written as [105],

$$\begin{bmatrix} \mathbf{e}^{(n+1)} \\ \mathbf{h}^{(n+\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3N} - \Delta t^2 \mathbf{M} & \Delta t \mathbf{D_h} \\ -\Delta t \mathbf{D_e} & \mathbf{I}_{3N} \end{bmatrix} \begin{bmatrix} \mathbf{e}^{(n)} \\ \mathbf{h}^{(n-\frac{1}{2})} \end{bmatrix}, \tag{4.25}$$

where $\mathbf{D_h}$ and $\mathbf{D_e}$ are the dicretization of the curl operator operating on $\mathbf{h}$ and $\mathbf{e}$ respectively, and $\mathbf{M} = \mathbf{D_h D_e}$. As mentioned earlier, due to the CFL limit, the time step Δt for FDTD/FETD simulation is usually quite smaller than required for capturing the time variation of the fields. We can write (4.25) in terms of DMD sampling interval $\Delta_t = \Delta t \Delta_n$, expressing the time evolution of the state from one time sample to next ($(k-1)^{\text{th}}$ to k^{th}) as

$$\begin{bmatrix} \mathbf{e}^{(n+\Delta_n)} \\ \mathbf{h}^{(n-\frac{1}{2}+\Delta_n)} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3N} - \Delta t^2 \mathbf{M} & \Delta t \mathbf{D_h} \\ -\Delta t \mathbf{D_e} & \mathbf{I}_{3N} \end{bmatrix}^{\Delta_n} \begin{bmatrix} \mathbf{e}^{(n)} \\ \mathbf{h}^{(n-\frac{1}{2})} \end{bmatrix} \tag{4.26}$$

$$\Rightarrow \begin{bmatrix} \mathbf{e}_\nu \\ \mathbf{h}_\nu \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3N} - \Delta t^2 \mathbf{M} & \Delta t \mathbf{D_h} \\ -\Delta t \mathbf{D_e} & \mathbf{I}_{3N} \end{bmatrix}^{\Delta_n} \begin{bmatrix} \mathbf{e}_{\nu-1} \\ \mathbf{h}_{\nu-1} \end{bmatrix} \tag{4.27}$$

Let

$$\begin{bmatrix} \mathbf{e}_\nu \\ \mathbf{h}_\nu \end{bmatrix} = \mathbf{u}_\nu \;\; ; \begin{bmatrix} \mathbf{I}_{3N} - \Delta t^2 \mathbf{M} & \Delta t \mathbf{D_h} \\ -\Delta t \mathbf{D_e} & \mathbf{I}_{3N} \end{bmatrix}^{\Delta_n} = \mathbf{G}. \tag{4.28}$$

Equation (4.27) can be written as first-order linear evolution of $\mathbf{u}$ with state transition matrix $\mathbf{G}_{6N \times 6N}$,

$$\mathbf{u}_\nu = \mathbf{G}\mathbf{u}_{\nu-1}. \tag{4.29}$$

Without loss of generality, any partial state observation $\tilde{\mathbf{u}}$ can be expressed as observing the first $\tilde{N}$ elements ($\tilde{N} < 6N$) of $\mathbf{u}$. We can re-write (4.29) as

$$\begin{bmatrix} \tilde{\mathbf{u}}_\nu \\ \widehat{\mathbf{u}}_\nu \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_{\nu-1} \\ \widehat{\mathbf{u}}_{\nu-1} \end{bmatrix}, \tag{4.30}$$

where $\tilde{\mathbf{u}} \in \mathbb{R}^{\tilde{N}}$, $\widehat{\mathbf{u}} \in \mathbb{R}^{6N-\tilde{N}}$, $\mathbf{G}_{11} \in \mathbb{R}^{\tilde{N}\times\tilde{N}}$, $\mathbf{G}_{12} \in \mathbb{R}^{\tilde{N}\times(6N-\tilde{N})}$, $\mathbf{G}_{21} \in \mathbb{R}^{(6N-\tilde{N})\times\tilde{N}}$ and $\mathbf{G}_{22} \in \mathbb{R}^{(6N-\tilde{N})\times(6N-\tilde{N})}$. After some basic algebraic manipulation, the time evolution of $\tilde{\mathbf{u}}$ can be written as,

$$\tilde{\mathbf{u}}_\nu = \mathbf{G}_{11}\tilde{\mathbf{u}}_{\nu-1} + \sum_{q=0}^{d-2} \mathbf{G}_{12}\mathbf{G}_{22}^{q}\mathbf{G}_{21}\tilde{\mathbf{u}}_{\nu-2-q} + \mathbf{G}_{12}\mathbf{G}_{22}^{d-1}\widehat{\mathbf{u}}_{\nu-d} \tag{4.31}$$

For sufficiently large d, the final term in (4.31) typically corresponds to the transient response [95]. Letting $\tilde{\mathbf{G}}_i = \mathbf{G}_{12}\mathbf{G}_{22}^{i-2}\mathbf{G}_{21}$ ($i = 2, 3, \ldots, d$), and neglecting the last term, we can approximately write

$$\tilde{\mathbf{u}}_\nu \approx \mathbf{G}_{11}\tilde{\mathbf{u}}_{\nu-1} + \tilde{\mathbf{G}}_2\tilde{\mathbf{u}}_{\nu-2} + \ldots + \tilde{\mathbf{G}}_d\tilde{\mathbf{u}}_{\nu-d}. \tag{4.32}$$

The striking similarity between (4.32) and (2.25) is noteworthy. Equation (4.32) essentially tells us that the time-delayed observations or higher-order terms are needed to improve the modeling of the partial state dynamics. The Hankel-DMD (2.25) strives to find a solution such that the snapshots of the state under consideration $\mathbf{x}$, best fits a relation of the form (4.32) in the least-square sense.

4.2.3 DMD Modes and Cavity Modes

Assuming the linear independence among $\mathbf{a}_p$ (4.24) is ensured through sufficient stacking, there exists a matrix $\mathbf{A_e}$ exactly satisfying $[\![\mathbf{e}]\!]_k = \mathbf{A_e}[\![\mathbf{e}]\!]_{k-1}$. If DMD is

able to extract exact eigenvectors and eigenvalues of $\mathbf{A_e}$, the DMD modes (truncated) and frequencies will match exactly with the cavity modes and resonance frequencies. However, as with any other data-driven method, DMD accuracy depends upon the quality of the training dataset, the judicious choice of training parameters, and the finite machine precision. It is also important to keep in mind that DMD seeks to estimate the eigenvalues and eigenvectors through a reduced-order approximation. As long as the training dataset (snapshot matrices) is "rich" enough, the extracted DMD modes and frequencies will be a close approximation of the cavity modes and resonance frequencies. The only exception occurs for degenerate modes sharing a common resonance frequency, in which case the extracted DMD modes extracted are just a linear combination of the spatial pattern of degenerate modes, which is further discussed in the next subsection.

4.2.4 Degenerate Modes

The case of degenerate resonance modes needs special attention since the one to one correspondence of DMD modes with that of the cavity modes fails in the presence of degenerate modes. Due to the presence of certain symmetries in the cavity geometry, cavity modes with different spatial pattern might share the same resonance frequency. A typical example is the closed regular rectangular cavity where length of one side is integer multiple of any other two sides. Note that the DMD formulation relies on the eigenvalues λ_p in (4.23) being distinct [56, 95, 39].

For repeating eigenvalues, the DMD formulation still remains valid after we club together the terms with same eigenvalues.

Without loss of generality, consider mode 1 and mode 2 to be degenerate ($\tilde{\omega}_1 = \tilde{\omega}_2 = \tilde{\omega}_{12}$) in (4.23). Combining those two terms,

$$\mathbf{e}(t) = \mathbf{a}_{12}\ e^{\tilde{\omega}_{12}t} + \overline{\mathbf{a}}_{12}\ e^{\overline{\tilde{\omega}}_{12}t} + \sum_{m=3}^{M} \mathbf{a}_m\ e^{\tilde{\omega}_m t} + \overline{\mathbf{a}}_m\ e^{\overline{\tilde{\omega}}_m t}, \tag{4.33}$$

where $\mathbf{a}_{12} = \mathbf{a}_1 + \mathbf{a}_2$. Naturally, the extracted DMD mode corresponding to $\tilde{\omega}_{12}$ will be approximately $\mathbf{a}_{12} = \mathbf{a}_1 + \mathbf{a}_2$. We can say that in the presence of degenerate modes, DMD will return linear combination of those degenerate modes as a single DMD mode. This is further explained in section 4.4.1 with the example of a losy rectangular cavity.

4.3 Selection of DMD Training Parameters

Like any other data-driven method, the DMD performance also depends on the appropriate choice of training parameters or hyperparameters. The crucial training parameters are the DMD time-window width Δt_w, the DMD sampling interval Δ_t (not to be confused with the FDTD time step interval Δt), the rank r for truncating SVD matrices, the number of Hankel stacks d, and the end point of the DMD window $t_f (= t_0 + \Delta t_w)$ (t_0 being the starting of DMD window), essentially the timestamp for terminating the FDTD/FETD simulation. We will discuss the empirical selection of these parameters and describe our proposed online sliding-window DMD algorithm

to detect t_f on-the-fly i.e. concomitantly with the ongoing FDTD/FETD simulation and in an unsupervised fashion.

4.3.1 Selecting $\Delta t_w, \Delta_t, r$ and d

The DMD window Δt_w should be large enough to capture multiple cycles of any oscillatory behavior, in our case the EM field oscillations inside the cavity. For offline applications, the lowest frequency component f_{min} can be easily extracted using FFT and Δt_w can be chosen accordingly. However, for on-the-fly applications (without the availability of the entire time history), Δt_w can be chosen based on a priori knowledge of the maximum cavity dimension, say L_{max}. A conservative lower bound for the possible resonance frequencies can be the fundamental resonance frequency of a cuboid of side length L_{max}. For inhomogeneous media, a homogeneous cuboid with volume average of ϵ and μ can be chosen. Once the lowest frequency is decided, as a rule of thumb we take Δt_w to be wide enough to incorporate at least 30 cycles of those oscillations.

The time interval Δ_t between the DMD snapshots plays a pivotal role in determining the DMD extrapolation accuracy. Excessive dense sampling (very small Δ_t) poses an ill-conditioned problem [95]. Also, smaller the sampling interval, more rapidly the error accumulates for long-term prediction. On the other hand, very large Δ_t might violate the Nyquist criterion. The ideal choice for Δt is based on the Nyquist limit. Once the window width Δt_w is decided, we extract the maximum frequency f_{max} using FFT and choose $\Delta_t = 1/(2f_{max})$. The value of r decides the

number of DMD modes $\widehat{M}$ ($r \leq 2\widehat{M}$) as in (2.23) and (2.24). In this work we choose r based on optimal hard thresholding, as discussed in [129, 130]. As far as the number of Hankel stacks d is concerned, there is no definitive way of selecting d in the absence of the full-system information [56, 95]. Even when available, the calculation of the optimum value of d can be very involved and computationally expensive. A rather straightforward way is to check convergence of the DMD solution within the training region as d is gradually increased in steps. Note that very high value of d might be computationally restrictive.

Another key parameter is the end point t_f for the training region or the DMD window. The quality of the training data is critical to the success of any data-driven method. In typical DMD applications, the transient region is conveniently ignored to ensure stability and good extrapolation accuracy for long-term predictions. However, in order to truly leverage the extrapolation ability of DMD, it is important to identify, in real-time, the onset of equilibrium or *pseudo* or *quasi* equilibrium dynamics. In a cavity, after the transient phase is over, the oscillating EM fields in a source free region can be expressed as

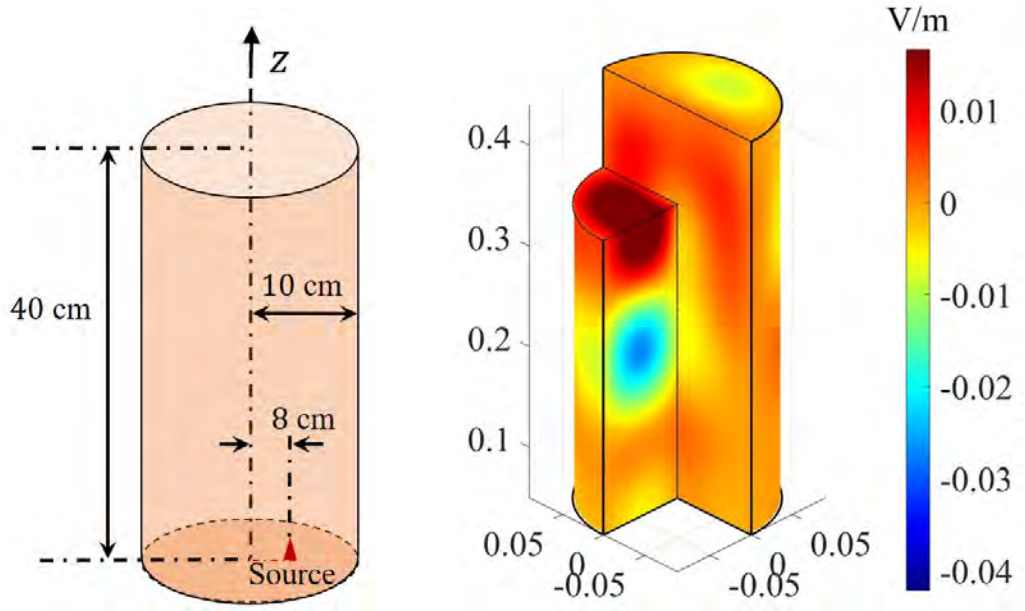
$$\mathbf{e}(t) = \sum_{m=1}^{M} \tilde{\mathbf{a}}_m e^{-\alpha_m t} \cos(\omega_{R,m} t + \theta), \tag{4.34}$$

where $\tilde{\mathbf{a}}_m \in \mathbb{R}^N$ and $\alpha_m, \omega_{R,m} \in \mathbb{R}^+$. Equation (4.34) is merely another way of representing the same fields as in (4.23), with α_m and $\omega_{R,m}$ corresponding to the real (imaginary) and imaginary (real) part of $\tilde{\omega}_m$ (ω_m). When $\alpha_m \neq 0$ (lossy cavity), the system is strictly not in equilibrium as the fields gradually decay with time. But since

α_m and ω_m remain constants after the cavity modes have settled in, we refer to this state as *quasi* equilibrium. We use the on-the-fly algorithm described in algorithm 3 for real-time detection of quasi equilibrium.

4.4 DMD Applied to Cavity Fields

We first demonstrate our DMD assisted FDTD/FETD scheme (Fig .4.1) for two simple test cases, *i)* closed cylindrical cavity and *ii)* closed rectangular cavity with lossy dielectric. Since the theoretical solution for resonance frequencies and mode shapes are available for canonical geometries, it is easier to check the sanity of the extracted DMD modes and frequencies. Next we demonstrate our method for a complicated L-shaped cavity with slot, and a coaxial cavity. Note that first three examples are simulated using in-house FDTD code, whreas coaxial cavity is simulated using CST software. In order to make it more challenging, we place a metal object inside the L-shaped cavity with partial dielectric loading and model the fields only over a two-dimensional (2-D) cross-section. Note that for all the test case we are interested in modeling the time evolution of $z-$directed electric field $\mathbf{e}_z$ using DMD. In the context of 2.2, $\mathbf{x} = \mathbf{e}_z = [e_{z,1}\ e_{z,2}\ \dots\ e_{z,N}]^{\mathrm{T}}$, where superscript 'T' denotes transpose and $e_{z,i}$ represent the $z-$directed electric field at i^{th} grid point in space $(i = 1, 2, \dots, N)$.



(a) Cylindrical cavity dimensions.

(b) Snapshot of $\mathbf{e}_z$ at 23.69 ns.

Figure 4.2: Closed cylindrical cavity test case

4.4.1 Closed Cylindrical Cavity

A closed cylindrical cavity of length 40 cm and radius 10 cm (Fig. 4.2a) is simulated using an in-house FDTD code for 5×10^5 time steps ($t = 370.13$ ns) with step size of $\Delta t = 0.740$ ps. The cavity is excited using a $z-$directed electric field source having the profile of a Gaussian pulse with center frequency of 2 GHz and 1 ns pulse width (Fig. 4.3). The walls are made of aluminium with conductivity $\sigma_{wall} = 3.7 \times 10^7$ S/m , relative permittivity $\epsilon_{r,wall} = 1.7$ and relative permeability $\mu_{r,wall} = 1$. Inside of the cavity is vacuum/air with $\epsilon_{r,cav} = \mu_{r,cav} = 1$ and $\sigma_{cav} = 0$.

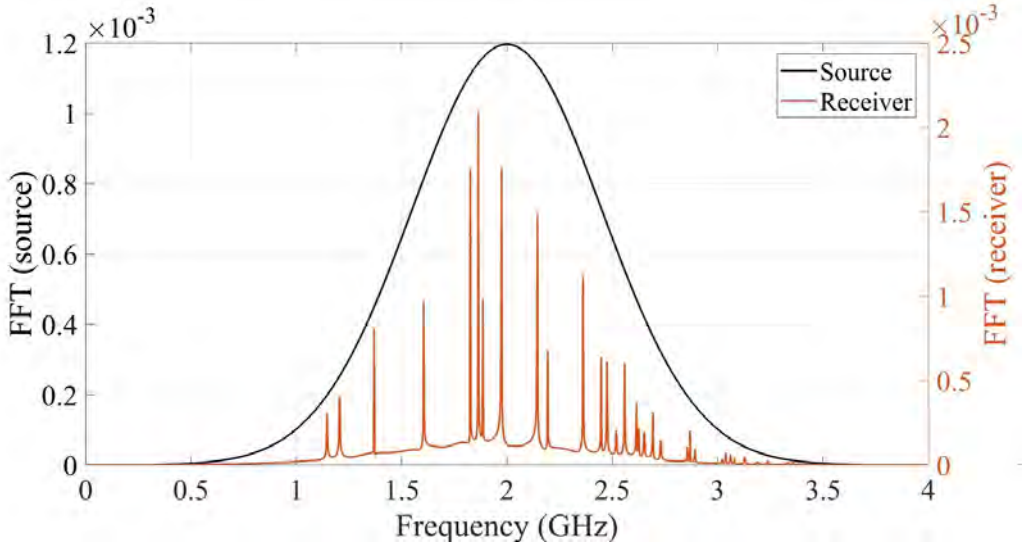
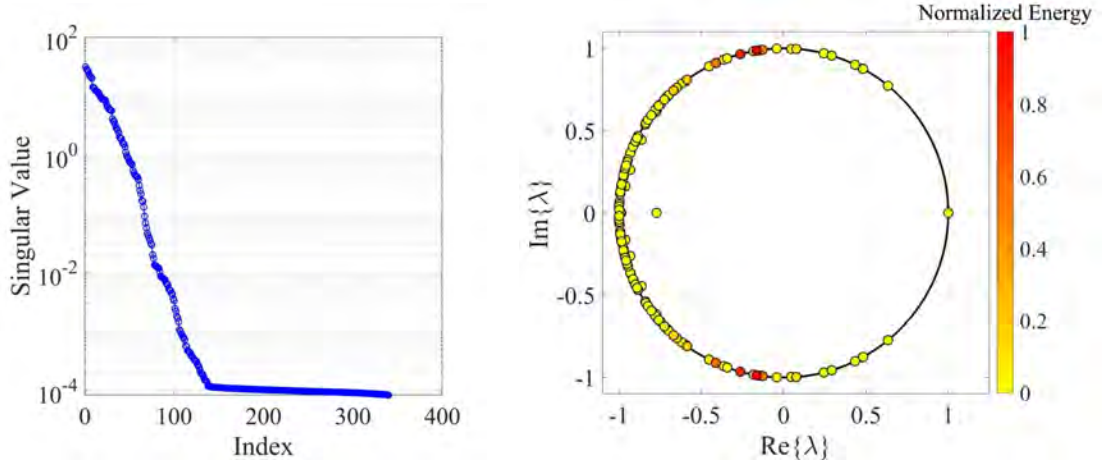


Figure 4.3: Frequency response for closed cylindrical case.

We are interested in modeling the time evolution of $z-$directed electric field (Fig. 4.2b) inside the entire cavity using DMD. The frequency response of the cavity is shown in Fig. 4.3, where FFT of 20 random points in space are averaged to generate the receiver FFT plot.

On-the-Fly DMD

The on-the-fly (online) DMD algorithm 3 is applied with $\Delta t_w = 51.68$ ns, $\Delta_t = 0.15$ ns, $d = 10$, and $t_{qry} = 222.07$ ns. The equilibrium is detected at $t_f = 64.63$ ns, flagging the end of FDTD simulation. The final window of the sliding-window which spans from $t_0 = t_f - \Delta t_w = 12.95$ ns to $t_f = 64.63$ ns, is used to extrapolate $\mathbf{e}_z$ beyond t_f, and extract relevant features for analysis purpose.



(a) Singular values (b) DMD eigenvalues.

Figure 4.4: Singular values and DMD eigenvalues for the closed cylinder case (Fig. 4.2). DMD eigenvalues are color-coded according to the normalized energy of the corresponding DMD modes.

Analysis

The sharp decay in singular values (Fig. 4.4a) indicates an underlying lower-dimensional structure, which is further validated by Fig. 4.3 which shows that finite number of cavity modes dominate the dynamics. Fig. 4.6 shows the first eight most energetic[2] DMD modes. The corresponding DMD eigenvalues are shown in Fig. 4.4b. Eigenvalues lying on or inside the unit circle (black) ensures stability for the reduced-order time-extrapolation (note that DMD eigenvalues are related to the DMD frequencies). Due to numerical noise, some of the DMD eigenvalues might

[2]The energy of a DMD mode is calculated by averaging the 2-norm of the DMD modes over all the time-samples inside the DMD window.

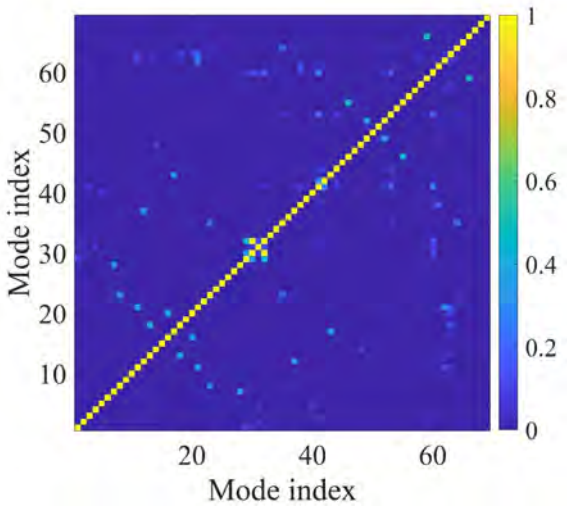


Figure 4.5: Correlation coefficient among DMD modes for closed cylindrical case.

lie outside the unit circle, causing instability. For our application, before performing the time-extrapolation, in order to ensure stability for long-term predictions, we force those eigenvalues radially to place them on the unit circle. As we see in Fig. 4.6, the extracted DMD modes resemble the familiar TM_{npq} modes (n, p and q corresponds to ϕ_c, ρ_c and z direction, not to be confused with the time step n and other indices used earlier) of a closed cylindrical cavity. This becomes obvious when we compare the corresponding DMD frequencies (f_{DMD}) with the theoretical ones (f_{theo}) and the FFT peaks (f_{FDTD}) obtained from long run of FDTD (Fig .4.3). This is summarized in table 4.1, where we compare the frequencies of the top eight most energetic DMD modes corresponding to almost 90% of the total energy. The cross correlation coefficient between the DMD modes (Fig. 4.5) further supports the claim

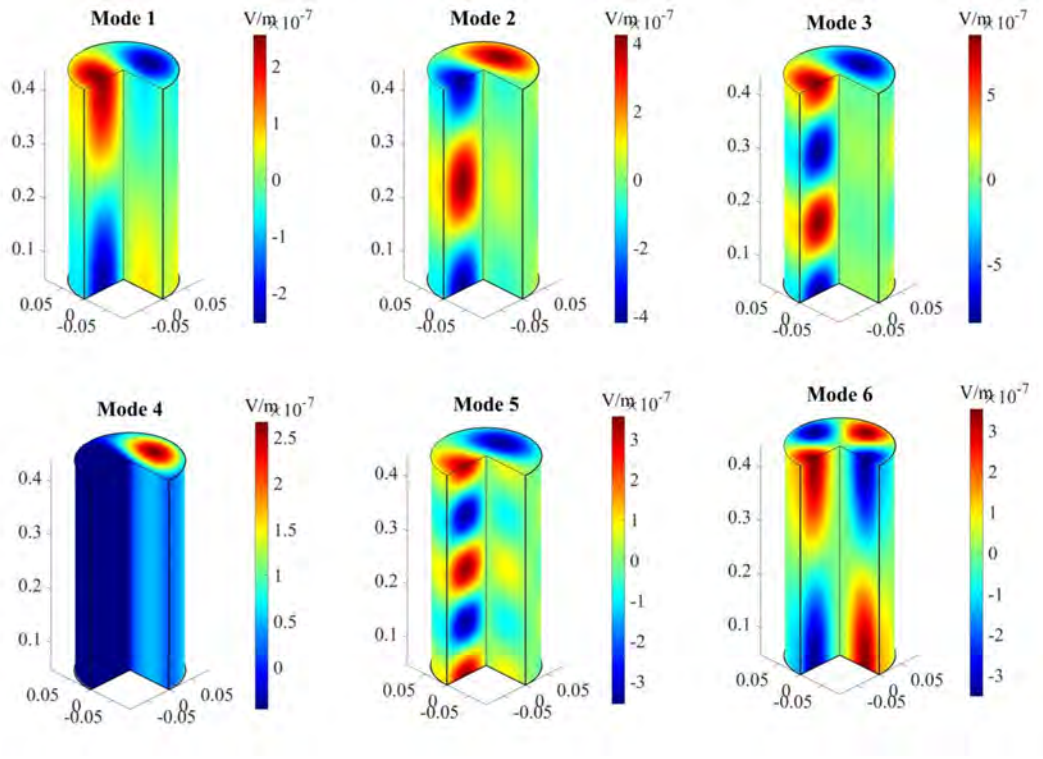


Figure 4.6: First six most energetic DMD modes for the setup in Fig. 4.2. The DMD modes resemble typical transverse magnetic (TM) modes of a closed cylindrical cavity. Mode 1 (ϕ_1) resembles TM_{111} mode, ϕ_2 TM_{112} mode and so on.

that DMD modes are indeed the same as cavity modes. Since the cavity modes are spatially orthogonal to each other, ideally we expect all the off-diagonal elements in Fig. 4.5 to be zero. However, similar to other data-driven methods, DMD is also prone to numerical noise and it is essentially a reduced-order approximation of the original physics. For the first 16 DMD modes, which corresponds to $> 90\%$ of the

total energy, the cross correlation is approximately zero. Beyond that the numerical noise and artifacts start corrupting the correlation values. We take the absolute value of the well-known modal assurance criterion (MAC) [74, 47] as our correlation coefficient for correlating two spatial modes $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$,

$$\mathrm{MAC}(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \frac{\left|\boldsymbol{\phi}_i^{\mathrm{T}}\ \boldsymbol{\phi}_j^{*}\right|^2}{(\boldsymbol{\phi}_i^{\mathrm{T}}\boldsymbol{\phi}_i^{*}) \cdot (\boldsymbol{\phi}_j^{\mathrm{T}}\boldsymbol{\phi}_j^{*})}. \tag{4.35}$$

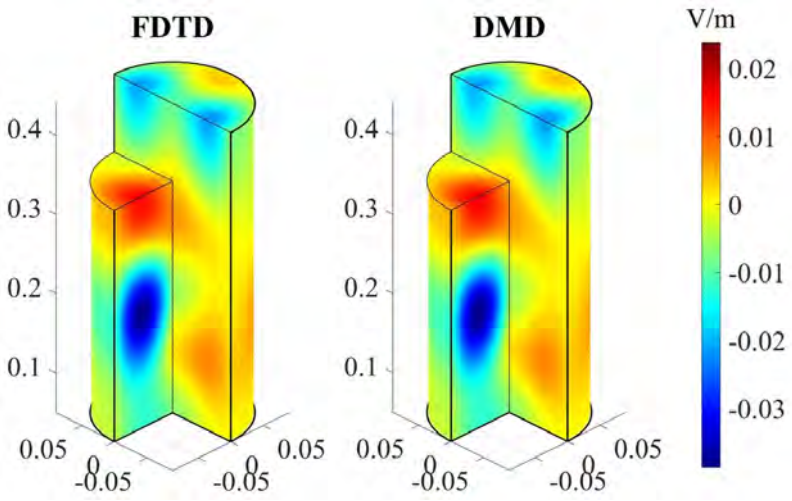


Figure 4.7: FDTD vs DMD prediction for closed cylindrical cavity.

Time-Extrapolation

The extracted DMD modes ($\boldsymbol{\phi}_i$) and frequencies ($\hat{\omega}_i$) are used for time-extrapolation of $\mathbf{e}_z$ beyond t_f (FDTD termination). The long-term DMD prediction matches well with the FDTD result as can be seen in Fig. 4.7. The relative 2-norm ($||\cdot||_2$) error

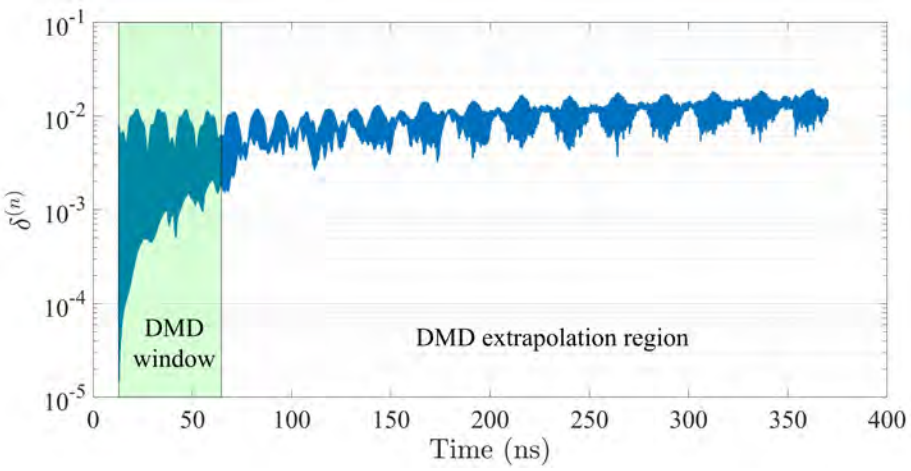


Figure 4.8: DMD reconstruction error for closed cylindrical cavity.

in DMD reconstruction is given by,

$$\delta^{(n)} = \frac{||\hat{\mathbf{e}}_z^{(n)} - \mathbf{e}_z^{(n)}||_2}{\langle ||\mathbf{e}_z^{(n)}||_2 \rangle_{[n_0, n_{ex}]}}, \tag{4.36}$$

where $\langle \cdot \rangle_{[n_0, n_{ex}]}$ denotes average over $[n_0, n_{ex}]$ time steps with n_0 being the discrete equivalent of t_0 and n_{ex} being the time step upto which we wish to check the extrapolation accuracy. Fig. 4.8 shows the DMD extrapolation error around 1%, which is sufficient for most applications. One of the motivations for rapid time-extrapolation is to facilitate the frequency domain analysis. High resolution in the frequency domain requires large number of time steps, in other words, long simulation time. Short run of FDTD/FETD simulation results in a coarse FFT plot which might not be accurate, as can be seen (blue curve) in Fig. 4.9. However, with the help of DMD extrapolation, even with short run of FDTD, we get very accurate FFT plot, as

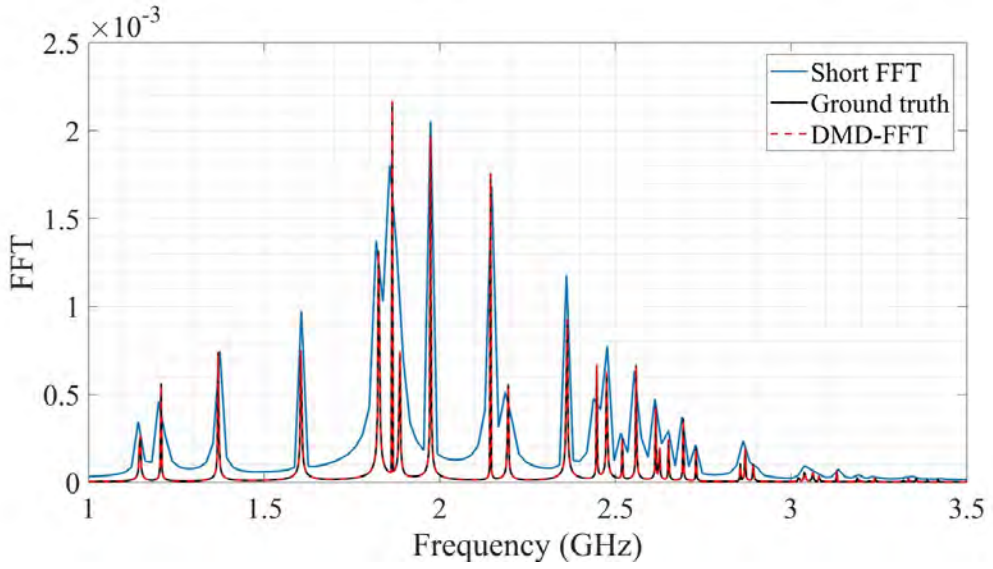


Figure 4.9: FFT with DMD extrapolated fields for closed cylindrical case.

suggested by the close overlap of the red dashed line (DMD-FFT) with the black one (long run of FDTD) in Fig. 4.9.

4.4.2 Closed Rectangular Cavity

A closed rectangular cavity of dimension 40 cm × 20 cm × 10 cm (Fig. 4.10) is simulated using the in-house FDTD code for 6×10^4 time steps ($t = 491.08$ ns) with the step size of $\Delta t = 8.185$ ps. The walls are made of aluminum and inside of the cavity is filled with a dielectric material with $\epsilon_{r,cav} = 2.5$ and $\mu_{r,cav} = 1$. We introduce loss through non-zero value of conductivity for the dielectric, $\sigma_{cav} = 0.0005$ S/m. A $z-$directed current source is used to excite the cavity. The excitation waveform is a Gaussian pulse with center frequency of 2 GHz and pulse width of 3 ns (Fig. 4.11).

Table 4.1: Comparison between FDTD, DMD and theoretical resonance frequencies for closed cylinderical cavity. The frequency of i^{th} DMD mode is denoted by f_i.

$\mathbf{f}_{\text{FDTD}}$ **(GHz)**	$\mathbf{f}_{\text{DMD}}$ **(GHz)**	$\mathbf{f}_{\text{theo}}$ **(GHz)**
1.864	1.865 (f_1)	1.866 (TM_{111})
1.975	1.974 (f_2)	1.976 (TM_{112})
2.145	2.144 (f_3)	2.146 (TM_{113})
1.826	1.827 (f_4)	1.828 (TM_{110})
2.361	2.362 (f_5)	2.364 (TM_{114})
2.475	2.476 (f_6)	2.479 (TM_{211})
1.886	1.886 (f_7)	1.888 (TM_{014})
1.605	1.605 (f_8)	1.606 (TM_{013})

This particular test case is carefully chosen since it reveals DMD's strength and limitations in the following ways,

- The theoretical expressions for resonance frequencies and mode shapes are available for a rectangular cavity. Thus, it is easy to check whether the extracted DMD modes and frequencies resemble the cavity modes and resonance frequencies.

- It helps us demonstrate the effectiveness of our proposed DMD assisted FDTD scheme for decaying fields (pseudo-equilibrium).

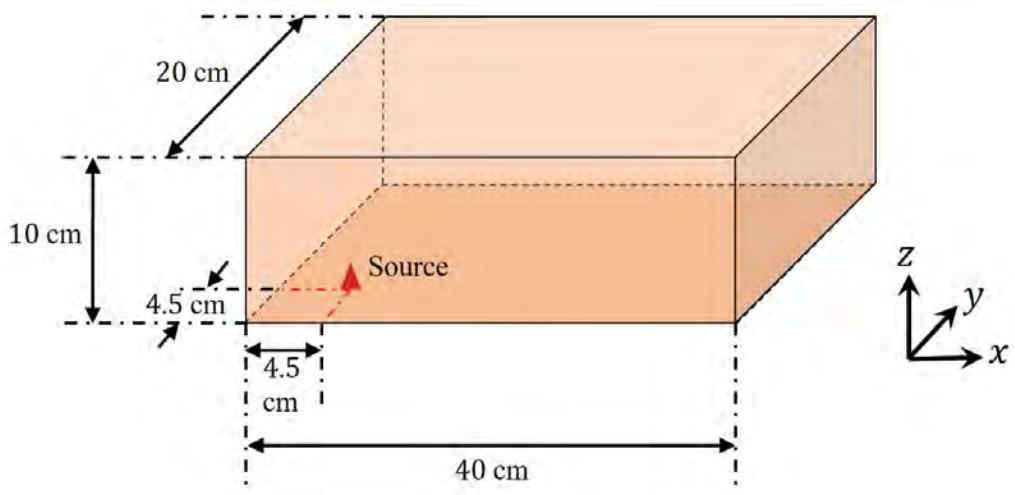


Figure 4.10: Closed rectangular cavity dimensions. Length is 40 cm, width is 20 cm and height is 10 cm. Source is located at the bottom, 4.5 cm far from each side of a corner.

- It addresses the issue of degenerate modes, a scenario where DMD modes do not necessarily represent the cavity modes. However, we will see that it does not affect DMD's extrapolation accuracy.

- It helps us showcase how we can extract $Q-$factors of individual modes by analyzing the DMD frequencies only.

On-the-Fly DMD

With $\Delta t_w = 102.80$ ns, $\Delta_t = 0.196$ ns, $d = 20$, and $t_{qry} = 409.22$ ns, equilibrium is detected at $t_f = 128.50$ ns. The final sliding DMD window spans from $t_0 = 25.70$ ns to $t_f = 128.50$ ns, which is used for analysis and time-extrapolation purpose.

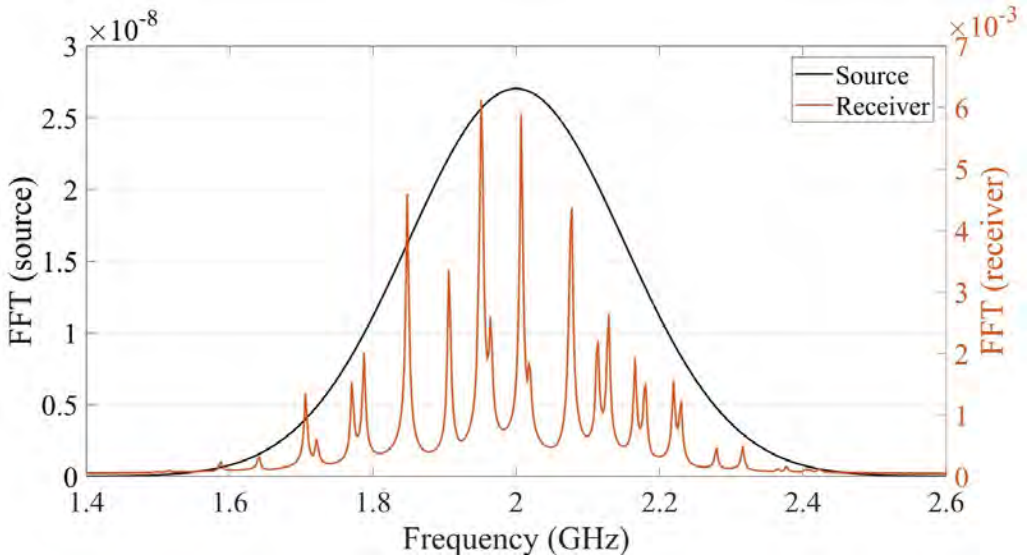


Figure 4.11: Frequency response for closed rectangular cavity.

Analysis

Most of the analysis is similar to the cylinder case, except the fact that in this case the dominant eigenvalues lie just inside the unit circle to capture decaying fields. The top five most energetic DMD modes are shown in Fig. 4.12. As expected, the DMD modes resemble the usual TM_{mnp} modes of a closed rectangular cavity, however with the exception of mode 1 (ϕ_1).

Degenerate modes: The ambiguity in shape of ϕ_1 arises from the degeneracy of some of the cavity modes. Since the dimensions of the cavity are integer multiples of each other, several cavity modes share common resonance frequencies. For example, TM_{341} and TM_{261} both have resonance frequency of 1.954 GHz (table 4.2). The

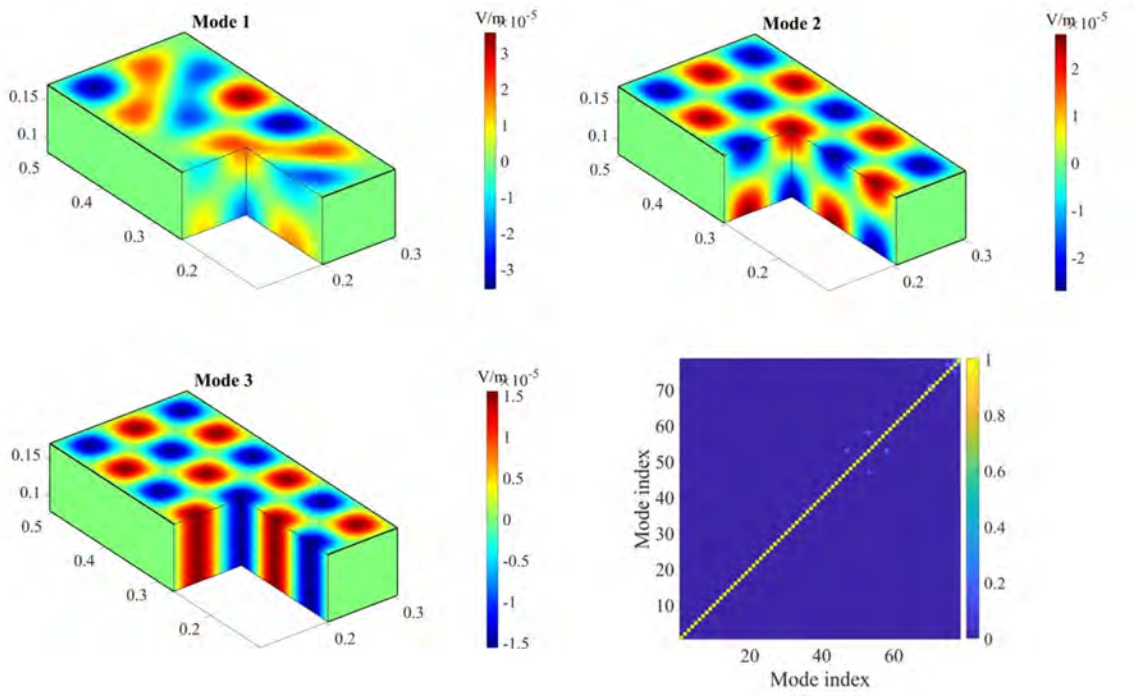


Figure 4.12: DMD modes and cross-correlation matrix (bottom right) for the setup in Fig. 4.10. First three most energetic DMD modes which resembles TM modes for a closed rectangular cavity. Mode 2 ($\boldsymbol{\phi}_2$) resembles TM_{351}, $\boldsymbol{\phi}_3$ TM_{360} and so on.

corresponding DMD mode $\boldsymbol{\phi}_1$ has the DMD frequency ($\widehat{\omega}_1$) of 1.952 GHz which very close to the theoretical value. However, the shape of $\boldsymbol{\phi}_1$ is neither of TM_{341} or TM_{261}, rather a linear combination of both the patterns. Let us consider i^{th} and j^{th} modes in (4.23) to be degenerate with common frequency $\tilde{\omega}_{ij}$. We can combine these terms together to get a new mode shape $(\mathbf{a}_i + \mathbf{a}_j)$ associated with $e^{\tilde{\omega}_{ij}t}$. Since DMD ensures temporal orthogonality, it extracts $(\mathbf{a}_i + \mathbf{a}_j)$ as the spatial pattern associated with

Table 4.2: Comparison between FDTD, DMD and theoretical resonance frequencies. The frequency of i^{th} DMD mode is denoted by f_i.

$\mathbf{f}_{\text{FDTD}}$ **(GHz)**	$\mathbf{f}_{\text{DMD}}$ **(GHz)**	$\mathbf{f}_{\text{theo}}$ **(GHz)**
1.951	1.952 (f_1)	1.954 ($\text{TM}_{341,261}$)
2.078	2.077 (f_2)	2.080 (TM_{351})
2.007	2.007 (f_3)	2.011 (TM_{360})
1.906	1.906 (f_4)	1.911 (TM_{270})
1.848	1.848 (f_5)	1.851 (TM_{350})

$\tilde{\omega}_{ij}$. This claim is further strengthened by the zero (or near zero) correlation (Fig. 4.12) of $\boldsymbol{\phi}_1$ with other DMD modes, suggesting that $\boldsymbol{\phi}_1$ must be linear combination of cavity modes which are mutually orthogonal in nature. Note that TM_{351} and TM_{132} are also degenerate in nature with common resonance frequency of 2.080 GHz (Table 4.2). However, DMD does not mix those modes since they are not exactly same. The Yee's FDTD implementation considers staggered grid points for $\mathbf{e}_z$ in the $z-$ direction, resulting in slight mismatch when the index "p" in TM_{mnp} differs for any set of degenerate (theoretically) modes. Similar observations can be made for other modes as well. Note that the theoretical values in 4.2 were calculated without taking into consideration the lossy material. As a result, they are slightly shifted from f_{FDTD} and f_{DMD}.

Q−factor extraction: Electric fields inside the cavity in a source free region can be described by (4.34). The $Q-$factor for m^{th} mode is given as, $Q = |\omega_{m,R}/(2\alpha_m)|$. With respect to (4.23), a more familiar expression for Q-factor would be $Q = |\text{Re}\{\omega_m\}/2\text{Im}\{\omega_m\}| = |\text{Im}\{\tilde{\omega}_m\}/2\text{Re}\{\tilde{\omega}_m\}|$. Since the DMD frequencies are also complex in nature and approximate the resonance frequencies $\tilde{\omega}$, we can write the $Q-$factor in terms of real and imaginary parts of the DMD frequency $\widehat{\omega}$. The $Q-$factor for m^{th} cavity mode which corresponds to the $\widehat{m}^{\text{th}}$ DMD mode, can be written as

$$Q_m^{\text{theo}} = \left|\frac{\text{Re}\{\omega_m\}}{2\text{Im}\{\omega_m\}}\right| = \left|\frac{\text{Im}\{\tilde{\omega}_m\}}{2\text{Re}\{\tilde{\omega}_m\}}\right| \approx \left|\frac{\text{Im}\{\widehat{\omega}_{\widehat{m}}\}}{2\text{Re}\{\widehat{\omega}_{\widehat{m}}\}}\right| = Q_{\widehat{m}}^{\text{DMD}} \tag{4.37}$$

Table 4.3: Comparison between theoretical $Q-$factor and $Q-$factor derived from DMD frequencies.

	$\text{TM}_{\mathbf{341}}$	$\text{TM}_{\mathbf{351}}$	$\text{TM}_{\mathbf{360}}$	$\text{TM}_{\mathbf{270}}$	$\text{TM}_{\mathbf{350}}$
$\mathbf{Q}^{\text{theo}}$	543.64	578.50	559.40	531.51	514.90
$\mathbf{Q}^{\text{DMD}}$	542.95	577.73	558.37	530.34	514.08

The $Q-$factor derived from DMD frequencies matches very closely with the theoretical ones (Table 4.3).

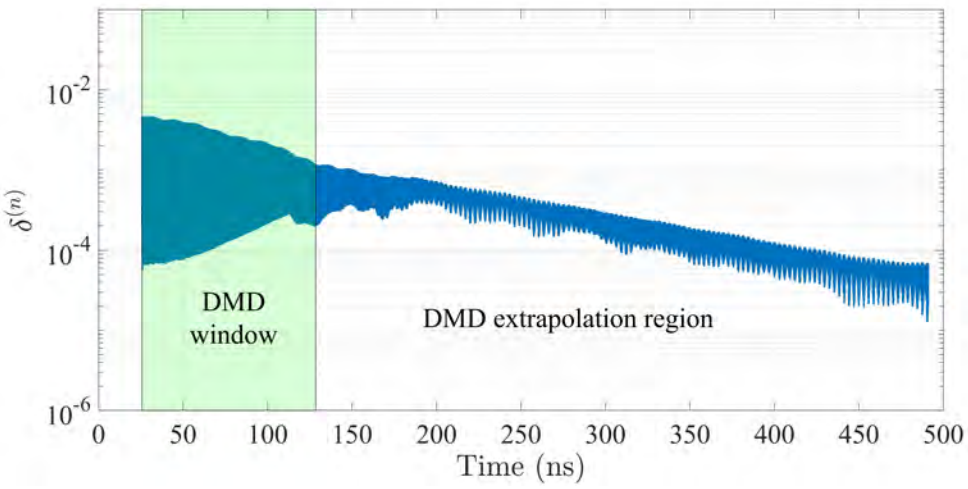


Figure 4.13: DMD reconstruction error for closed rectangular cavity.

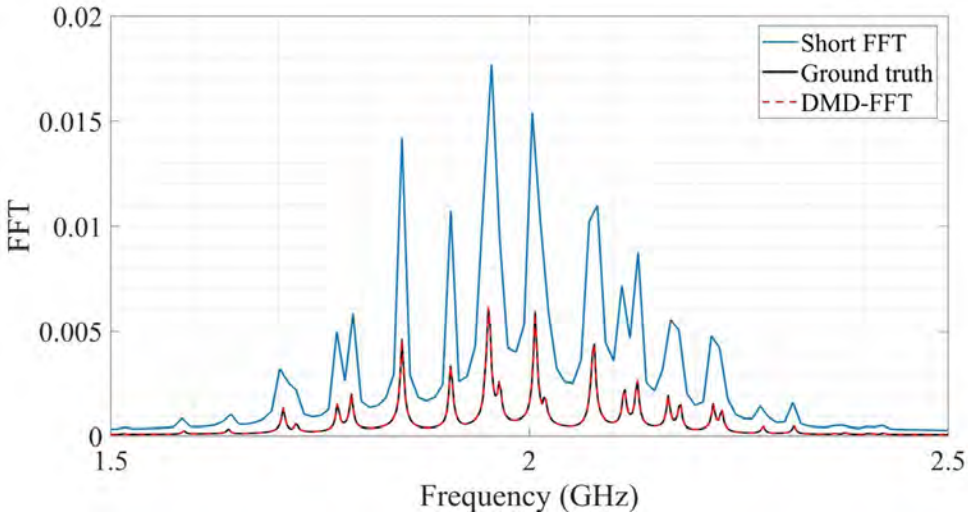


Figure 4.14: FFT with DMD extrapolated fields for closed rectangular cavity.

Time-Extrapolation

The relative error in the extrapolation region is much below 1% (Fig. 4.13), indicating DMD modes are able to capture the dynamics accurately. This is also reflected in the accurate reconstruction of the high-resolution FFT plot in Fig. 4.14.

4.4.3 L-Shaped Cavity

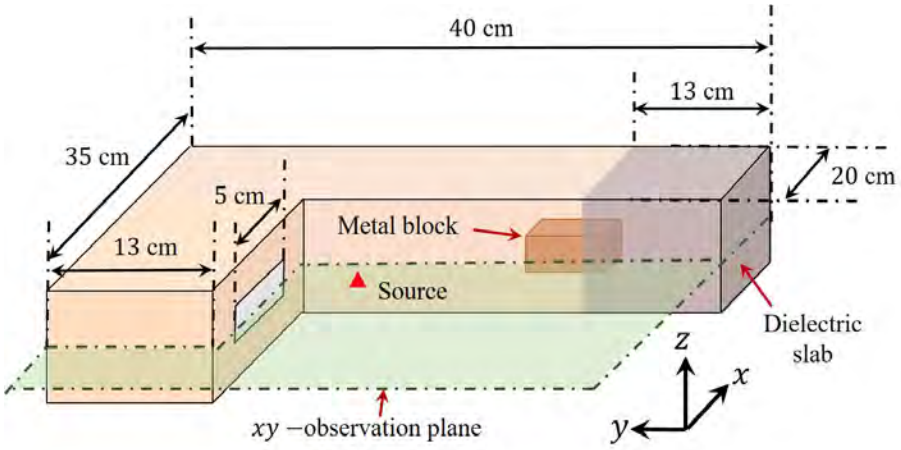


Figure 4.15: L-shaped cavity dimensions.

Next we consider a more complicated and practical test case scenario of a L-shaped cavity with slot. In order to make it more challenging, we put a metal object inside it with dielectric loading, with the aim of modeling $\mathbf{e}_z$ on the xy-plane (horizontal plane) passing through the slot, metal object and the dielectric. The whole setup is illustrated in Fig. 4.15. The width of the slot is 4 cm and the center

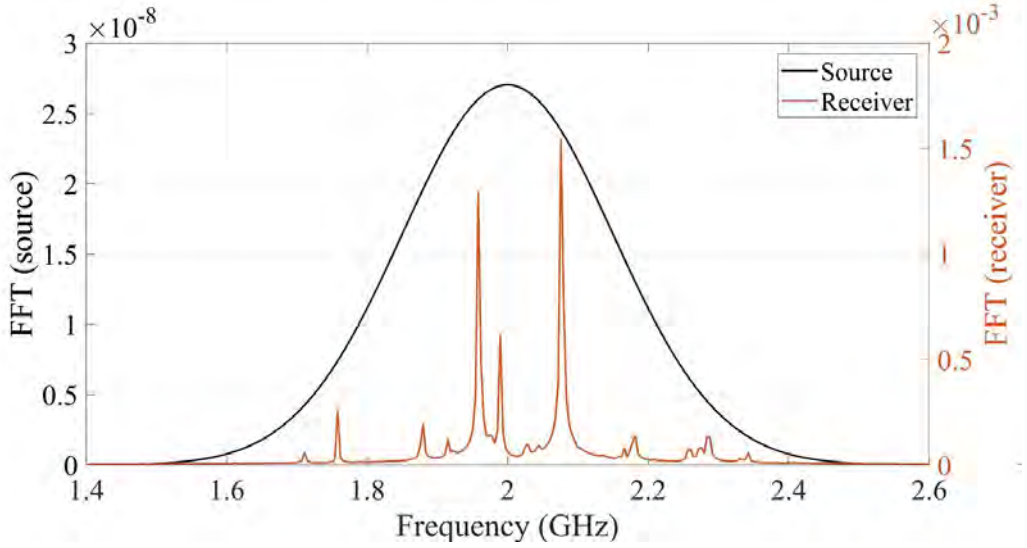


Figure 4.16: Frequency response for L-shaped cavity.

of the slot is located 8.5 cm away from the far end of the shorter arm and 6.5 cm above from the bottom of the cavity. The metal block has dimension of 3.5 cm $\times$2.5 cm $\times$1.5 cm along x, y and $z-$direction respectively and located just after the dielectric block ($\epsilon_{r,block} = 2.5$) along $y-$ direction. The center of the block is located 4.5 cm above the cavity bottom and 8.5 cm from the side wall (closer to the slot) of the long arm. Both the metal block and the cavity walls are made of aluminum. The cavity is excited with a $z-$directed current source from outside the cavity. The fields couple through the slot and excites cavity modes. The simulation is run for 6×10^4 time steps ($t = 491.08$ ns), and the frequency response is shown in Fig. 4.16. This example is particularly challenging for following reasons,

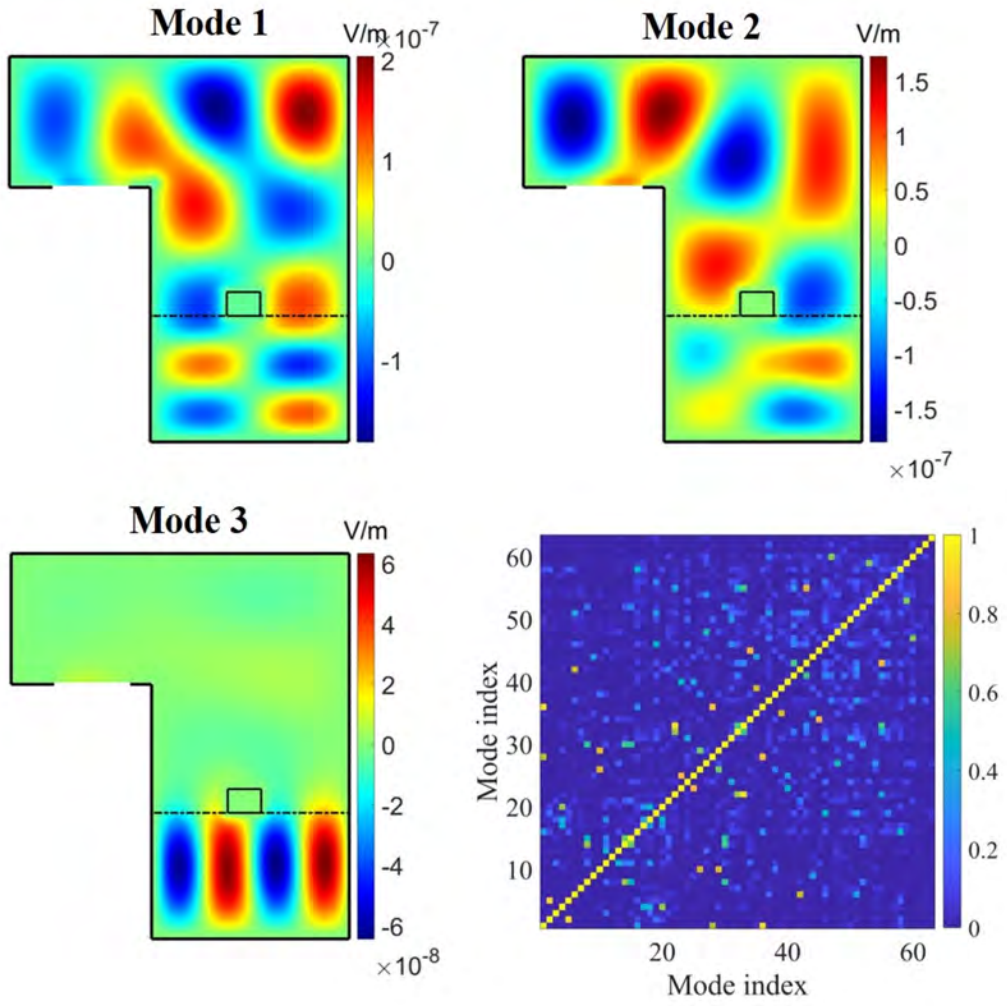


Figure 4.17: First three most energetic DMD modes and spatial correlation matrix for the setup in Fig. 4.15. The boundary between the dielectric block and rest of the cavity is shown by black dotted line, whereas the metal block is denoted by a rectangle.

- The open slot radiates, resulting in more continuous frequency response, which is not necessarily favourable for DMD, since it attempts to model the time evolution with finite number of discrete frequencies (DMD frequencies).

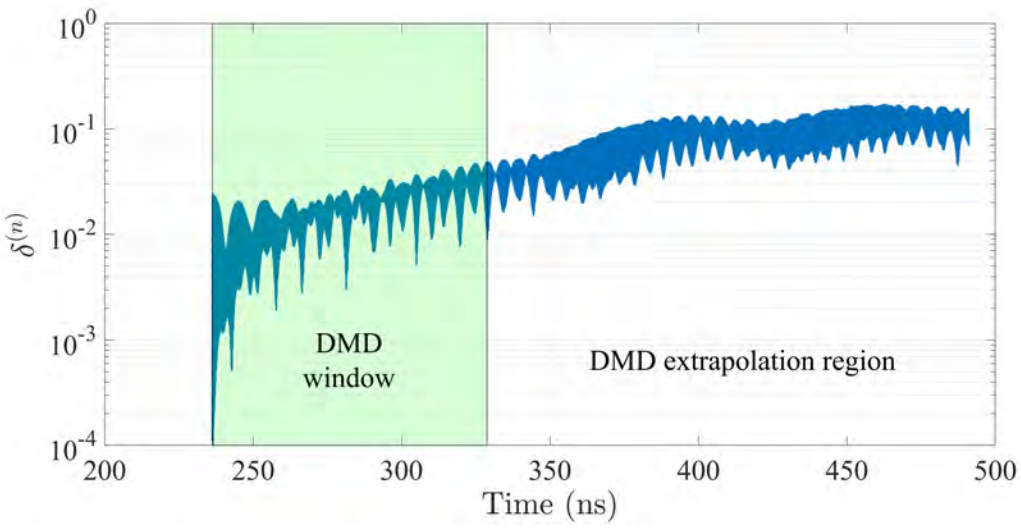


Figure 4.18: DMD reconstruction error.

- Multiscale nature of the dynamics due to presence of a dielectric block, and metal object makes the modeling task challenging.
- The fields are observed only on a cross-section of the cavity. Modeling partial state using a linear model such as DMD poses certain challenges which were discussed in sec. 4.2.2.

On-the-Fly DMD

With $\Delta t_w = 92.52$ ns, $\Delta_t = 0.213$ ns, $d = 20$, and $t_{qry} = 409.24$ ns, equilibrium is detected at $t_f = 328.78$ ns. The final sliding DMD window spans from $t_0 = 236.26$ ns to $t_f = 328.78$ ns, which is used for analysis and time-extrapolation purpose. Note

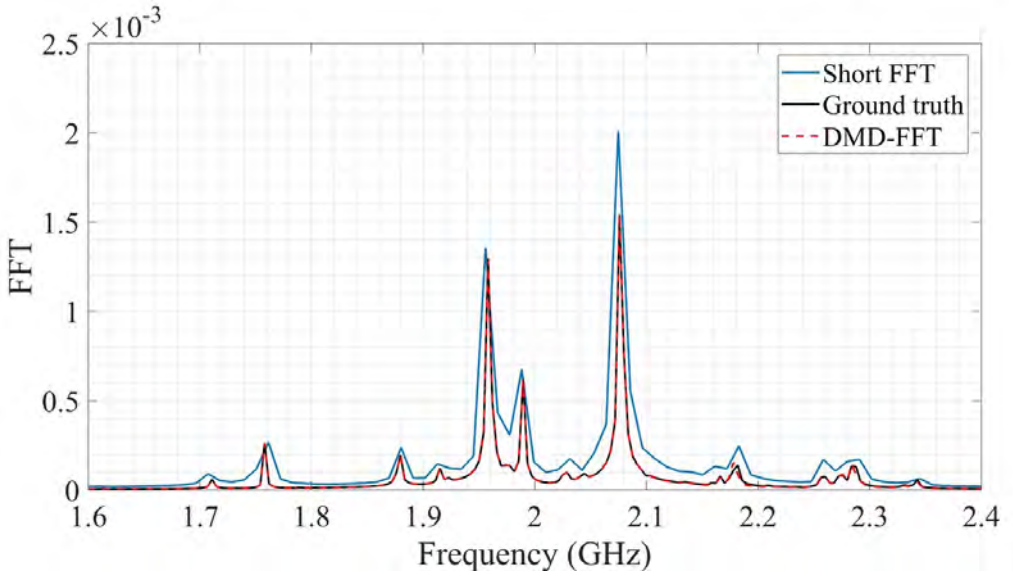


Figure 4.19: FFT with DMD extrapolated fields.

that due to complex geometry and time taken by the cavity modes to settle in, the pseudo-equilibrium is detected much later compared to the earlier test cases.

Analysis and Time-Extrapolation

Fig .4.17 shows the first three most energetic DMD modes and spatial correlation among different DMD modes. We observe that DMD modes are able to capture the multiscale phenomena, especially mode 3 (ϕ_3) which captures the oscillations inside the dielectric block. However, correlation matrix has several non-zero off-diagonal elements, indicating not all the DMD modes corresponds to the cavity modes. However, this is expected in the presence of a radiating slot. Total 63 DMD modes are utilized to approximate the dynamics.

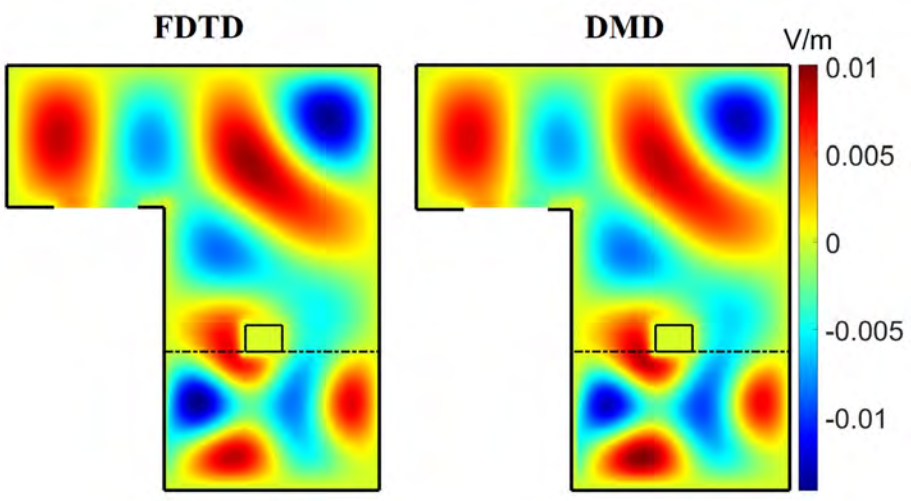


Figure 4.20: FDTD vs DMD prediction of $\mathbf{e}_z$ at $t = 486.17$ ns.

DMD's difficulty in modeling the dynamics is reflected in the extrapolation error in Fig. 4.18 which is around 8.83%, higher compared to previous test-cases. However, this is good enough for accurate reconstruction of the high-resolution FFT graph as can be seen in Fig. 4.19. The predicted $\mathbf{e}_z$ pattern also matches closely with that from FDTD in the deep extrapolation region (Fig. 4.20).

4.4.4 Coaxial Cavity

We also test our method on data generated from commercial software like CST Studio Suite. We simulate a coaxial cavity resonator (4.21) which is typically used in filter design. The cavity is excited with a z-directed current source spanning from top to bottom of the cavity. The excitation signal is a Gaussian pulse spanning 4

GHz - 10 GHz frequency band. The walls of the cavity are 1 mm thick and have conductivity of $\sigma = 37.7 \times 10^6$ S/m. Inside of the cavity is filled with vacuum/air. The simulation is run until 60 ns in CST, and fields are sampled at every 0.01 ns. We are interested in modeling the z-directed electric field on the xy plane 5 mm above the bottom of the cavity.

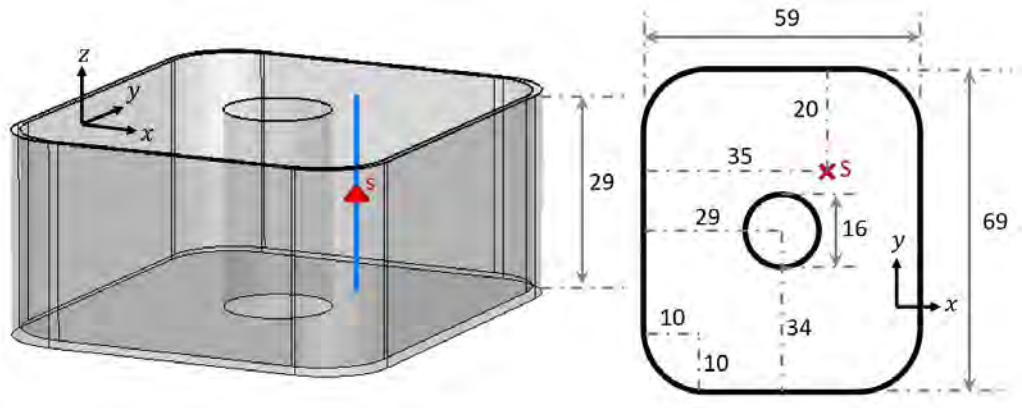


Figure 4.21: Dimensions of the coaxial cavity geometry. The figure on the right shows the top view. All the units are in mm. The z-directed current source (blue line (left), red cross (right)) spans from the top to bottom of the cavity.

With $\Delta t_w = 13.55$ ns, $\Delta_t = 0.04$ ns, $d = 20$, and $t_{qry} = 60$ ns, equilibrium is detected at $t_f = 16.95$ ns. A total of 74 DMD modes are used for time extrapolation beyond the final DMD window. The DMD modes and frequencies are compared with the resonant modes and frequencies obtained from CST's eigenmode solver. Comparison for first three most energetic DMD modes are provided in Fig 4.22. The DMD modes and frequencies are in good agreement with the eigenmodes and

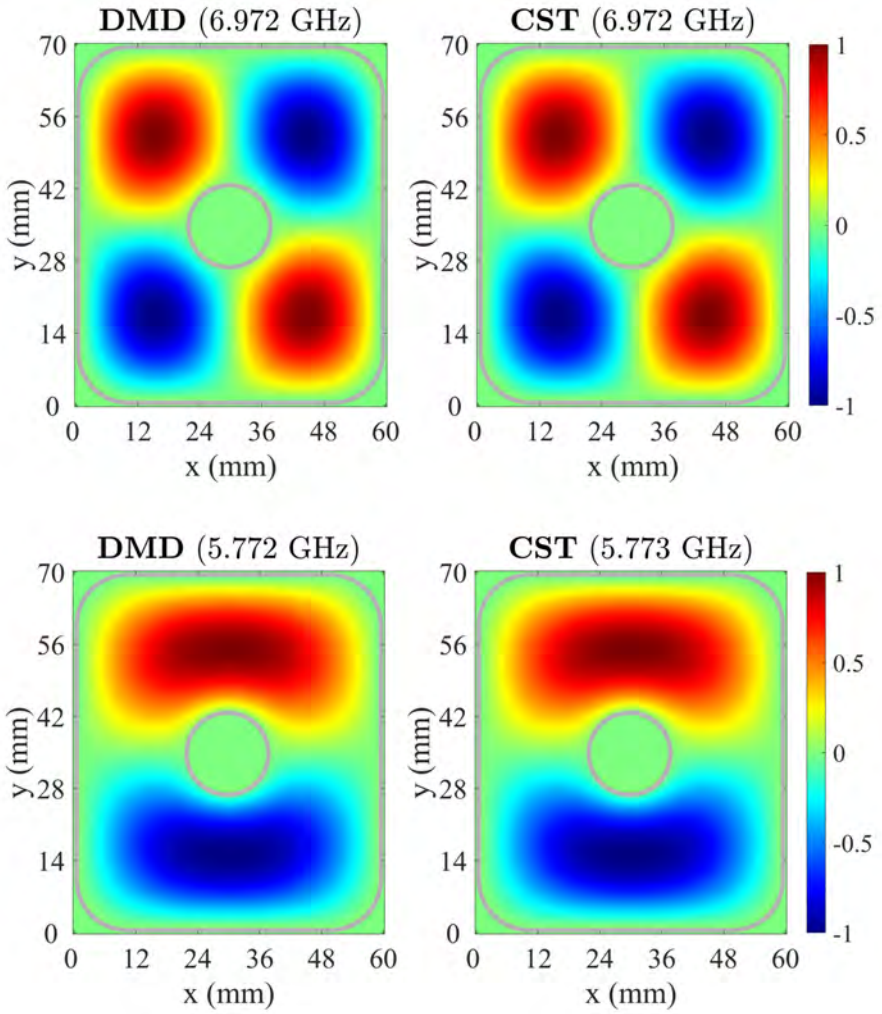


Figure 4.22: Comparison between cavity modes and frequencies obtained from DMD and CST. DMD mode 1 is shown at the top and mode 2 at bottom.

frequencies obtained from CST. Fig. 4.23 shows that the average extrapolation error is below 1%.

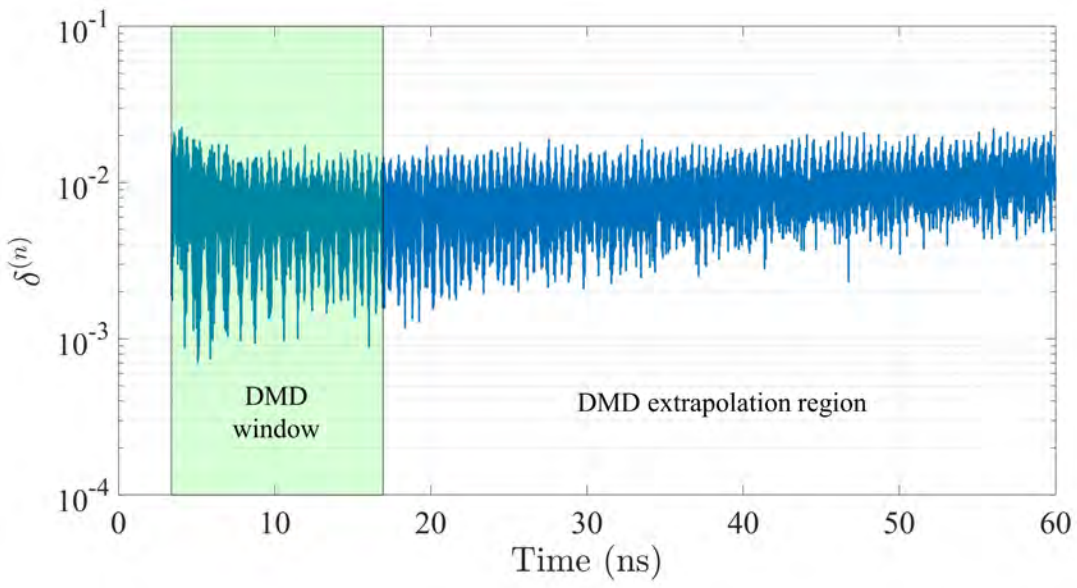


Figure 4.23: Relative 2-norm error in the extrapolation region for coaxial cavity.

4.4.5 Effect of d

One of the key parameters for Hankel-DMD is the number of stacks d. As mentioned earlier, there is no straight-forward way to select d. For a large number of training samples typically larger d results in better accuracy, but with higher computation cost. We study the effect of d on our proposed DMD-FDTD algorithm for L-shaped and coaxial cavity. First, we observe the effect of d on detection of pseudo-equilibrium. Second, we keep the window location and other parameters fixed and vary d to observe the variation in average extrapolation error $\langle\delta\rangle_{\text{ext}}$ ($\delta^{(n)}$ averaged

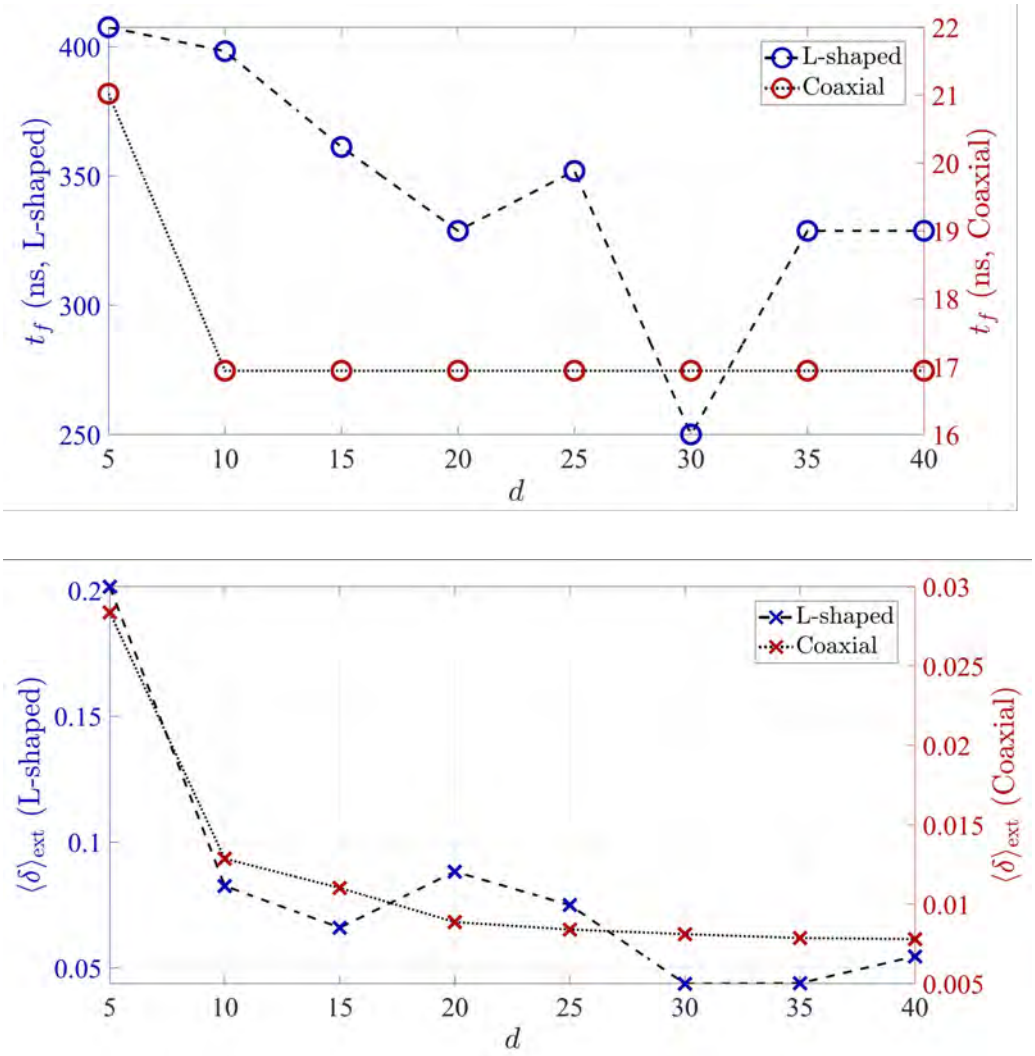


Figure 4.24: Effect of different values of d on (a) pseudo-equilibrium detection (t_f), and (b) average extrapolation error $\langle\delta\rangle_{\text{ext}}$.

over all n beyond the DMD window). Since DMD is typically able to model the dynamics better with large d, we expect to see convergence in t_f as well with $\langle\delta\rangle_{\text{ext}}$ (Fig.

4.24). Convergence characteristics for the L-shaped cavity are not ideal, primarily because of the modeling difficulty due to several reasons as mentioned in Section 4.4.3.

4.5 Computational Gain, Stability and Error Bounds

The computational complexity of FDTD/FETD algorithm for each time step is $\mathcal{O}(N)$, N being the aggregate mesh dimension, i.e. the number of mesh elements. For query at $n = n_{qry}$, total computational complexity is given by $\mathcal{O}(n_{qry}N)$.

The computation cost of DMD is dominated by the SVD step (2.19). For $(l+1)$ DMD snapshots, and d stacks, the runtime complexity of DMD is $\mathcal{O}(ld^2N^2)$. Note that for our proposed DMD assisted FDTD/FETD scheme, we need to perform DMD several times due to the on-the-fly nature of our algorithm. Since the equilibrium or pseudo-equilibrium is detected at n_f, number of times DMD is performed is $\frac{n_f - n_w + 1}{\Delta_n}$. For conciseness, we consider the upper bound $\frac{n_f}{\Delta_n}$ as the number of times we need to perform DMD for online or on-the-fly application. The computation complexity for on-the-fly DMD is then given by $\mathcal{O}(\frac{ld^2N^2n_f}{\Delta_n})$. Comparing it with FDTD time complexity of $\mathcal{O}(n_{qry}N)$, we can conclude that the DMD-FDTD or DMD-FETD scheme will be useful if $n_{qry} \gg n_f$. Note that the computation cost for on-the-fly DMD can be further reducd using incremental approaches [70]. The gain in runtime is summarized in Table 4.4 when $n_{qry} = 5 \times 10^5$ for cylindrical cavity and $n_{qry} = 6 \times 10^4$ for rectangular and L-shaped cavity. For the coaxial example, we choose $n_{qry} = 6001$. The DMD-FDTD runtime is derived by calculating the time FDTD would take if it

Table 4.4: Computational gain in runtime using DMD-FDTD

Test-cases	**FDTD runtime (mins)**	**DMD-FDTD runtime (mins)**
Cylindrical cavity	1959	347
Rectangular cavity	224	64
L-shaped cavity	435	301
Coaxial cavity (CST)	4	1.47

was stopped at n_f, and then adding the approximate DMD runtime in MATLAB. Note that the computational gain depends on the time step up to which it is desired to extrapolate. The gain in runtime is not large for L-shaped cavity 4.4, because the system takes longer time to attain pseudo equilibrium compared to the other cases. As a result, raising the FDTD termination flag is delayed. With $n_{qry} \gg n_f$, the numbers in Table 4.4 will look more favourable towards DMD-FDTD, as long as the long-term predictions satisfy a certain error criterion.

The stability of the DMD model depends on the relative position of DMD eigenvalues ($\lambda_i,\ i = 1, 2, \ldots, r$) compared to the unit circle on the complex plane. The eigenvalues residing outside the unit circle essentially translate to positive real part for $\widehat{\omega}$ in (2.23), resulting in an exponentially growing solution with time. However, for our application, we force the λ_i s residing outside the unit circle radially inwards

to lie exactly on the unit circle. As a result, our method does not suffer from instability. In other words, our proposed DMD-FDTD/FETD scheme is unconditionally stable.

In order to apply any data-driven extrapolation method for practical systems, it is useful to have prior insights regarding the error bounds. Although authors in [131] provided error bounds for DMD extrapolation in the context of parabolic partial differential equations (PDEs), to the best of the authors' knowledge, error bounds for general dynamical systems or hyperbolic PDEs are still undefined. However, DMD is not entirely blackbox and users can look into certain parameters to get some idea regarding the extrapolation accuracy of DMD. Typically, the extrapolation error is higher than the interpolation error (DMD reconstruction error inside DMD window)[3]. So, if the interpolation error is higher than acceptable error limit (defined by user), it is not wise to use DMD for extrapolation. A similar conclusion can be made by observing the term $\langle \delta^{(n)} \rangle_w$ as well. For cavity resonances, the spatial correlation matrix can give some idea whether DMD is able to extract the orthogonal cavity modes or not. A "cleaner" (i.e. zero or near zero off-diagonal elements) correlation matrix for the dominant modes is generally a good sign that DMD is able to model the dynamics well.

[3]The extrapolation error in Fig. 4.13 goes down in the extrapolation region because of the decaying field amplitudes. Typically as we go deep into the extrapolation region, the difference between the ground truth and DMD prediction increases. But due to the decaying fields, as an overall effect, the difference between ground truth and DMD decreases. Since the denominator in the error expression is fixed, the overall relative error decreases.

4.6 Summary

In this chapter, we discussed how a linear Koopman Reduced-Order Model (ROM) such as DMD can effectively model electromagnetic cavity resonances, which are inherently linear. Using a linear ROM to model a linear system allows DMD to be interpreted from a state-space modeling perspective for EM cavity problems, rather than solely from a Koopman perspective. We showcased how Hankel DMD effectively extracts physical cavity modes, corresponding resonance frequencies, and individual Q-factors from only partial state observations. We also addressed scenarios where the extracted DMD modes might not accurately reflect the actual cavity modes. We demonstrated how DMD can accelerate FDTD/FETD-based simulations and indirectly facilitate frequency domain analysis.

Chapter 5: Modeling Self-Fields in Particle-in-Cell Plasma Simulations using Dynamic Mode Decomposition

In this chapter, we will first discuss the details of EMPIC algorithm which is used to generate the high-fidelity data. Then we will explore the effectiveness of DMD in modeling the self-fields in EMPIC simulations and the effect of DMD predicted fields on the particle phase-space. In order to minimize the computational cost, ideally one would like to perform reduced-order modeling such as DMD using data from high-fidelity simulations based on relatively short time windows and extrapolate the results in future time. However, as is shown in this chapter, accurate prediction of the equilibrium dynamics using data-driven methods such as DMD requires sufficient data harvesting near equilibrium. As a result, a related important question to be addressed is how to leverage DMD to optimally predict the equilibrium state. In this chapter, We will use an on-the-fly DMD algorithm discussed in Chapter 3 for real-time termination of EMPIC simulations. This chapter is based on [47, 49].

5.1 EMPIC Algorithm

5.1.1 Overview

EMPIC algorithms have been widely used for collisionless plasma problems for their ease of implementation and their ability to accurately capture transient and nonlinear effects. There are many EMPIC algorithm variants [132, 133, 134, 135, 136, 15]. However, at their core, any EMPIC algorithm implements a marching-on-time evolution wherein, at each time step, four stages are involved (Fig. 5.1): (*i*) field-update, (*ii*) gather, (*iii*) pusher, and (*iv*) scatter. These four stages are executed in a cyclic fashion for every time step.

Within a EMPIC algorithm, the field-update solution of Maxwell's equations can be obtained by conventional partial-differential equation (PDE) solvers such as finite differences or finite elements. The former is typically used in conjunction with structured (regular) meshes and the latter with unstructured (irregular) meshes. The fields **E** and **B** and the current density **J** are defined on mesh elements (element edges and facets) whereas the (super)particles are defined on ambient space (i.e. they can move continuously anywhere across the various finite difference or finite element cells comprising the solution domain). This is where the "particle-in-cell" terminology comes from [137].

Before delving into a more detailed discussion of the particular EMPIC algorithm considered here, it is important to discuss the type of Maxwell field discretization

scheme that we will employ. Depending on the chosen field discretization, the details of the equations governing the four cyclic steps may vary.

The spatial domain of interest can be discretized by a structured (regular) mesh or by an unstructured (irregular) mesh. A common example of a structured mesh is a Cartesian one where each mesh element is a rectangle with identical dimensions. On the other hand, unstructured meshes are typically comprise triangular elements in two dimensions or tetrahedral elements in three dimensions. The shape of each triangle or tetrahedron can vary so as to conform to the specific problem geometry. In general, structured meshes have certain advantages over unstructured meshes such as simplified bookkeeping of the mesh elements and the local solution. However, unstructured meshes can better capture complex geometries and are free from staircase errors. Unstructured meshes can also provide better grid (mesh) dispersion performance in plasma problems and mitigate spurious numerical artifacts such as artificial Cherenkov radiation [138].

One of the critical aspects related to the spatial discretization of EMPIC algorithms based on unstructured meshes is charge conservation [139, 140, 141]. On the finite element mesh, the macroscopic charge density ρ_q is a variable based on mesh nodes and the macroscopic current density $\mathbf{J}$ is a variable based on mesh element edges. Such an assignment choice is based on first-principle rules set by discrete exterior calculus, as discussed in [142]. Both these macroscopic quantities need to be computed numerically from the instantaneous positions and velocities of the charged

(super)particles in ambient space. However, unless this calculation is performed in a consistent manner, charge conservation can be violated. Indeed, many attempts at developing unstructured mesh EMPIC algorithms over the years have led to violation of conservation laws and other spurious effects. Charge conservation has been a especially challenging problem for unstructured mesh EMPIC algorithms because of irregular connectivity among the mesh elements. Ad hoc approaches such as pseudo-currents [140] or Poisson correction steps [139] have been utilized in order to enforce charge conservation a posteriori; however, these approaches either subtly alter the physics of the problem or call for an additional, time-consuming linear solver at each time step. Recently, the problem of charge conservation was successfully resolved from first-principles by employing Whitney forms as the correct finite-element basis functions to both represent the fields and macroscopic charges/currents as mesh variables, and to project the effects of the movement of the ambient point charges back onto the mesh variables during the scatter procedure [132]. Other variants of first-principle conservation strategies in EMPIC algorithms have been considered in [135, 141, 15]. In this chapter, we will follow references [132, 143, 144] and discuss the implementation of a finite-element-based charge-conservation EMPIC algorithm on unstructured meshes from first principles with a matrix-free field-update scheme. This implementation is used to generate the full-order numerical results presented later on in this Chapter. The key feature of Whitney forms is that they represent consistent interpolants for discrete differential forms of various degrees representing

the macroscopic variables ρ_q, $\mathbf{J}$, $\mathbf{E}$, and $\mathbf{B}$ on the mesh. In the language of the exterior calculus of differential forms, the electric field intensity $\mathbf{E}$ is considered as a differential form of degree one or 1-form, the magnetic flux density $\mathbf{B}$ as a 2-form, and the (Hodge dual of the) current density $\mathbf{J}$ as a 1-form, and the charge density ρ_q as a 0-form. A detailed discussion on discrete exterior calculus of differential forms is out of scope for this chapter. The interested reader is referred to [145, 132, 146, 142] for more details on this topic. We will next discuss a few important aspects which are pertinent to explain the four cyclic procedures in the EMPIC algorithm and later on to their relation to the reduced-order approaches.

In a nutshell, the present charge-conserving EMPIC algorithm on unstructured meshes has been developed based on first principles. In each time step, the electric field intensity values $\mathbf{E}$ (defined on element edge), and the magnetic flux density values $\mathbf{B}$ (defined on element facet) are solved using a finite element time-domain algorithm with a spatial discretization based on discrete exterior calculus approach as mentioned above. The discrete time-update is based on a second-order leap-frog algorithm ([143, 144]). The $\mathbf{E}$ and $\mathbf{B}$ fields are then interpolated at each particle position (gather) using Whitney forms, and used to update the position and velocity of each particle via Lorentz force law and the kinematic equations of motion (pusher). Finally, the motion of charged particles is mapped back to the mesh variables representing the macroscopic current $\mathbf{J}$ and charge ρ_q densities (scatter). These steps

are repeated again in the next time step. The details of each of these steps are as follows:

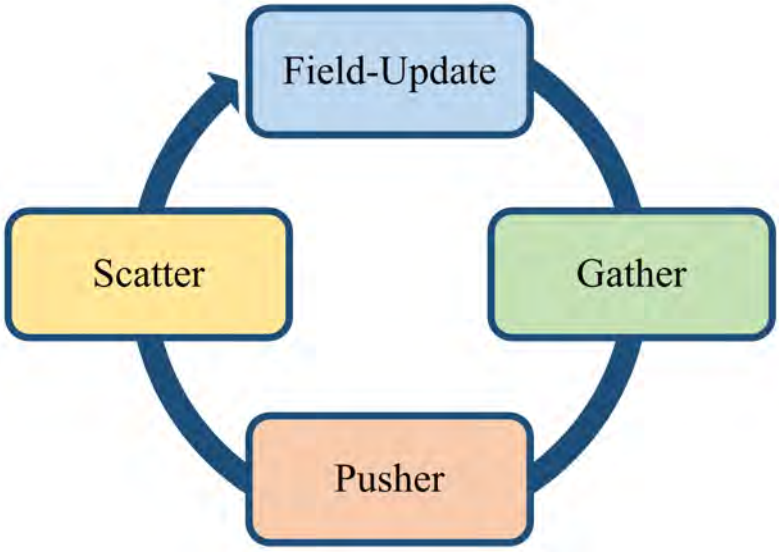


Figure 5.1: Cyclic stages of the EMPIC algorithm at each time step.

5.1.2 Field Update Stage

In the present finite element setting, the electric field intensity $\mathbf{E}^{(n)}(\mathbf{r})$ and the magnetic flux density $\mathbf{B}^{(n+\frac{1}{2})}(\mathbf{r})$ at the n^{th} (half-integer time steps for $\mathbf{B}$) time step can be represented using vector proxies of the Whitney forms [147, 148, 149, 133] as

$$\mathbf{E}^{(n)}(\mathbf{r}) = \sum_{i=1}^{N_1} e_i^{(n)} \mathbf{W}_i^1(\mathbf{r}), \tag{5.1a}$$

$$\mathbf{B}^{(n+\frac{1}{2})}(\mathbf{r}) = \sum_{i=1}^{N_2} b_i^{(n+\frac{1}{2})} \mathbf{W}_i^2(\mathbf{r}), \tag{5.1b}$$

where N_1 and N_2 denote, respectively, the total number mesh element edges and mesh element facets. The basis functions $\mathbf{W}_i^1(\mathbf{r})$ and $\mathbf{W}_i^2(\mathbf{r})$ are the vector proxies of Whitney 1- and 2-forms [147, 148]. These functions have a one-to-one correspondence with the element edges and element facets of the mesh. The e_i and b_i coefficients are the degrees of freedom (DoF) for the discretized $\mathbf{E}$ and $\mathbf{B}$ fields, respectively. For ease of discussion, with mild abuse of language, we will refer to the above vector proxies of Whitney forms simply as Whitney forms. Note that for a two-dimensional (2-D) unstructured mesh composed of triangular elements, the mesh element facets are simply the triangular elements themselves. One can similarly represent the electric current density and the charge density on the mesh as

$$\mathbf{J}(\mathbf{r})^{(n+\frac{1}{2})} = \sum_{i=1}^{N_1} j_i^{(n+\frac{1}{2})} \mathbf{W}_i^1(\mathbf{r}), \tag{5.2a}$$

$$Q(\mathbf{r})^{(n)} = \sum_{i=1}^{N_0} q_i^{(n)} \mathbf{W}_i^0(\mathbf{r}), \tag{5.2b}$$

where N_0 is the total number of nodes on the mesh and $\mathbf{W}_i^0$ is a set of Whitney 0-forms having one-to-one correspondence with the mesh nodes, with j_i and q_i being the DoFs for current density and charge density, respectively. From the above, it is clear that the Whitney forms in (5.1) and (5.2) acts as basis functions in the usual finite element context, and the unknown DoFs are simply the weights corresponding to each basis function. Using the leap-frog discretization for the time derivatives to construct the time update scheme, the following discrete Maxwell's equations are

obtained [132, 143, 133],

$$\mathbf{b}^{(n+\frac{1}{2})} = \mathbf{b}^{(n-\frac{1}{2})} - \Delta t \mathbf{C} \cdot \mathbf{e}^{(n)}, \tag{5.3a}$$

$$\mathbf{e}^{(n+1)} = \mathbf{e}^{(n)} + \Delta t \left[\star_{\epsilon}\right]^{-1} \cdot \left(\tilde{\mathbf{C}} \cdot \left[\star_{\mu^{-1}}\right] \cdot \mathbf{b}^{(n+\frac{1}{2})} - \mathbf{j}^{(n+\frac{1}{2})}\right), \tag{5.3b}$$

with constraints

$$\mathbf{S} \cdot \mathbf{b}^{(n+\frac{1}{2})} = 0, \tag{5.4a}$$

$$\tilde{\mathbf{S}} \cdot \left[\star_{\epsilon}\right] \cdot \mathbf{e}^{(n)} = \mathbf{q}^{(n)}, \tag{5.4b}$$

where $\mathbf{e}$, $\mathbf{b}$, $\mathbf{j}$ and $\mathbf{q}$ represent column vectors of DoFs, i.e. $\mathbf{e} = [e_1\ e_2\ \ldots\ e_{N_1}]^{\mathrm{T}}$, $\mathbf{b} = [b_1\ b_2\ \ldots\ b_{N_2}]^{\mathrm{T}}$, $\mathbf{j} = [j_1\ j_2\ \ldots\ j_{N_1}]^{\mathrm{T}}$, and $\mathbf{q} = [q_1\ q_2\ \ldots\ q_{N_0}]^{\mathrm{T}}$, with $'\mathrm{T}'$ denoting the transpose. In addition, $\mathbf{C}$ and $\mathbf{S}$ refer to the incidence matrices encoding the discrete representation of the curl and divergence operators on the (primal) mesh while $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{S}}$ represent their counterparts on the dual mesh [146, 142, 150, 151]. These matrices are related through $\tilde{\mathbf{C}} = \mathbf{C}^T$ and $\tilde{\mathbf{S}} = \mathbf{S}^T$ [150, 151]. The incidence operators are metric-free in the sense that their elements are equal to $1, 0$ or -1, depending on the connectivity of mesh elements and their assigned relative orientation. Since, $\mathbf{E}$, $\mathbf{B}$ and $\mathbf{J}$ are directional in nature, e_i, b_i and j_i are signed scalar quantities. We should point out that $\mathbf{J}$ defined in (5.2a) and (5.3) is actually the Hodge dual of 2-form current density. In the language of differential forms, the current density is a 2-form defined on the dual mesh facets, where as the dual of current density is a 1-form defined on the primal mesh edges (from the Hodge duality, there is a one-to-one correspondence between dual mesh facets and primal mesh edges). However, to

facilitate the discussion, we will again abuse the language slightly and refer to $\mathbf{J}$ as the current density. Note that if the initial conditions satisfy the pair of equations (5.4) (which correspond to the divergence constraints $\nabla \cdot \mathbf{B} = 0$ and $\nabla \cdot \epsilon_0 \mathbf{E} = \rho_q$) then these equations are automatically satisfied [143] at all subsequent times steps. The metric information is encoded in the symmetric positive definite matrices $[\star_{\mu^{-1}}]$ and $[\star_\epsilon]$ in (5.3b), representing the discrete Hodge star operators [142, 133, 148]. One of the features of (5.3) is the explicit nature of the field update, which obviates the need for a linear solver at each time step. However, this comes at the one-time cost of calculating the approximate inverse $[\star_\epsilon]^{-1}$ in Eq. (5.3b). Instead of computing $[\star_\epsilon]^{-1}$ directly (which is computationally impractical for large problems), a sparse approximate inverse (SPAI) can be precomputed with tunable accuracy [143, 133].

5.1.3 Field Gather Stage

As explained before, the DoFs of the fields and currents are defined on the mesh; however, the particles can be anywhere in the solution domain. In order to model the effect of fields on the charged particles, it is necessary to interpolate the fields at the particle position. This is performed at the gather stage using the same Whitney forms as in Eqs. (5.1a) and (5.1b). The fields at p^{th} particle position $\mathbf{r}_p$ are simply given by

$$\mathbf{E}^{(n)}(\mathbf{r}_p) = \sum_{i=1}^{N_1} e_i^{(n)} \mathbf{W}_i^1(\mathbf{r}_p), \tag{5.5a}$$

$$\mathbf{B}^{(n+\frac{1}{2})}(\mathbf{r}_p) = \sum_{i=1}^{N_2} b_i^{(n+\frac{1}{2})} \mathbf{W}_i^2(\mathbf{r}_p), \tag{5.5b}$$

5.1.4 Particle Pusher Stage

Once the fields are interpolated at the particle position, the position and velocity of the particle can be updated using the Lorentz force equation and Newton's laws of motion, considering relativistic effects if necessary. The p^{th} particle (or superparticle) kinematic equations in the continuous time domain are given by

$$\gamma_p^2 = \frac{1}{1 - |\mathbf{v}_p|^2/c^2} \tag{5.6}$$

$$\frac{d\mathbf{r}_p}{dt} = \mathbf{v}_p = \frac{\mathbf{u}_p}{\gamma_p}, \tag{5.7}$$

$$\frac{d\mathbf{u}_p}{dt} = \frac{q_p}{m_0}\left[\mathbf{E}\left(\mathbf{r}_p, t\right) + \mathbf{v}_p \times \mathbf{B}\left(\mathbf{r}_p, t\right)\right], \tag{5.8}$$

where $\mathbf{r}_p$, $\mathbf{v}_p$ and q_p denotes the position, relativistic velocity and charge of p^{th} superparticle. The velocity of light and relativistic factor are given by c and γ_p respectively. Using a standard finite-difference approximation for the time derivatives, the discrete-time version of (5.7) and (5.8) becomes

$$\frac{\mathbf{r}_p^{(n+1)} - \mathbf{r}_p^{(n)}}{\Delta t} = \frac{\mathbf{u}_p^{(n+\frac{1}{2})}}{\gamma_p^{(n+\frac{1}{2})}}, \tag{5.9}$$

$$\frac{\mathbf{u}_p^{(n+\frac{1}{2})} - \mathbf{u}_p^{(n-\frac{1}{2})}}{\Delta t} = \frac{q_p}{m_0}\left(\mathbf{E}_p^{(n)} + \frac{\bar{\mathbf{u}}_p}{\bar{\gamma}_p} \times \mathbf{B}_p^{(n)}\right), \tag{5.10}$$

where $\bar{\mathbf{v}}_p$ denotes the mean particle velocity between the $(n \pm \frac{1}{2})^{\text{th}}$ time steps, and given the mean $\mathbf{u}_p$ and mean γ_p, we set $\bar{\mathbf{u}}_p = \bar{\gamma}_p \bar{\mathbf{v}}_p$. The mean velocity for non-relativistic case ($\gamma_p \to 1$) is simply $\bar{\mathbf{v}}_p^n = \mathbf{v}_p^n = \left(\mathbf{v}_p^{n+\frac{1}{2}} + \mathbf{v}_p^{n-\frac{1}{2}}\right)/2$. In the relativistic regime, however, the mean velocity $\bar{\mathbf{v}}_p$ should be calculated carefully. The reader is referred to [144] for more details on the different types of relativistic pushers.

5.1.5 Current and Charge Scatter Stage

In the scatter stage, the effect of the collective movement of charged particles are mapped back to currents defined on the element edges of the mesh, and mapped to charges defined on the mesh nodes. This mapping is performed using the Whitney forms to ensure consistency, and in particular, charge conservation on the mesh. The charge density is assigned to the mesh nodes from the instantaneous positions of the charged particles using Whitney 0-forms, whereas Whitney 1-forms are used to map current density to mesh element edges from the trajectory traversed by charged particles during a single time step [132, 143].

The total charge assigned to the i^{th} node at the n^{th} time step is calculated as follows

$$q_i^{(n)} = \sum_p q_p \mathbf{W}_i^0(\mathbf{r}_p^{(n)}) = \sum_p q_p \widehat{\lambda}_i(\mathbf{r}_p^{(n)}), \tag{5.11}$$

where the summation index p runs over all the N_p superparticles, q_p is the charge and $\mathbf{r}_p^{(n)}$ is the position of the p^{th} superparticle at the n^{th} time step, $\mathbf{W}_i^0$ represents the Whitney 0-form or equivalently, the barycentric coordinate $\widehat{\lambda}_i(\mathbf{r}_p)$ of the point $\mathbf{r}_p$ with respect to the i^{th} node. Similarly, the current density generated at mesh element edges due to movement of charged particles are calculated using Whitney 1-forms. Let us consider the p^{th} (super)particle moving from $\mathbf{r}_p^{(n)}$ to $\mathbf{r}_p^{(n+1)}$ during one time step (comprising Δt time duration). The current density $j_{ij,p}^{(n+\frac{1}{2})}$ generated

due to movement of the p^{th} particle[4] at the ij^{th}edge (indexed by its two end nodes, i^{th} and j^{th} nodes) is given in terms of the line integral of the Whitney 1-form along the path from $\mathbf{r}_p^{(n)}$ to $\mathbf{r}_p^{(n+1)}$, i.e.

$$\begin{aligned} j_{ij,p}^{(n+\frac{1}{2})} &= \frac{q_p}{\Delta t} \int_{\mathbf{r}_p^{(n)}}^{\mathbf{r}_p^{(n+1)}} \mathbf{W}_{ij}^1(\mathbf{r}_p) \cdot d\mathbf{l} \\ &= \frac{q_p}{\Delta t} \left[\widehat{\lambda}_i(\mathbf{r}_p^{(n)}) \widehat{\lambda}_j(\mathbf{r}_p^{(n+1)}) - \widehat{\lambda}_i(\mathbf{r}_p^{(n+1)}) \widehat{\lambda}_j(\mathbf{r}_p^{(n)}) \right], \end{aligned} \tag{5.12}$$

where $\mathbf{W}_{ij}^1(\mathbf{r}_p)$ is the Whitney-1 form with respect to ij^{th} edge calculated at $\mathbf{r}_p$. In the above, we have used the standard relation between Whitney 1-forms and Whitney 0-forms to evaluate the integral in closed form [132]. The total current density generated at the ij^{th} edge is obtained by summing the contributions from all superparticles:

$$j_{ij}^{(n+\frac{1}{2})} = \sum_p \frac{q_p}{\Delta t} \left[\widehat{\lambda}_i(\mathbf{r}_p^{(n)}) \widehat{\lambda}_j(\mathbf{r}_p^{(n+1)}) - \widehat{\lambda}_i(\mathbf{r}_p^{(n+1)}) \widehat{\lambda}_j(\mathbf{r}_p^{(n)}) \right]. \tag{5.13}$$

It can be shown that due to this intrinsic relation between Whitney-1 and 0 forms, the following discrete charge continuity equation (5.14) is always satisfied

$$\frac{\mathbf{q}^{(n+1)} - \mathbf{q}^{(n)}}{\Delta t} + \widetilde{\mathbf{S}} \cdot \mathbf{j}^{(n+\frac{1}{2})} = 0. \tag{5.14}$$

For more details, the reader is referred to [144, 132].

5.1.6 Computational Challenges

High-fidelity EMPIC simulations in general suffer from high runtime and large memory requirements. Along with the potentially large number of mesh elements,

[4]Note that the j_{ij} in (5.13) and j_i in (5.2a) represent the same quantity. The only difference is how we index the mesh element edge over which j is defined. In (5.13) we use the indices of the two end nodes to index an mesh element edge, while we directly use edge indices in (5.2a).

the position and velocities of thousands or millions of superparticles need to be stored in memory to execute each time step.

The computational complexity of a typical EMPIC algorithm for each time step is given by $\mathcal{O}(N^s + N_p)$ [152], where N is the aggregate mesh size and N_p is the total number of superparticles. The specific dependence of computational complexity on the mesh size depends on the particular choice of field update scheme (5.3). Typically for implicit field updates, $s \geq 1.5$. However, for an explicit field-update scheme such as in (5.3), $s = 1$. Depending on the problem setup, especially if the solution domain is large, resulting in large N, the repeated solution of (5.3) for many time steps can pose significant computational challenges.

The dependence on N_p arises from the EMPIC stages involving the superparticles, namely, the *gather*, *pusher* and *scatter* stages. These stages require the EMPIC algorithm to perform computations across the entire set of superparticles at any given time step. Although these computations are embarrassingly parallelizable, they can become a serious bottleneck unless a extremely large number of processors are available. Typically, the number of superparticles is far greater than the number of mesh elements ($N_p \gg N$), resulting in $\mathcal{O}(N + N_p) \approx \mathcal{O}(N_p)$. In other words, the cost of superparticles computations can dominate the computational burden of EMPIC simulations in serial computers.

5.2 DMD Applied to EMPIC Self-Fields

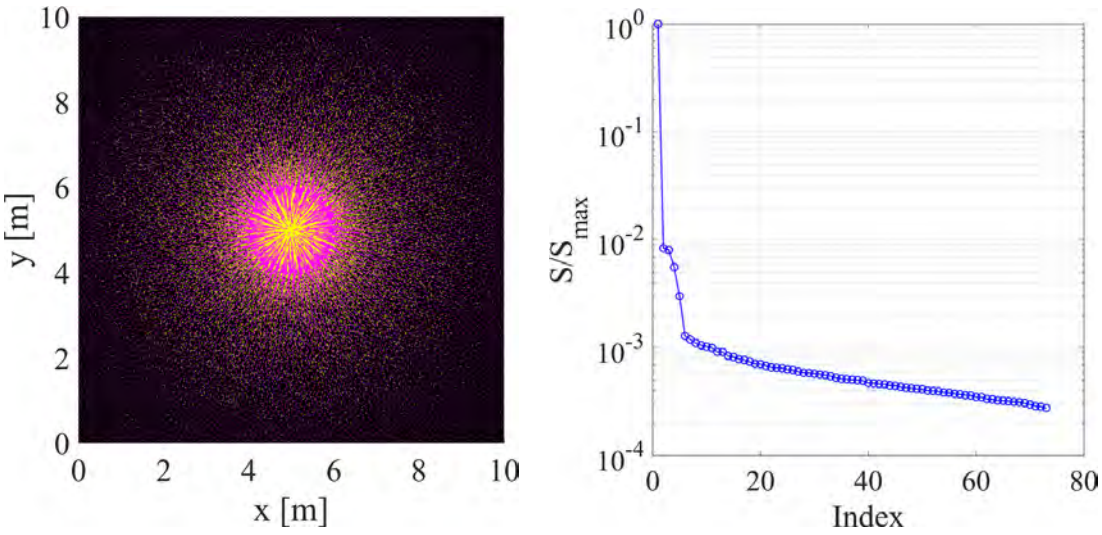
In this section, the effectiveness of DMD in the modeling and prediction of self electric fields as well as its effect on the particle dynamics is demonstrated. The proposed equilibrium detection algorithm is discussed in details in Chapter 3. This section presents three test cases. The first two examples consider a two dimensional (2-D) plasma ball expansion and an oscillating electron beam respectively. We establish the effectiveness of DMD in extracting low dimensional key features from self electric field data $\mathbf{e}(t)$ of EMPIC kinetic plasma simulations and reconstruct the data with good accuracy. Finally, we investigate the extrapolation accuracy beyond the detected equilibrium point for both the predicted fields as well as the particle dynamics. The final example deals with virtual cathode formation, where the main focus is on the accuracy of predicted self-fields and its effect on particle dynamics. We treat the data generated from high-fidelity EMPIC simulation as the "ground truth" to evaluate DMD performance. Nevertheless, for long-term predictions, we should keep in mind that long simulation runtimes might introduce numerical noise in high-fidelity data queried at later time due to "numerical heating" effects [153]. Due to this and other sources of numerical error mentioned earlier, some of the dominant DMD eigenvalues might not lie exactly on the unit circle. After detecting equilibrium and before extrapolation, we adjust the dominant DMD eigenvalues in the radial direction so that they are exactly on the unit circle. Note that for all the cases we use two Hankel stacks, i.e. $d = 2$.

5.2.1 Plasma ball expansion

The solution domain is a $L \times L$ square two-dimensional cavity ($L = 10$ m: see Fig. 5.2a). It is discretized using an irregular triangular mesh with $N_0 = 8037$ nodes, $N_1 = 23797$ edges and $N_2 = 15761$ triangles. Superparticles are initially placed at the center of the cavity within a circle of radius 0.5 m. The plasma ball is initially assumed to be neutral as each electron-ion pair is initially located at the exact same position. All four sides of the cavity are assumed to be perfect magnetic conductors (PMC). Superparticles are given an initial radial velocity with Maxwellian distribution. The time step interval is 0.1 ns and each superparticle represents 2×10^5 electrons. Superparticles are absorbed as they hit the boundary. We store the data at every 500^{th} time steps until $n = 500000$.

Self Electric Field Reconstruction

In equilibrium, the self electric field attains a steady state with a constant spatial configuration (Fig. 5.2a). For extracting low-dimensional features in equilibrium through DMD, we harvest data from $n = 200500$ to $n = 275000$ with interval $\Delta_t = 100$ ns between consecutive snapshots. A selection of $r = 19$ leads to 12 DMD modes, effectively reducing the degrees of freedom from 23797 to only 12. Fig. 5.2b shows the exponential decay of singular values, revealing the dominance of a single mode. The DMD eigenvalue distribution in the complex plane and dominant stationary mode ($\boldsymbol{\phi}_1^{(ss)}$) field configuration are shown in Fig. 5.3. The modes are numbered according to their energy content ($|A_m|^2$), with $\boldsymbol{\phi}_1^{(ss)}$ being the most energetic mode.

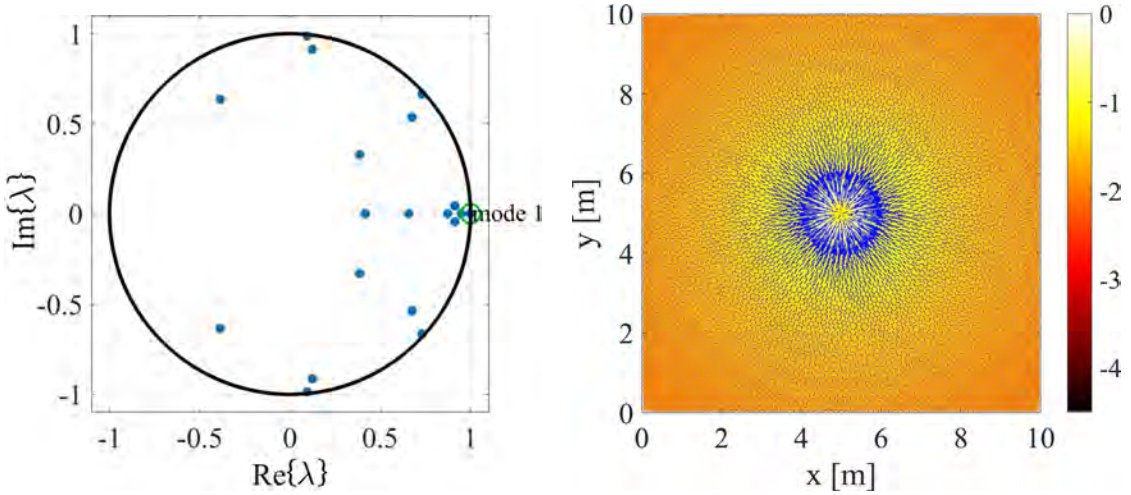


(a) (b)

Figure 5.2: (a) Snapshot of plasma ball expansion at $n = 400000$ in a square cavity. The yellow dots represent superparticles and magenta arrows show the self electric field quiver plot. (b) Normalized singular values from SVD of snapshot matrix in equilibrium state.

With increasing mode indices (decreasing energy), the field configuration becomes more random, as can be observed in the recessive modes (Fig. 5.4).

For comparison, we also perform DMD in the transient state with the harvesting region spanning from $n = 500$ to $n = 75000$, snapshots $\Delta_t = 100$ ns apart. We choose $r = 27$, giving us 15 DMD modes. For comparing the spatial configuration of DMD modes, we plot the absolute value of MAC (ρ) in a matrix form in Fig. 5.5. Unlike other projection-based reduced order model techniques such as the proper orthogonal decomposition (POD), DMD does not enforce orthogonality of modes in the spatial domain, which explains the presence of nontrivial off-diagonal elements. At the same

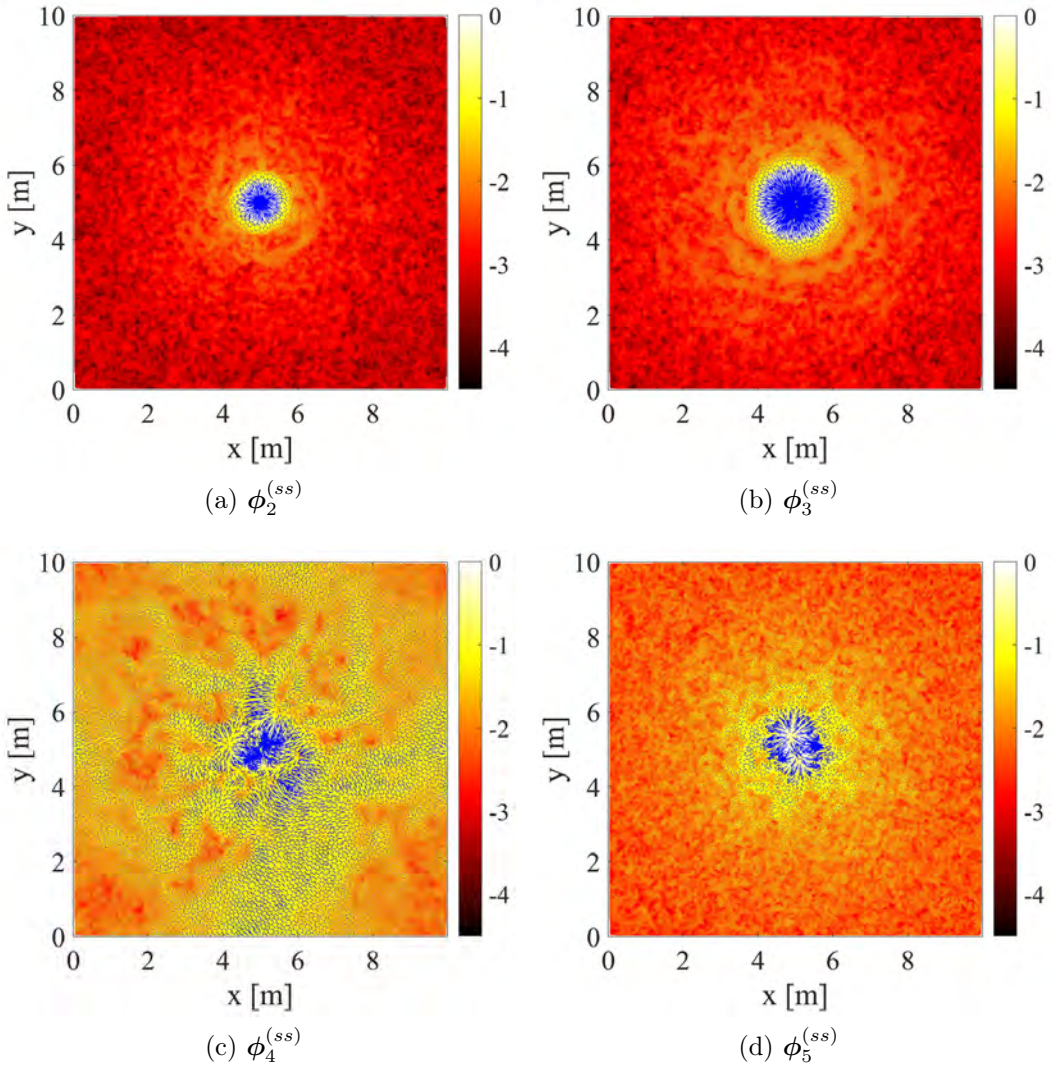


(a) (b) $\boldsymbol{\phi}_1^{(ss)}$

Figure 5.3: (a) DMD eigenvalues in the complex plane. The green circle denotes the dominant mode and black curve indicates the unit circle. (b) Dominant mode $\boldsymbol{\phi}_1^{(ss)}$. The blue arrows show the self electric field quiver plot. The colormap indicates logarithm (base 10) of amplitude.

time, Fig. 5.5 reveals a clear distinction among various equilibrium DMD modes. On the other hand, transient-state DMD modes are less distinguishable from each other due to more complex dynamics. The 2-norm relative error in reconstructed self electric field is shown in Fig. 5.6 for different sampling rates. The 2-norm relative error in DMD reconstruction ($\hat{\mathbf{e}}$) compared to the full-order solution ($\mathbf{e}$) at n^{th} time step ($\delta^{(n)}$) is given by,

$$\delta^{(n)} = \frac{||\hat{\mathbf{e}}^{(n)} - \mathbf{e}^{(n)}||_2}{||\mathbf{e}^{(n)}||_2}. \tag{5.15}$$

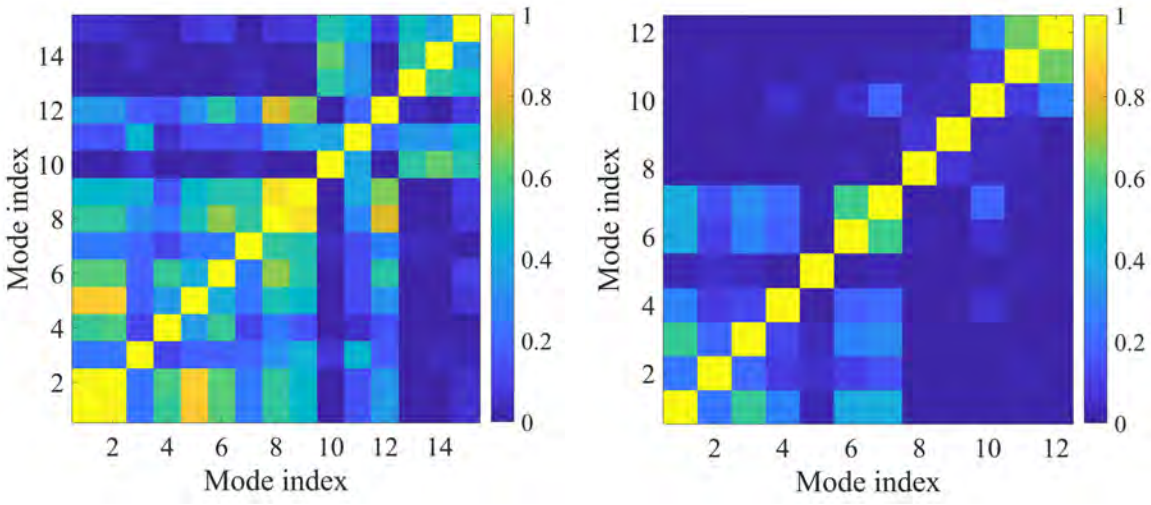


(a) $\phi_2^{(ss)}$ (b) $\phi_3^{(ss)}$

(c) $\phi_4^{(ss)}$ (d) $\phi_5^{(ss)}$

Figure 5.4: First four recessive DMD modes for plasma ball in equilibrium.

As expected, decreasing sampling interval ensures better accuracy, but the solution diverges more rapidly in the extrapolation region. Note that the surprisingly good performance for $\Delta_t = 200$ ns in Fig. 5.6a can be attributed to the aliasing effect

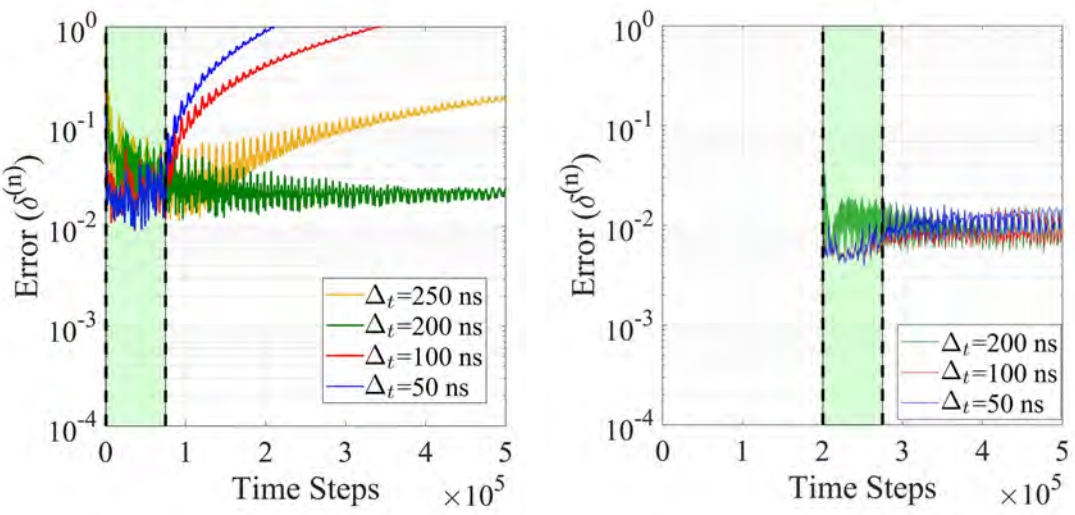


(a) Transient state DMD modes

(b) Equilibrium state DMD modes

Figure 5.5: (a) Absolute value of MAC coefficient ρ between transient state DMD modes. (b) Coefficient ρ between equilibrium state DMD modes.

for this particular sampling interval. The $\Delta_t = 200$ ns case is an anomaly for which the DMD frequencies are such that it produces a stable solution with relative error oscillating around a fixed value. This is further confirmed by the fact that the $\Delta_t = 250$ ns case continues to follow the trend as shown by the $\Delta_t = 50$ and 100 ns cases. Higher error at the beginning of the simulation can be attributed simply to the very low field magnitudes then, causing a spike in the relative error. The extrapolation error is higher for transient state DMD compared to DMD in the equilibrium state, which further evokes the need to correctly determine the equilibrium state for good prediction accuracy.



(a) Transient state DMD

(b) Equilibrium state DMD

Figure 5.6: (a) 2-norm relative error when the DMD window (green shaded area) is in the transient region. (b) 2-norm relative error when the DMD window (green shaded area) is in the equilibrium region.

Sliding-Window DMD

In the plasma ball expansion case, the self-fields attain a non-oscillatory steady state in equilibrium. This makes the prediction task trivial once equilibrium is detected and presents an opportunity to verify the accuracy of the equilibrium detection algorithm. First, we discuss the robustness of our algorithm with respect to the sampling interval and sliding window width. Then, the convergence of dominant mode shapes and accuracy of predicted particle dynamics is presented.

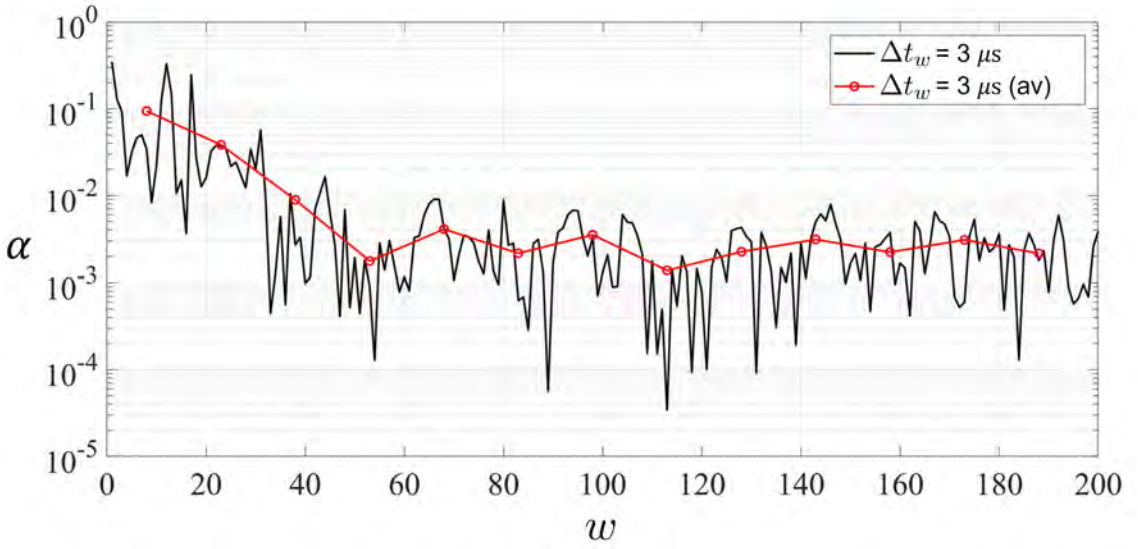


Figure 5.7: Variation in $\alpha(\Delta t_w)$ as the window slides towards the equilibrium state for $\Delta_t = 100$ ns.

Equilibrium Detection: Algorithm 2 is used for identifying the onset of equilibrium state with $\beta_{thr} = 0.01$ and $\delta t_w = 200$ ns. In this case we know that the fields will eventually attain steady state without limit-cycle oscillations, whereby the selection of Δt_w is not a critical factor. We choose $\Delta t_w = 3$ μs with 30 snapshots inside the harvesting window. Starting and ending points of the w^{th} DMD window are given by $n_{st}(w) = 500 + (w-1) \times \delta n_w$ and $n_{en}(w) = (500 + \Delta n_w) + (w-1) \times \delta n_w$ respectively.

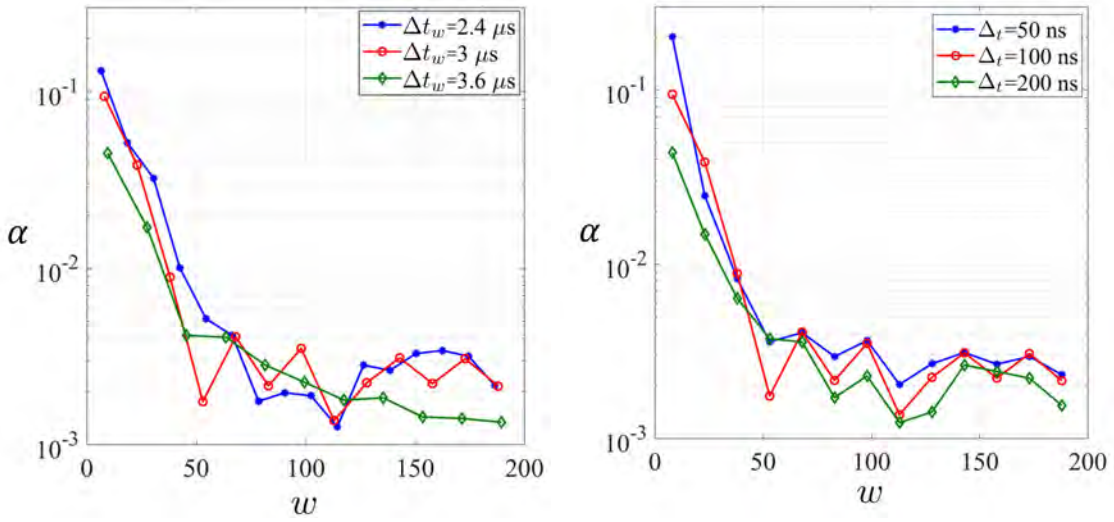


Figure 5.8: (a) Sensitivity of Algorithm 2 towards window width Δt_w ($\pm 20\%$), keeping fixed $\Delta_t = 100$ ns. (b) Sensitivity of Algorithm 2 towards sampling interval Δ_t, keeping fixed $\Delta t_w = 3$ μs.

As seen in Fig. 5.7, $\alpha(\Delta t_w)$ decreases initially with increasing w and eventually converges, the knee/elbow region marking the transition from transient to steady-state. The algorithm detects the steady-state at $w = 75$ ($n_{st}(75) = 148500$). The sensitivity of $\alpha(\Delta t_w)$ towards variation in Δt_w and Δ_t is shown in Figs. 5.8a and 5.8b respectively. For better comparison, we set $\delta t_w = 200$ ns for all the three cases in Fig. 5.8b. Using the non-negative slope criterion, the algorithm stops at $w = 96, 75$ and 144 for $\Delta t_w = 2.4$ μs, 3 μs and 3.6 μs respectively. As explained in 3.1.3, non-negative slope is employed to indicate a "knee", which can potentially delay the

detection of equilibrium. For $\Delta t_w = 3.6$ μs in Fig. 5.8a, $\alpha(\Delta t_w)$ encounters non-negative slope at a much later time compared to the other two cases, even though the actual knee region appears earlier. For $\Delta_t = 50$ ns, 100 ns and 200 ns, the algorithm detects equilibrium at $w = 75, 75$ and 105 respectively.

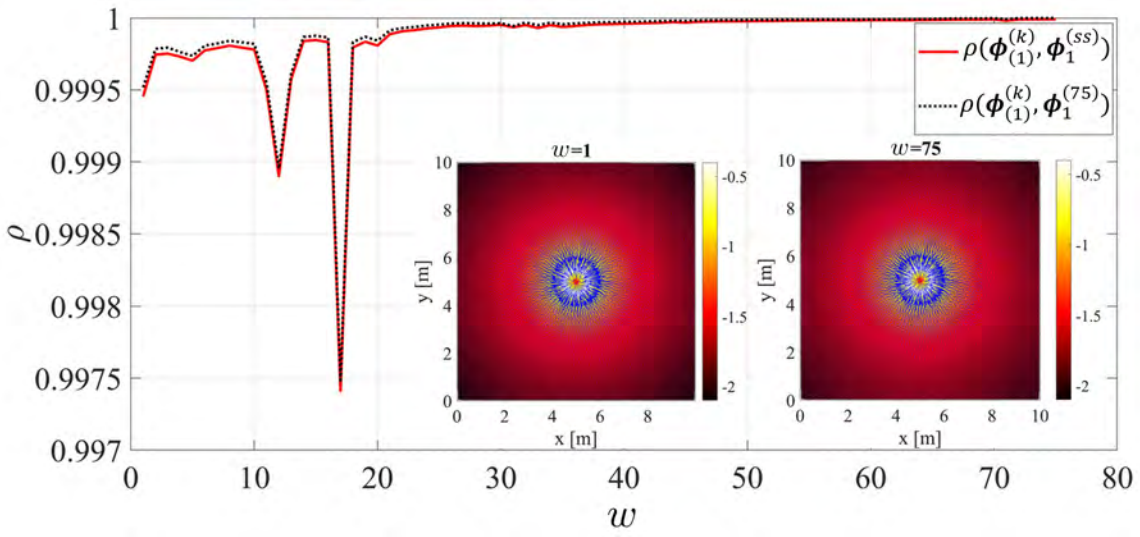
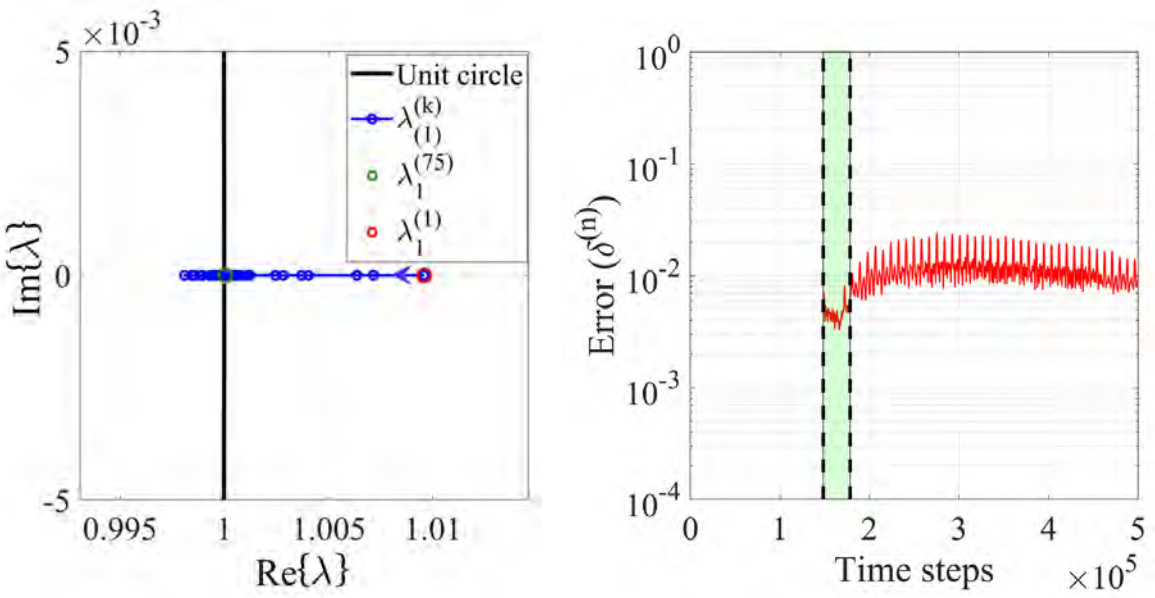


Figure 5.9: Correlation coefficient (ρ) of $\boldsymbol{\phi}_1^{(75)}$ with its predecessors (black dotted curve). ρ between $\boldsymbol{\phi}_1^{(ss)}$ and predecessors of $\boldsymbol{\phi}_1^{(75)}$ (red curve). Inset: $\boldsymbol{\phi}_1^{(75)}$ and its predecessor $\boldsymbol{\phi}_1^{(1)}$ at $w = 1$.

Convergence in DMD Mode Shapes: Algorithm 1 helps track the evolution of DMD mode shapes through the parameter ρ. We correlate the steady-state mode $\boldsymbol{\phi}_1^{(ss)}$ (from sec. 5.2.1) and dominant DMD mode in the last window $\boldsymbol{\phi}_1^{(75)}$ with its predecessors $\boldsymbol{\phi}_{(1)}^{(k)}$ ($k = 1, 2, \ldots, 75$, where $\boldsymbol{\phi}_{(1)}^{(75)} = \boldsymbol{\phi}_1^{(75)}$) . As can be seen in Fig.



(a) (b)

Figure 5.10: (a) Movement the eigenvalue corresponding to $(\lambda_1^{(75)}, \boldsymbol{\phi}_1^{(75)})$. (b) 2-norm relative error in self electric field reconstruction. The green shaded area denotes the DMD window corresponding to $w = 75$.

5.9, the high value of ρ indicates that the dominant mode shape remains almost time invariant. Close proximity of red and black curves (Fig. 5.9) further confirms that the equilibrium is attained at $w = 75$ as $\boldsymbol{\phi}_1^{(ss)}$ and $\boldsymbol{\phi}_1^{(75)}$ are almost identical. Fig. 5.10a shows convergent movement of the dominant eigenvalue towards the unit circle.

Prediction of Self-Fields and Particle Dynamics: The high-fidelity simulation is stopped after detecting the equilibrium state. The final data harvesting window

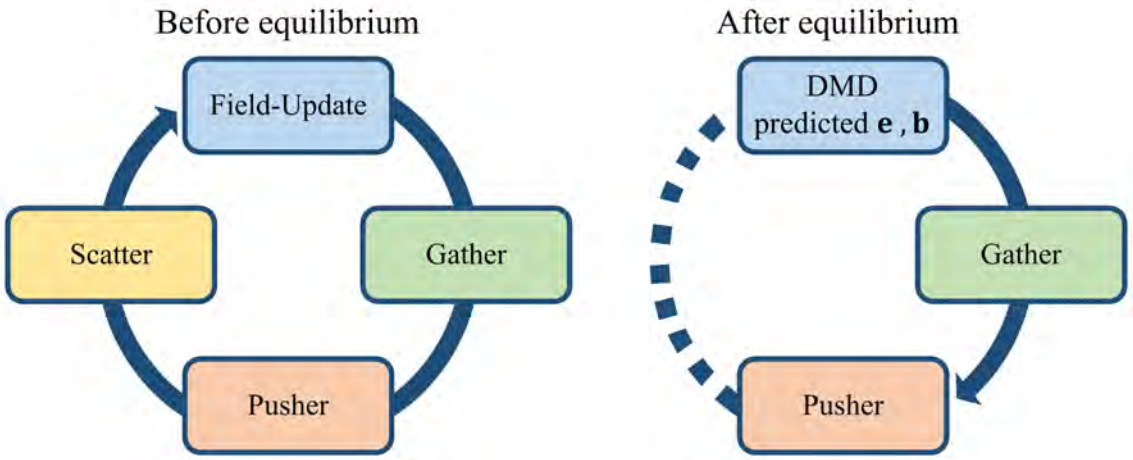


Figure 5.11: Schematic representation of EMPIC algorithm with DMD predicted self-fields. Prior to detection of equilibrium state, the EMPIC algorithm consists of usual four stages. After the equilibrium is detected, we perform DMD to extrapolate self-field values and utilize those values bypassing field-update stage for future time. To observe the effect of predicted self-fields on particle behavior, we also perform the gather stage and particle pusher stage.

($w = 75$) is then used for extrapolation (Fig. 5.10b). Again, as mentioned earlier, extrapolating the self-fields in this particular example is trivial because of the non-oscillatory steady-state nature of the solution.

As we are interested in how this predicted self electric field affects the (predicted) particle dynamics, we will substitute it in place of the self electric field generated by original EMPIC algorithm. However, we can entirely bypass the field solver (update) stage of the EMPIC algorithm as illustrated in Fig. 5.11 by performing DMD on the self magnetic flux $\mathbf{b}(t)$ as well. In this work we identify the equilibrium

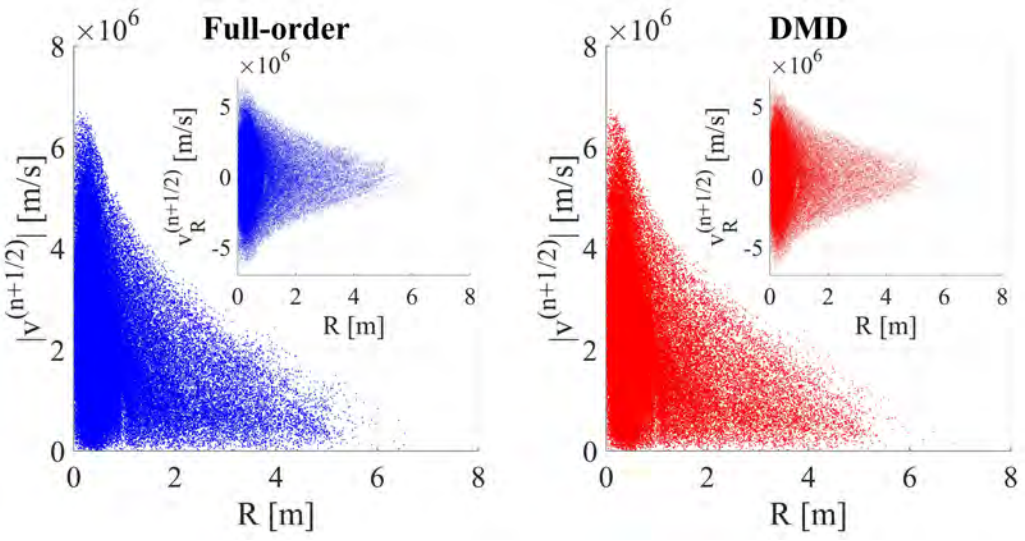


Figure 5.12: Phase-space plot comparison between finite-element full-order EMPIC simulation (blue) and reduced-order DMD (red) in extrapolation region ($n = 225000$). Phase-space plot for absolute velocity and radial distance (R) from center of mesh (5,5). Inset: Phase-space plot corresponding to radial velocity and radial distance.

performing sliding-window DMD on electric field dataset and extrapolate both the self electric and magnetic field from the last DMD window. However, the DMD extrapolated self-fields do not ensure energy conservation in the extrapolated region. To the extent that the extrapolated fields remain close to the original solution, energy is approximately conserved in the extrapolation region given that the high-fidelity algorithm itself is energy-conserving.

We next compare the particle dynamics generated from the full-order and reduced-order DMD in the extrapolation region at $n = 225000$, beyond the final snapshot ($n = 178500$) of the last window. Fig. 5.12 shows a good match between the phase

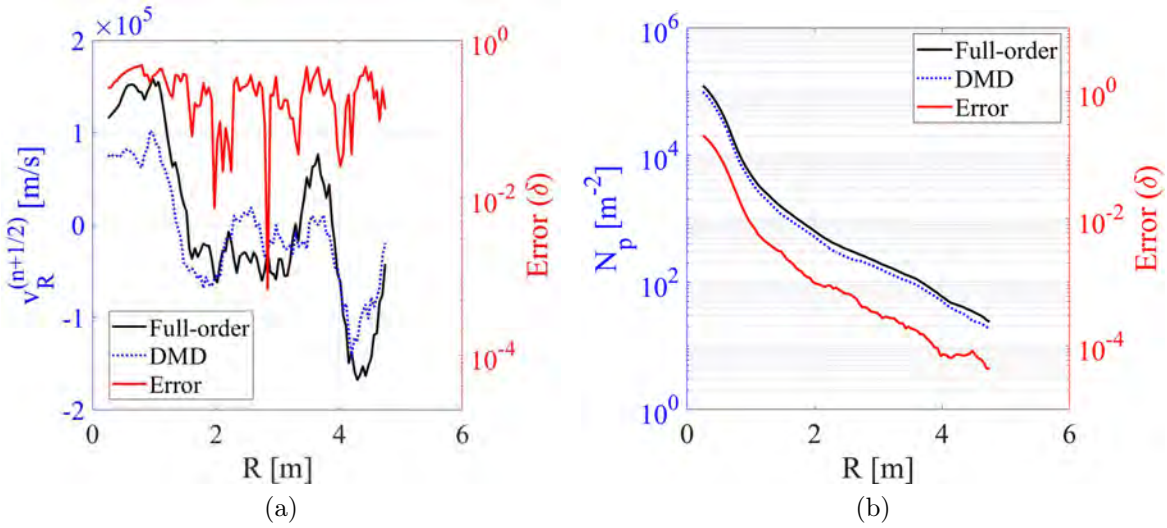
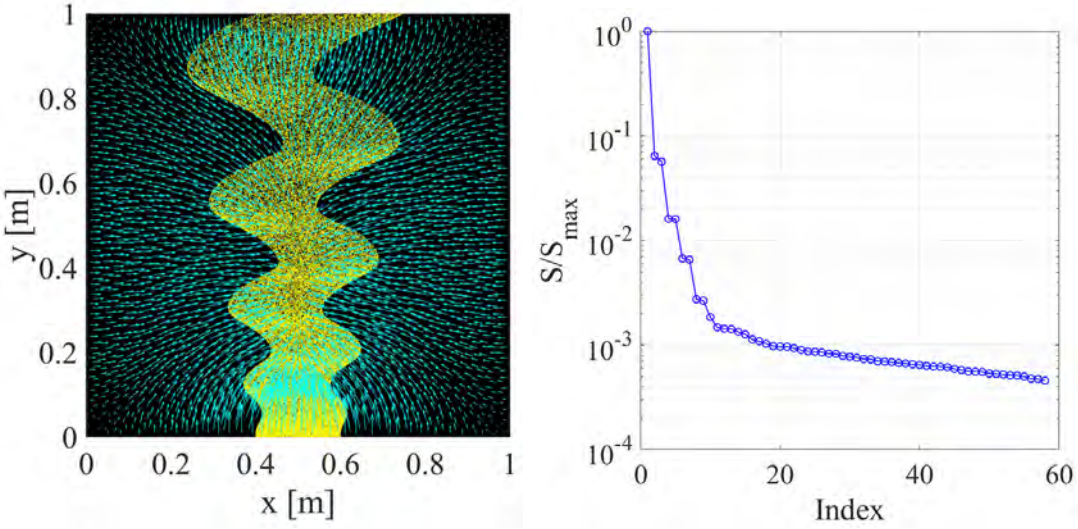


Figure 5.13: Particle dynamics comparison at $n = 225000$. (a) Radial variation of average radial velocity of particles. (b) Radial variation of particle density. For both cases, relative error is defined as $\delta = |\hat{\mathcal{X}}(R) - \mathcal{X}(R)|/\max|\mathcal{X}(R)|$, where $\mathcal{X}$ represents either $v_R^{(n+1/2)}$ or N_p and "hat" denotes DMD approximation.

space plots of the full-order and reduced-order models in the radial direction (R). Fig. 5.13 compares the average radial velocity and particle density as a function of radial distance from the center of the plasma ball. For calculating the average radial particle velocity and particle density at R, we consider a thin annular region with outer radius $R + L/40$ and inner radius $R - L/40$ and perform the averaging for all the particles present inside that annular region. It is clear that the predicted fields produce good prediction of the particle dynamics and thus have the potential to speed-up EMPIC simulations for long term predictions.

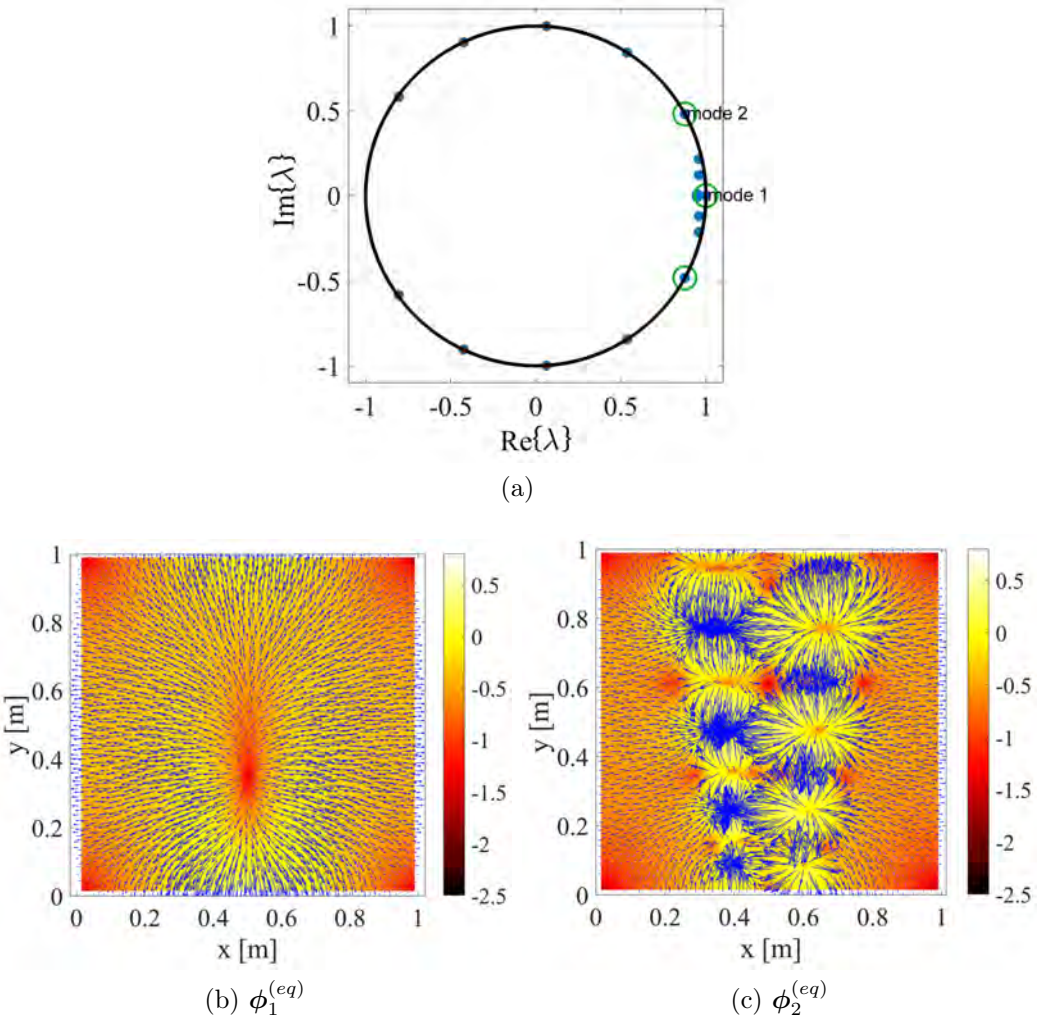
5.2.2 Oscillating Electron Beam



(a) (b)

Figure 5.14: (a) Snapshot of an oscillating 2-D electron beam at $n = 64000$ in a square cavity, propagating along +ve y direction. The cyan arrows show the self-electric field lines. (b) Normalized singular values from SVD of snapshot matrix in equilibrium state.

Consider the case of a 2-D electron beam propagation along the positive y direction in the xy plane, under the influence of an external oscillating transverse magnetic flux (Fig. 5.14a). The solution domain is a square cavity of size 1 m $\times$ 1 m that is discretized via an irregular triangular mesh wit $N_0 = 1647$ nodes, $N_1 = 4788$ edges and $N_2 = 3142$ triangles. Superparticles are injected randomly with a uniform distribution at the bottom of the cavity in the region $[0.5 - b_h, 0.5 + b_h]$. Here, $b_h = 0.1$ m is the half-beam width. All four sides of the cavity are assumed to be perfect electric



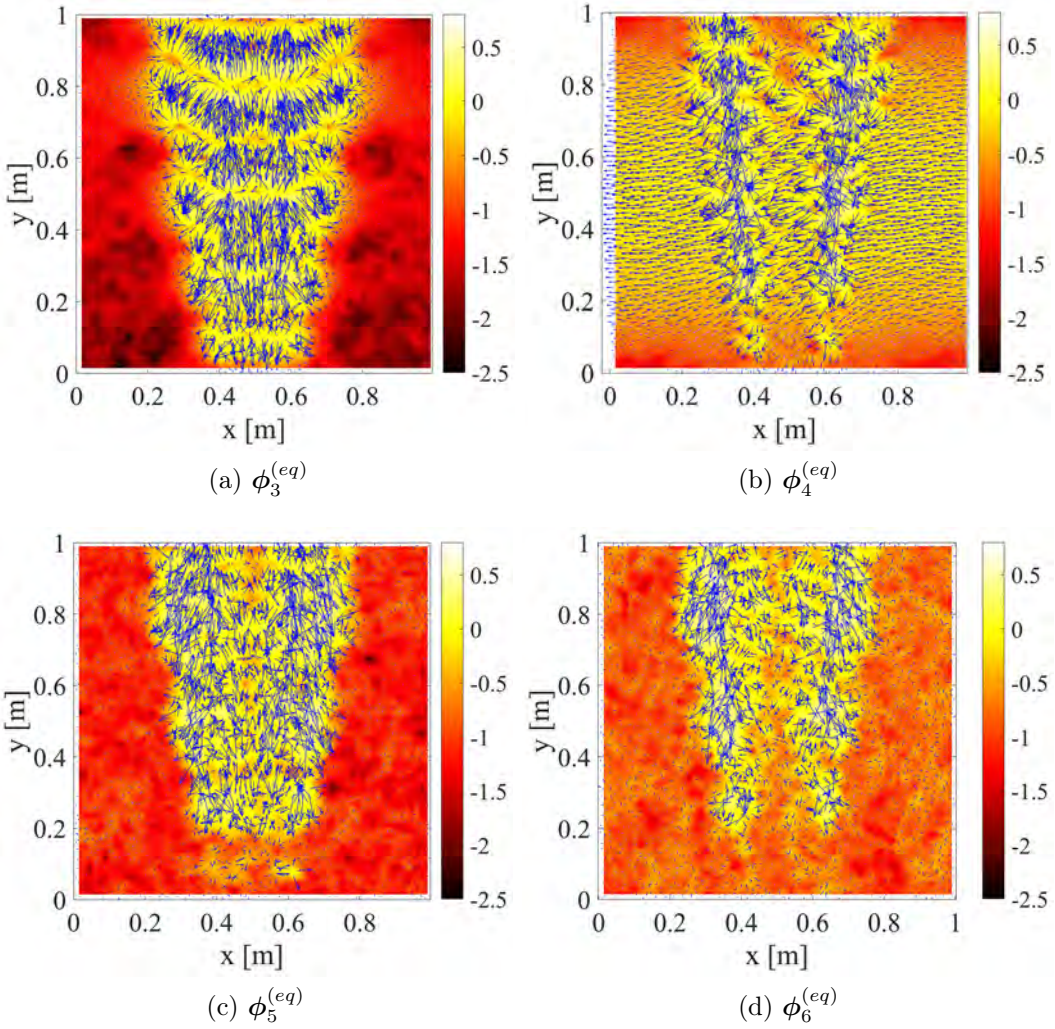
(a)

(b) $\phi_1^{(eq)}$

(c) $\phi_2^{(eq)}$

Figure 5.15: Oscillating electron beam case: (a) DMD eigenvalues in complex plane when DMD is performed on data from equilibrium. (b) First dominant DMD mode. (c) Second dominant DMD mode.

conductors (PEC). Superparticles are injected with initial velocity $v_0 = 5 \times 10^6$ m/s along the positive y direction at rate 10 superparticles (1 superparticle $\equiv 2 \times 10^5$ electrons) per time step (= 0.01 ns). The external voltage bias is set to $V_b = 2 \times 10^3$



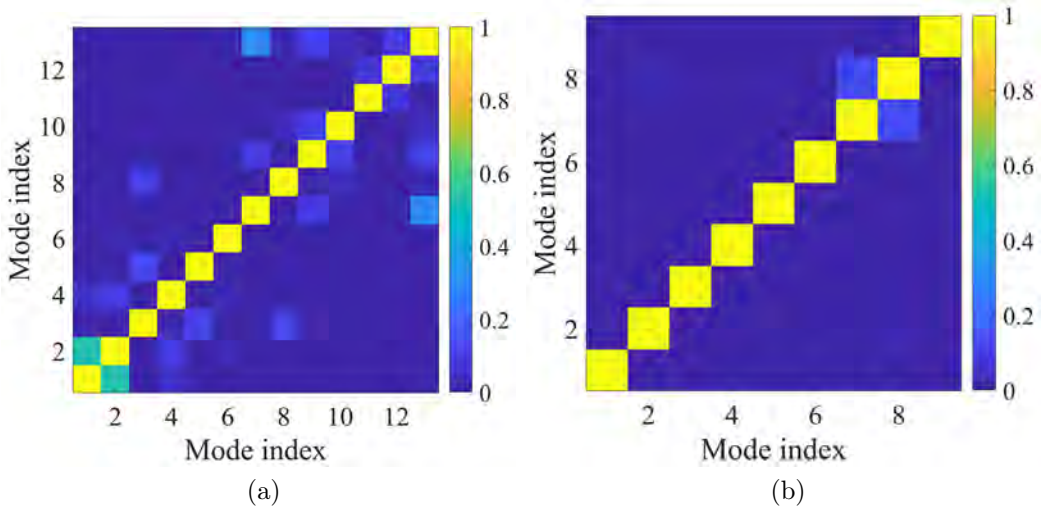
(a) $\boldsymbol{\phi}_3^{(eq)}$ (b) $\boldsymbol{\phi}_4^{(eq)}$

(c) $\boldsymbol{\phi}_5^{(eq)}$ (d) $\boldsymbol{\phi}_6^{(eq)}$

Figure 5.16: First four recessive DMD modes extracted from equilibrium region of oscillating electron beam.

V and external magnetic flux to $B_{ext} = B_0 \sin(2\pi t/T_b)\ \hat{\mathbf{z}}$, where $B_0 = 10^{-3}$ T and $T_b = 20$ ns. Superparticles are absorbed as they hit the upper boundary. Time series

data of degrees of freedom (DoF) of self-fields is stored at every 80^{th} time step. The data set spans $n = 80$ to $n = 80000$ (1000 datapoints).



(a) (b)

Figure 5.17: Oscillating electron beam case: (a) Coefficient ρ between DMD modes from transient region. (b) Coefficient ρ between DMD modes from equilibrium region.

Self Electric Field Reconstruction

Transience ends shortly after the beam reaches the upper boundary of the domain. The DMD window in equilibrium spans form $n = 40080$ to $n = 49600$, with consecutive samples $\Delta_t = 1.6$ ns apart. As seen in Fig. 5.14b, energy is primarily concentrated in the first few (~ 10) modes, revealing existence of underlying low-dimensional coherent features. We truncate the SVD matrices at $r = 17$, generating 9 DMD modes, resulting in a reduced-order model with only 9 degrees of freedom

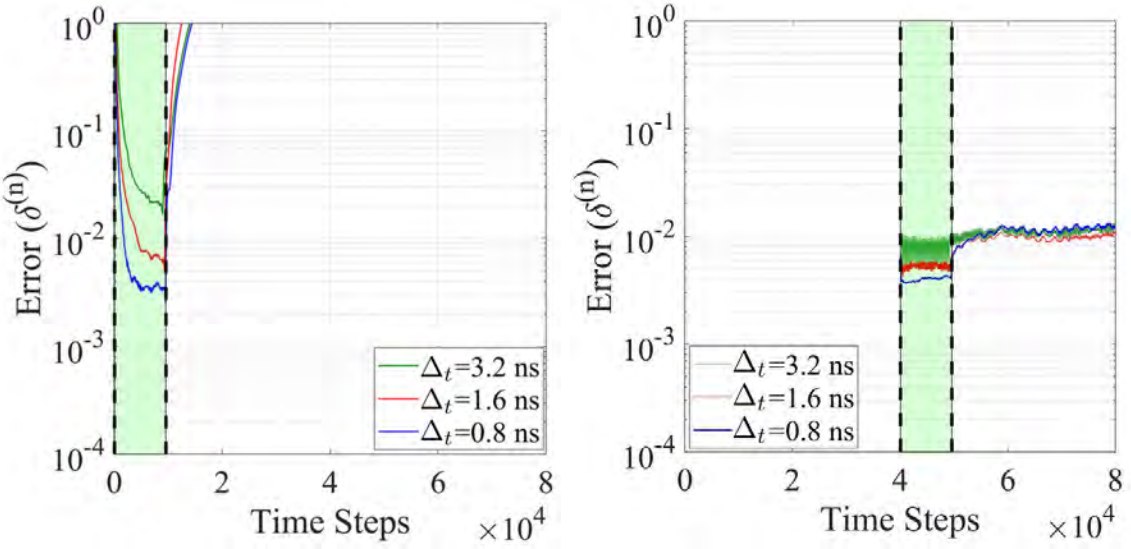


Figure 5.18: Relative 2-norm error for reconstruction of self electric field for oscillating electron beam, with green shaded area denoting the DMD window. (a) DMD window is in transient region. (b) DMD window is equilibrium region.

compared to 4788 in the full-order finite element model. Fig. 5.15a with dominant eigenvalues highlighted using green circles indicate that DMD is able to successfully extract the stationary component $\boldsymbol{\phi}_1^{(eq)}$ and the oscillating component $\boldsymbol{\phi}_2^{(eq)}$ from the equilibrium state, with the oscillation frequency matching the frequency of oscillation of external magnetic flux. In equilibrium, these two modes contain more than 99% of the energy.

As for the plasma ball example, we perform DMD during transience as well. This DMD window spans from $n = 80$ to $n = 9600$ with $\Delta_t = 1.6$ ns, $r = 25$ and 13

DMD modes. Fig. 5.17 reveals a clear distinction in nature of correlation among equilibrium modes versus correlation among transient modes: the former have greater separation while the latter have more overlap among each other. This phenomenon is similar to the plasma ball case.

Similar to the plasma ball case, the self-field reconstruction error stays within reasonable limits inside the interpolation region, but rapidly increases in the extrapolation region for transient DMD (Fig. 5.18a). However, for DMD in the equilibrium region (Fig. 5.18b), the extrapolation error remains within acceptable bounds.

Sliding-Window DMD

We set $\beta_{thr} = 0.01$ and $\delta t_w = 3.2$ ns. Using prior knowledge about the oscillation period of the external magnetic flux ($T_b = 20$ ns), we choose $\Delta t_w = 56$ ns so that it covers multiple cycles of the forced oscillation. The resulting interval between successive snapshots is $\Delta_t = 1.6$ ns.

Equilibrium Detection: The algorithm detects the steady-state at $w = 136$ ($n_{st}(136) = 43280$). The sensitivity of $\alpha(\Delta t_w)$ towards variation in Δt_w and Δ_t is shown in Fig.5.20.

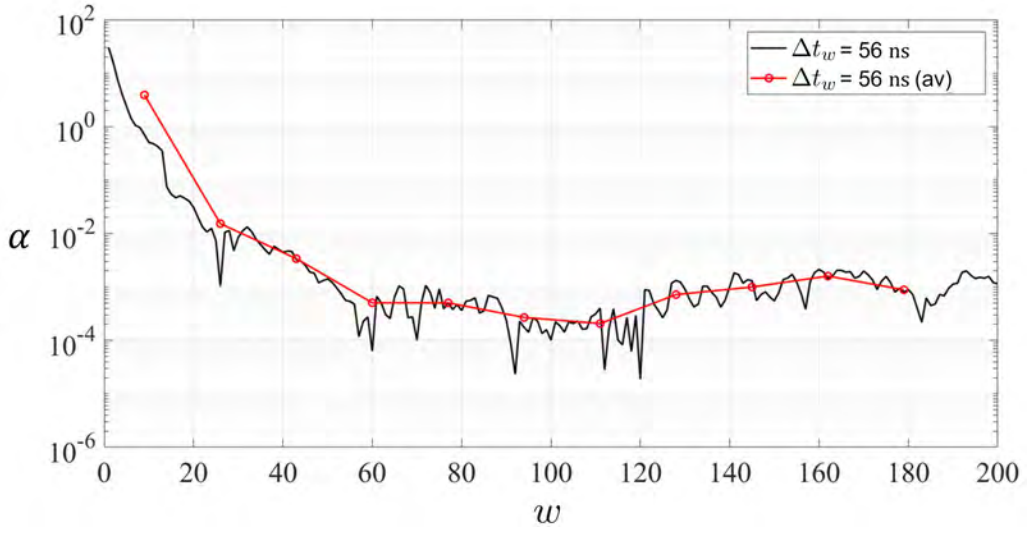


Figure 5.19: Variation in $\alpha(\Delta t_w)$ for $\Delta_t = 1.6$ ns, as the DMD window slides towards equilibrium for oscillating electron beam. The red curve shows averaged α over 17 windows.

Convergence in DMD Mode Shapes: There are two dominant DMD modes that describe equilibrium dynamics: the stationary mode and an oscillating mode corresponding to external magnetic flux oscillation frequency. It is of interest to track their evolution to their final spatial configuration ($\boldsymbol{\phi}_1^{(136)}$ and $\boldsymbol{\phi}_2^{(136)}$) in equilibrium. The tracking algorithm reveals that the dominant stationary mode $\boldsymbol{\phi}_1^{(1)}$ in transient state ($w = 1$) eventually evolves to the dominant stationary mode $\boldsymbol{\phi}_1^{(136)}$ in equilibrium ($w = 136$). The inset in Fig. 5.22 reveals that the mode shape $\boldsymbol{\phi}_1^{(1)}$ at $w = 1$ is nothing but the self-field configuration of the straight beam (stationary component) emitting from the lower boundary of the mesh, whereas that of $\boldsymbol{\phi}_1^{(136)}$ suggests a full

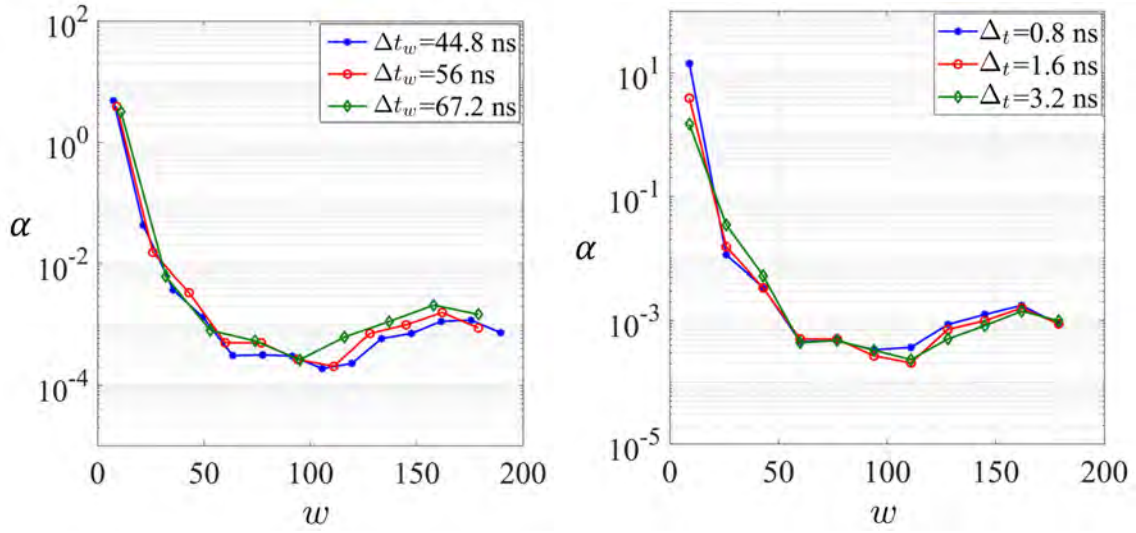


Figure 5.20: (a) Sensitivity of Algorithm 2 towards window width Δt_w ($\pm 20\%$), keeping fixed $\Delta_t = 1.6$ ns. (b) Sensitivity of Algorithm 2 towards sampling interval, keeping fixed $\Delta t_w = 56$ ns.

fledged straight electron beam. Fig. 5.21a shows convergent migration of the DMD eigenvalue towards the unit circle. Similar behavior is observed for the eigenvalue corresponding to the oscillating mode $\boldsymbol{\phi}_2^{(136)}$ (Fig. 5.21b), which is traced back to $\boldsymbol{\phi}_4^{(1)}$ in the first window. Interestingly, the fourth most energetic mode at $w = 1$ evolves to become the second most energetic mode at $w = 136$. Note that during transience, it is harder to separate modes in terms of energy due to complex dynamics and rapidly time varying amplitudes. Tracking evolution of the oscillating mode underscores how a relatively hidden feature in transience can become prominent in

equilibrium. This gradual evolution in mode shape is captured by the continuous variation of parameter ρ as seen in Figs. 5.22 and 5.23.

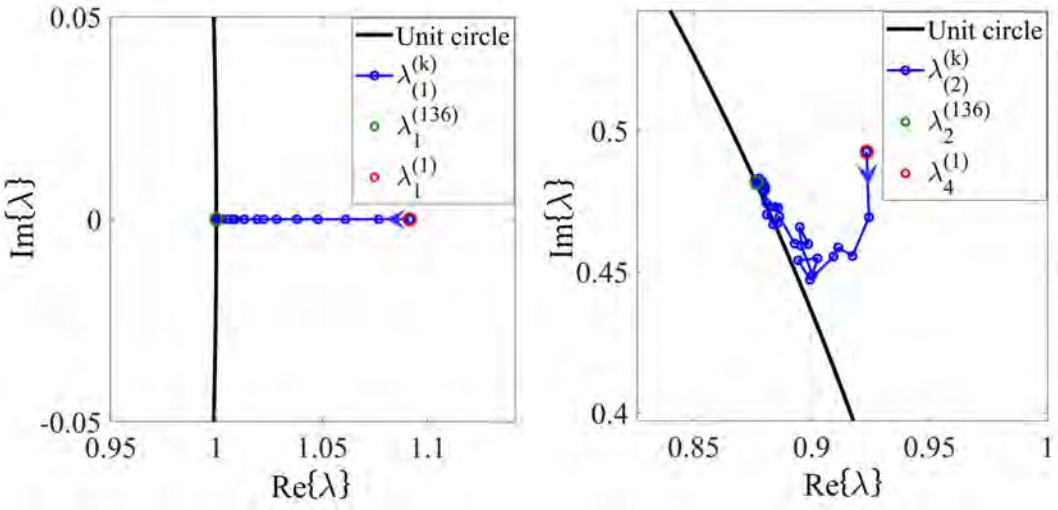


Figure 5.21: (a) DMD eigenvalue movement corresponding to $(\lambda_1^{(136)}, \boldsymbol{\phi}_1^{(136)})$. (b) DMD eigenvalue movement corresponding to $(\lambda_2^{(136)}, \boldsymbol{\phi}_2^{(136)})$.

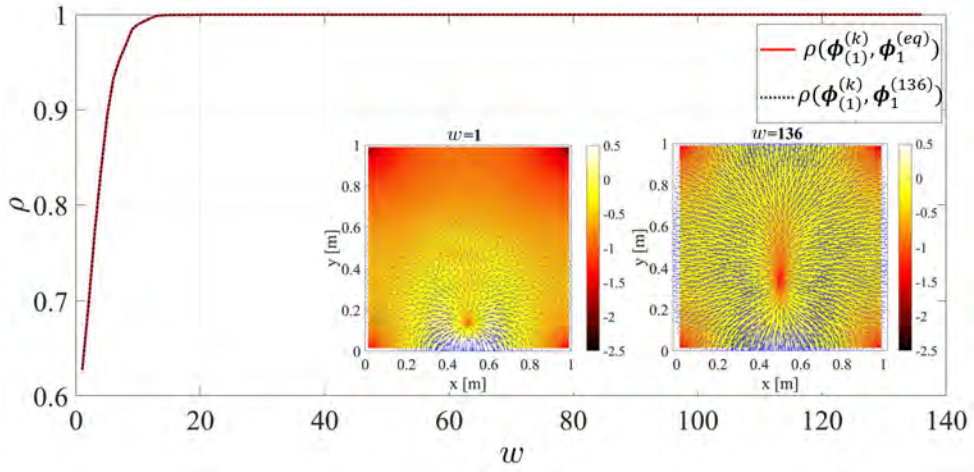


Figure 5.22: Coefficient ρ of late-time DMD mode 1 with its predecessors (black dotted curve). ρ between equilibrium mode 1 and predecessors of late-time DMD mode 1 (red curve). Inset: late-time DMD mode 1 and its predecessor at $w = 1$.

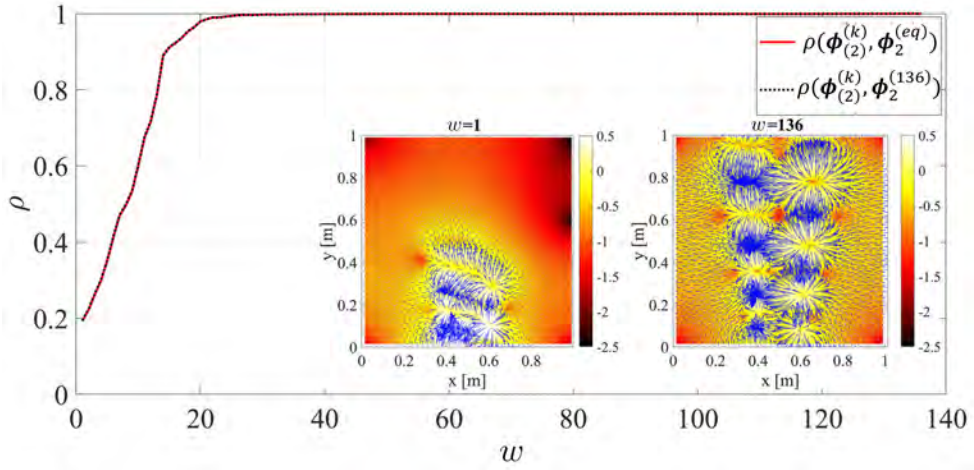


Figure 5.23: Coefficient ρ of $\boldsymbol{\phi}_2^{(136)}$ with its predecessors (black dotted curve). ρ between $\boldsymbol{\phi}_2^{(eq)}$ and predecessors of $\boldsymbol{\phi}_2^{(136)}$ (red curve). Inset: $\boldsymbol{\phi}_2^{(136)}$ and its predecessor at $w = 1$.

Predicted Field and Particle Dynamics: Recall that a key motivation for using a ROM such as DMD is to expedite the EMPIC simulation by predicting future self-fields and particle dynamics. The 2-norm relative error in predicted fields is close to 1% after extrapolation from the window at $w = 136$. We compare the x and y directional phase-space plots (Figs. 5.24-5.25) and the x and y directional average velocity and particle density (Figs. 5.26-5.27) at $n = 76000$, which extends well into the extrapolation region.

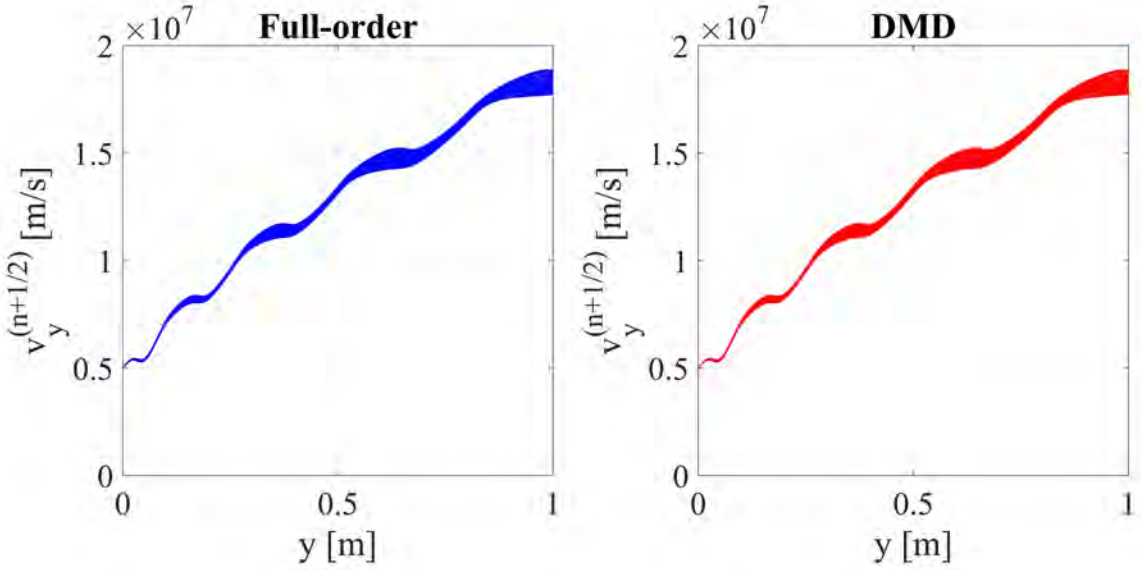


Figure 5.24: The y-directional phase-space plot comparison between finite-element full-order EMPIC simulation (blue) and DMD (red) in extrapolation region ($n = 76000$).

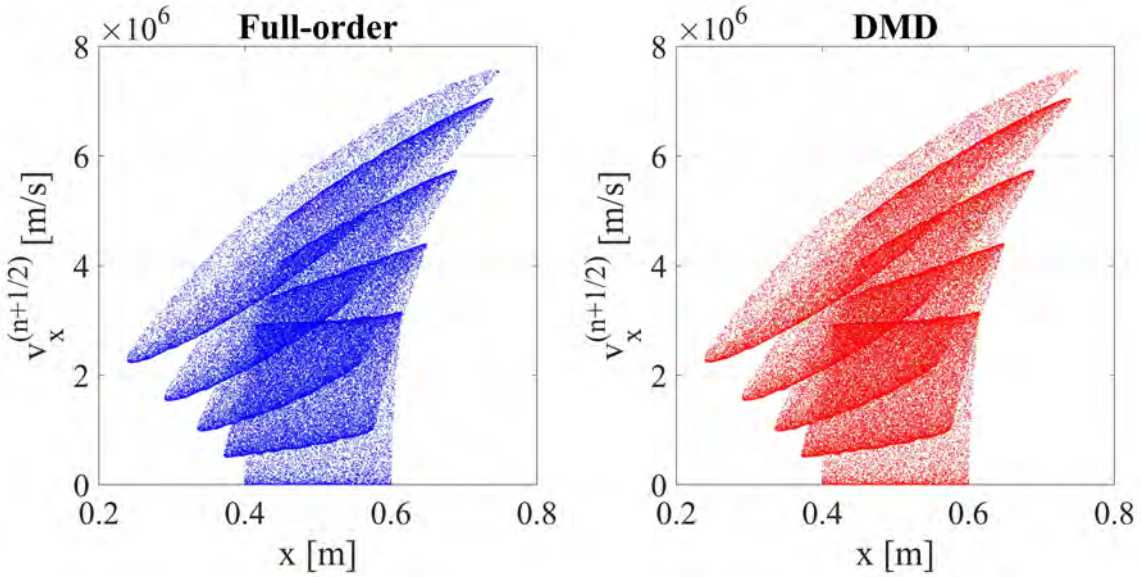


Figure 5.25: The x-directional phase-space plot comparison between finite-element full-order EMPIC simulation (blue) and DMD (red) in extrapolation region ($n = 76000$).

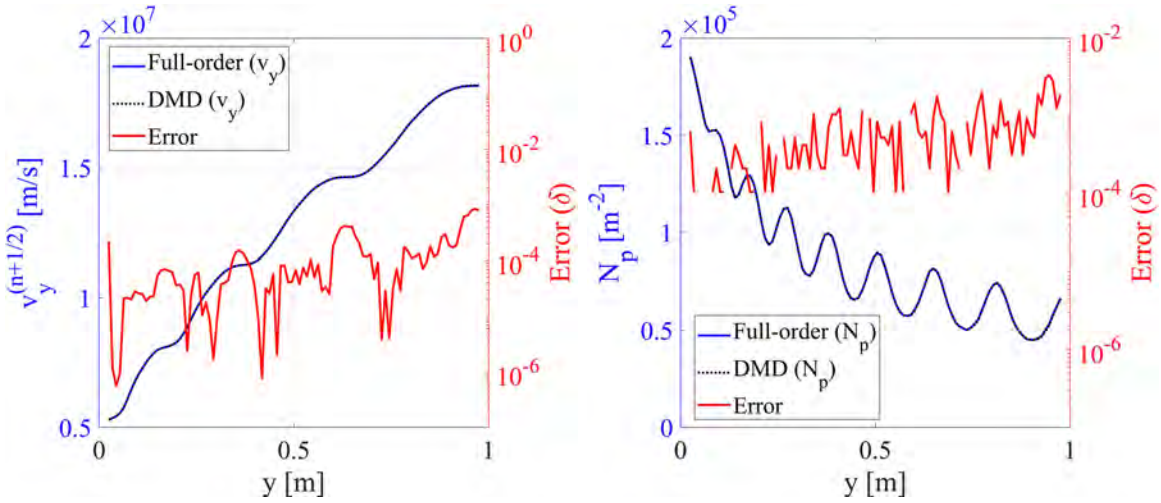


Figure 5.26: Comparison between full-order and DMD predicted average velocity and particle density at $n = 76000$ in the y-direction. Relative error for $\mathcal{X}(y)$ is defined as $\delta = |\hat{\mathcal{X}}(y) - \mathcal{X}(y)|/\max|\mathcal{X}(y)|$, where "hat" denotes the DMD approximation. (a) y-directional average velocity (left axis) and relative error (right axis) plot. (b) Particle density variation along the y-direction (left axis) and relative error plot (right axis). Few missing points in the error graph correspond to points where the error is below the log scale range shown.

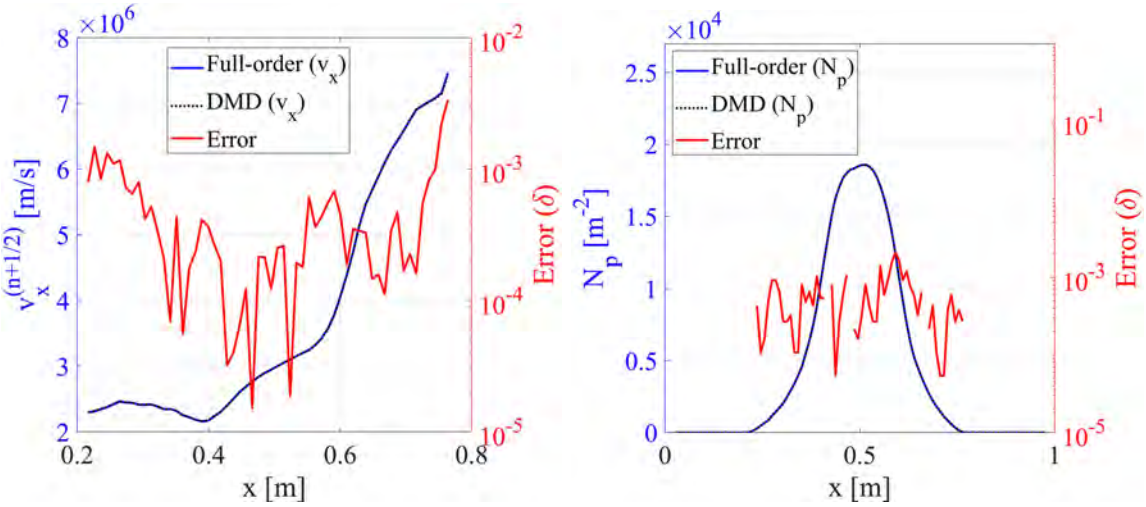
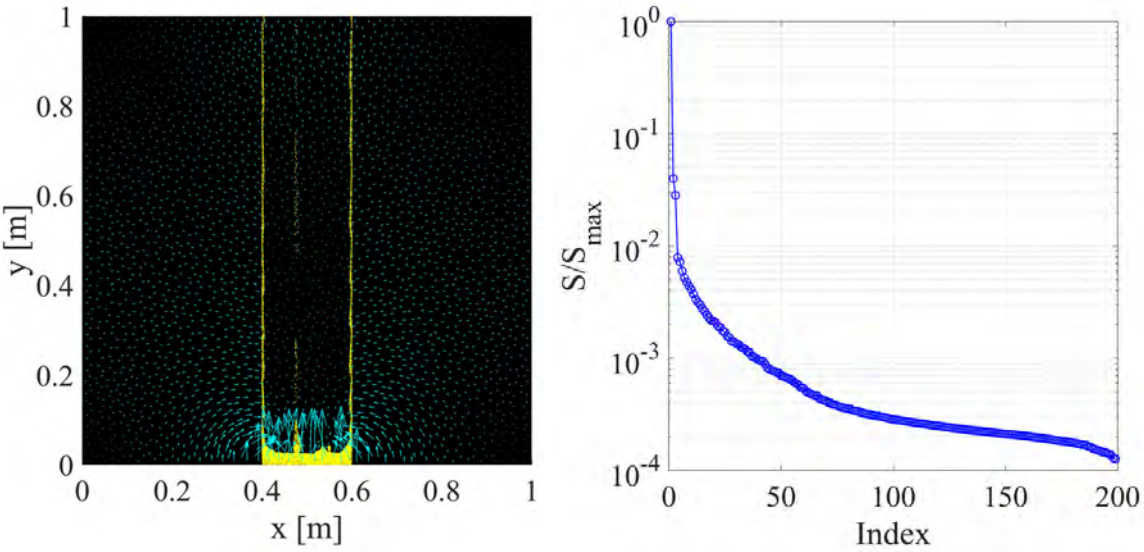


Figure 5.27: Comparison between full-order and DMD predicted average velocity and particle density at $n = 76000$ along the x-direction. Relative error is similarly defined as in Fig. 5.26. (a) x-directional average velocity (left axis) and relative error (right axis) plot. (b) Particle density variation along x-direction (left axis) and relative error plot (right axis). The missing points in the error graph are below the log scale range shown.

5.2.3 Electron Beam with Virtual Cathode Formation

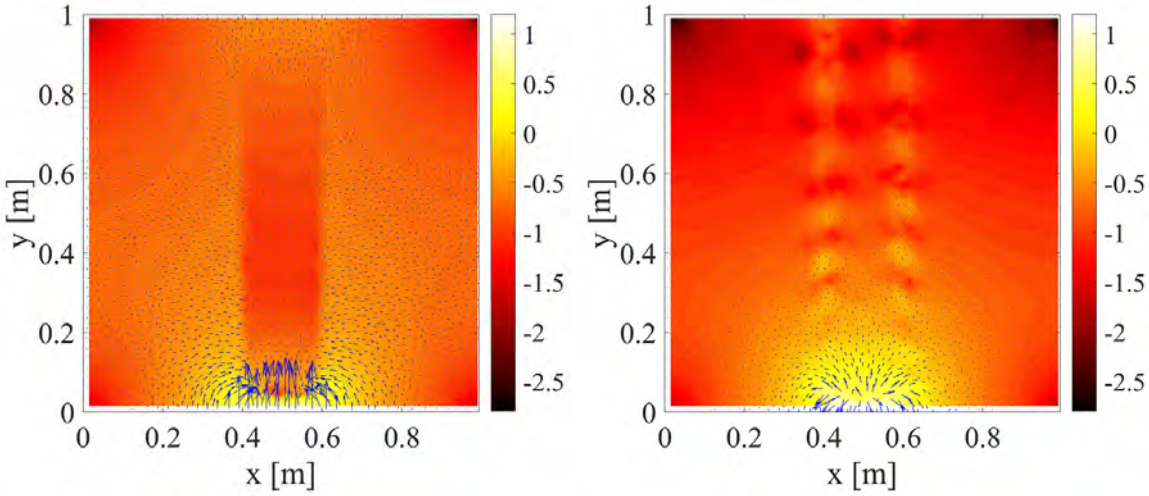
A relatively complex example of interest is the reduced-order modelling of virtual cathode oscillations. The setup of 5.2.2 is adopted with two major differences: (i.) the amount of injected current is increased 15 times, and, (ii.) a y-directional non-oscillating confining magnetic flux is employed instead of a transverse oscillating magnetic flux. The superparticle ratio is increased to 3×10^6, while holding the same injection rate. The external voltage bias is turned off and a strong magnetic flux, $B = B_y \hat{y}$ is applied in the y direction, with $B_y = 100$ A/m. The increased current

injection initiates virtual cathode formation, eventually leading to small oscillations near the root of the beam in the equilibrium state (Fig. 5.28a). The data set spans from time step $n = 80$ to $n = 160000$, containing a total of 2000 data points (stored at every 80^{th} time step), with $\Delta t = 0.02$ ns. Unlike the previous examples, we only discuss the key takeaways from DMD analysis for this problem.



(a) (b)

Figure 5.28: (a) Snapshot of virtual cathode formation for 2-D electron beam at $n = 80000$. The cyan arrows show the self electric field lines. (b) Normalized singular values for DMD in equilibrium region.



(a) $\phi_1^{(eq)}$ (b) $\phi_2^{(eq)}$

Figure 5.29: Dominant modes extracted from equilibrium region of virtual cathode formation.

DMD in Equilibrium State

The harvesting window spans from $n = 88000$ to $n = 120000$ with $r = 71$ and 37 DMD modes, although only two dominant modes capture more than 99% of the total energy in equilibrium. Exponential decay in singular values (Fig. 5.28b) reveals the underlying low-dimensional structure in equilibrium dynamics. The stationary structure of the virtual cathode is represented by the mode $\boldsymbol{\phi}_1^{(eq)}$ and the small oscillations at the location of virtual cathode formation are captured by $\boldsymbol{\phi}_2^{(eq)}$. The relative 2-norm error remains close to 1% inside the harvesting window and oscillates around 5% margin in the extrapolation region.

Predicted Particle Dynamics

We apply the sliding-window DMD method on self electric field data from the virtual cathode, with $\beta_{thr} = 0.01$, $\delta t_w = 6.4$ ns, $\Delta t_w = 160$ ns and $\Delta_t = 3.2$ ns. Equilibrium is detected at $k = 180$ ($n_{en}(180) = 65360$), at which point the field-update is replaced with extrapolated self-field values from DMD. The predicted particle dynamics at $n = 128000$ is shown in Figs. 5.30-5.33.

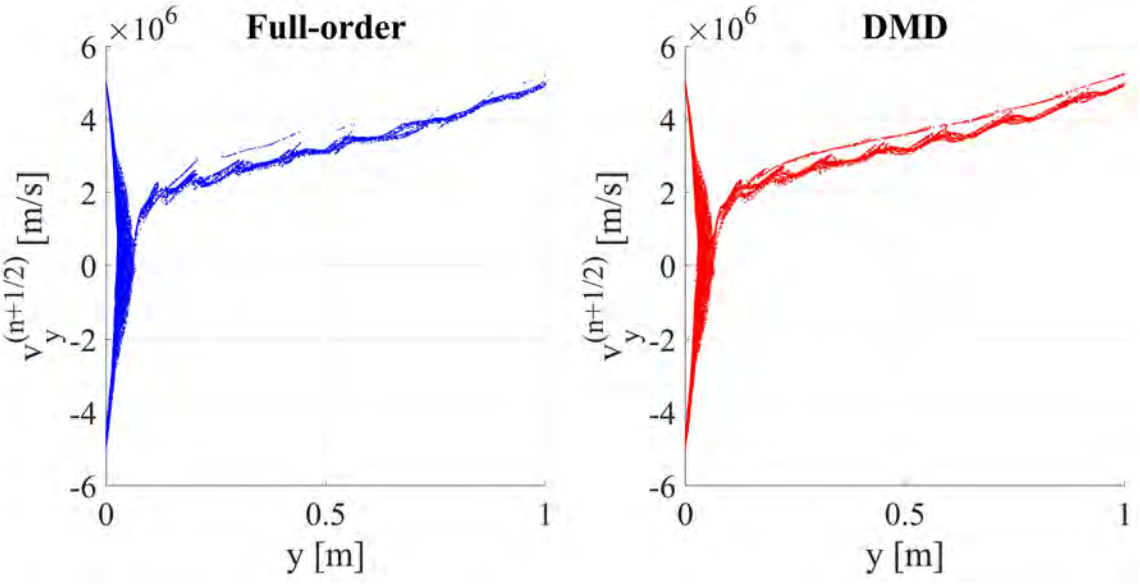


Figure 5.30: The y-directional phase-space plot comparison between finite-element full-order EMPIC simulation (blue) and DMD (red) in extrapolation region ($n = 128000$).

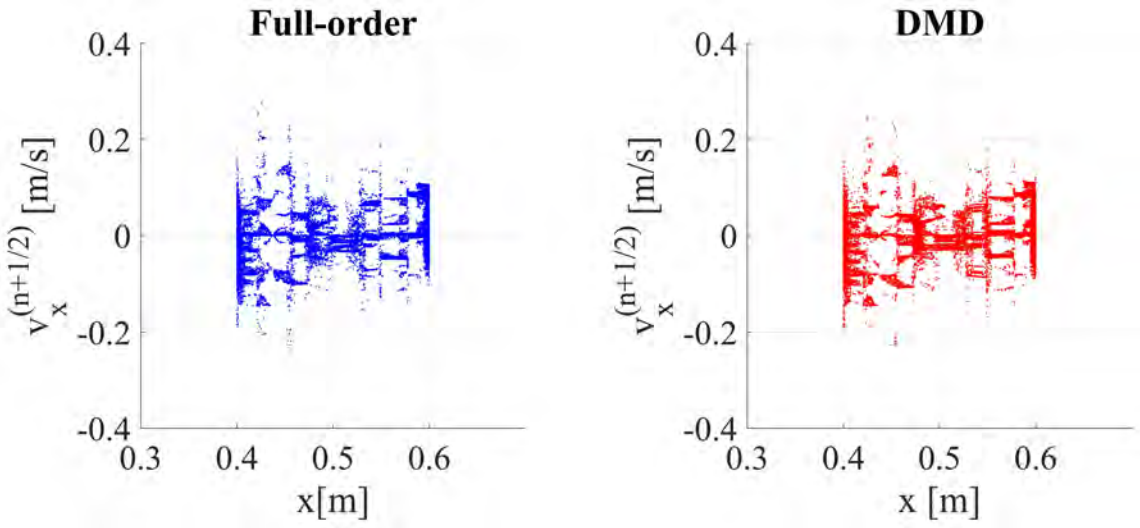


Figure 5.31: The x-directional phase-space plot comparison between finite-element full-order EMPIC simulation (blue) and DMD (red) in extrapolation region ($n = 128000$).

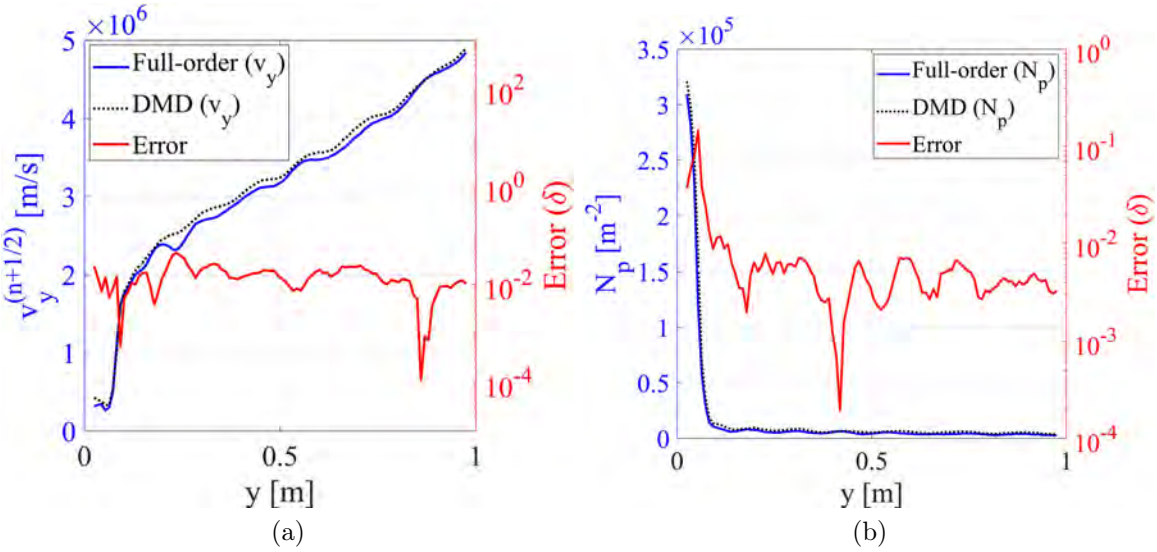


Figure 5.32: Comparison between full-order and DMD predicted average velocity and particle density at $n = 128000$ along the y-direction. Relative error in $v_y^{(n+1/2)}$ and N_p are as defined in Fig. 5.26. (a) y-directional average velocity plot and relative error. (b) Particle density variation along the y-direction and relative error.

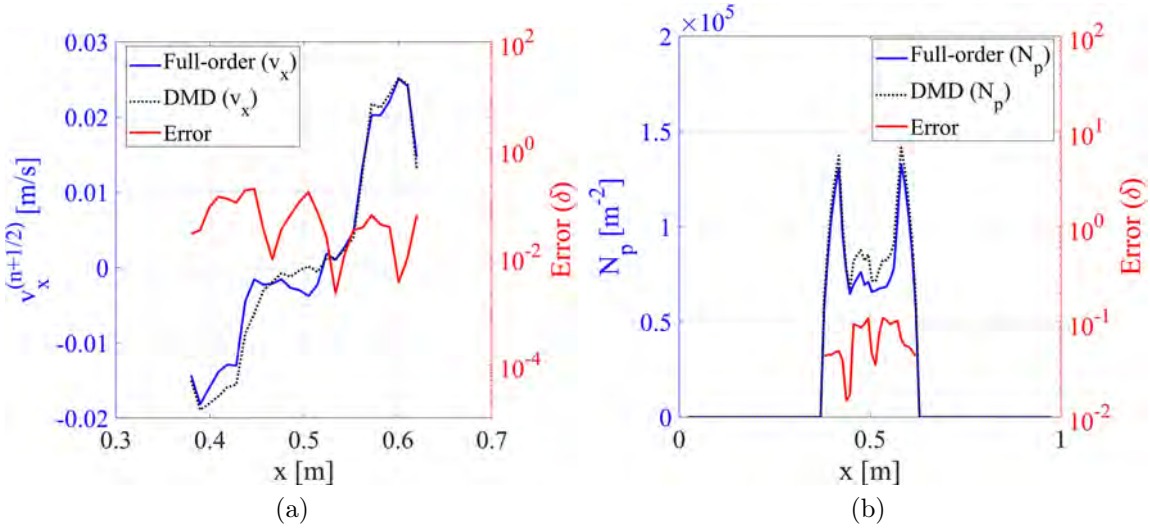


Figure 5.33: Comparison between full-order and DMD predicted average velocity and particle density at $n = 128000$ along the x-direction. (a) x-directional average velocity plot and relative error. (b) Particle density variation along the x-direction and relative error.

5.3 Computational Complexity

The time step complexity (runtime computational complexity to evolve through one time step) in our *explicit* particle-in-cell algorithm is $\mathcal{O}(N_p + N)$ [152] where N_p is the number of particles and N represents aggregate mesh dimension. For *implicit* field solvers the time step complexity[5] is $\mathcal{O}(N_p + N^s)$, with $s \geq 1.5$. We will use $\mathcal{O}(N^s)$ to denote the complexity of field solver with $s = 1$ for explicit solver, and $s > 1.5$ for implicit solver.

[5]The time step complexity of our solver is reduced by employing a sparse approximate inverse of the finite-element mass matrix in the time stepping procedure [133, 143]. This strategy basically trades the reduction in time step complexity for the one-time cost (incurred prior to time stepping) of computing the sparse approximate inverse.

Usually, $N_p \gg N$ and therefore the field gather, particle push, and current scatter stages represent the main bottleneck, especially in serial computers. On the other hand, in parallel computers, one can exploit the fact that the particle steps are embarassingly parallelizable. Nevertheless, in large problems with millions of grid nodes and edges, the field update can also consume significant amount of time. DMD based reduced-order models for self-fields can reduce this cost for long term predictions. In addition, EMPIC simulations are often run beyond the equilibrium onset (which is not known a priori). Let this post-equilibrium time step upto which we want tp query the solution be denoted as n_q. The runtime of a typical EMPIC simulation up to time step n_q in a serial computer is then $\mathcal{O}(n_q N_p + n_q N^s)$. The runtime complexity of exact DMD is dominated by the SVD step, given by $\mathcal{O}(l^2 N)$, where l is the number of DMD snapshots. For Hankel DMD, effective dimension is Nd and number of snapshots is typcally $l_d = l - d$, leading to complexity of $\mathcal{O}(l_d^2 Nd)$. Note that for the sliding-window DMD method, the DMD features can be calculated in an incremental manner. However, let us assume the wort case scenario, i.e. the usual DMD calculations are performed for each sliding window, and shift is by one time snapshot ($\delta n_w = \Delta_n$). If the equilibrium onset is detected at n_{eq}, the runtime complexity of equilibrium detection is $\mathcal{O}(\frac{l_d^2 N d n_{eq}}{\Delta_n})$. Here Δ_n represents the number of time steps between two consecutive DMD snapshots. The resulting overall computational complexity of the sliding-window DMD is thus $\mathcal{O}(\frac{l_d^2 N d n_{eq}}{\Delta_n} + n_{eq} N_p + n_{eq} N^s)$. Consequently, if we are only interested in self-fields, the presented

method is advantageous compared to EMPIC ($\mathcal{O}(n_q N_p + n_q N^s)$) for $N_p \gg N$ and/or $n_q \gg n_{eq}$.

If the particle dynamics at n_q is also sought, then the reduced-order model for self-fields also provides some advantages given $n_q \gg n_{eq}$ since the field solver is obviated beyond n_{eq}. The overall computation complexity becomes $\mathcal{O}(\frac{l_d^2 N d n_{eq}}{\Delta_n} + n_q N_p + n_{eq} N^s)$ compared to the original cost of $\mathcal{O}(n_q N_p + n_q N^s)$. If $N_p \gg N$, it turns out that the computation advantage is insignificant. However, if N_p and N are comparable, then the sliding-window DMD model is advantageous for $n_q \gg n_{eq}$.

For simplicity, the above estimates assume a serial implementation. As noted, in parallel computers, one can readily exploit the fact that all particle steps (gather, pusher, and scatter) are embarrassingly parallelizable. In that case, the runtime estimates would of course depend on other factors such as the number of available processors.

5.4 Summary

In this section, we delved into the nonlinear dynamics of plasma. We examined the effectiveness of DMD in modeling self-fields within EMPIC simulations. Additionally, we evaluated the quality of fields predicted by DMD by analyzing their impact on particle dynamics. Our exploration covered plasma ball expansion, oscillating electron beams, and virtual cathode oscillators. We also demonstrated how transient DMD features evolve into equilibrium features, supporting our hypothesis behind DMD mode tracking. This approach can significantly reduce simulation time

when the focus is solely on self-fields. However, a notable drawback is that DMD-predicted fields do not inherently adhere to the underlying physical laws, making it challenging to provide performance guarantees.

Chapter 6: Accelerating Particle-in-Cell Plasma Simulations via Reduced-Order Modeling of Space-Charge Dynamics using Dynamic Mode Decomposition

In this chapter, we demonstrate the application of DMD in current density forecasting to accelerate EMPIC simulations. We demonstrate an interpretable, reduced-order model for the space-charge dynamics in EMPIC simulations. This is achieved by DMD modeling of the current density which is essentially the manifestation of charged particle dynamics. While our main goal is to accelerate the EMPIC simulations, such reduced-order modeling of space-charge dynamics also helps analyze and diagnose the problem at hand. Note that using DMD-based linear reduced-order models (ROMs) of the inherently nonlinear plasma dynamics also opens the doors for leveraging control theoretic tools that already exist for linear systems. The DMD modes and frequencies can help in analyzing the efficiency loss due to harmonic generation in high-power microwave devices. By analyzing the spatial patterns and growth rates of the modes, DMD can help predict areas that are prone to decay or damage due to high energy densities, heating, or other factors. DMD can also

identify unstable modes and flaws in device design or the numerical solver itself. Furthermore, understanding the dominant modes and their characteristics can help in optimize the design parameters of such devices. A novel DMD-EMPIC algorithm (Fig. 6.1) is presented to accelerate the EMPIC simulations showing post-transient behavior, i.e. either steady-state, equilibrium, or any type of periodic behavior. The time-domain DMD model of the current density implements rapid prediction of the current density at any time instant, and thus eliminates the need for EMPIC stages involving particles. The DMD-EMPIC strategy utilizes the on-the-fly algorithm discussed in Chapter 3 to detect the end of transience in real-time. It then replaces the computationally expensive *gather, pusher* and *scatter* with DMD predicted current beyond that point in time. It is important to highlight that the key distinction between this work and previous research, as outlined in [154], lies in the application of the DMD model to current density for replacing EMPIC stages involving particles. This approach not only facilitates the prediction of future current density but also enables fast and accurate forecasting of self-field values, adhering precisely to the discrete Maxwell's equations. Moreover, this study employs a more versatile on-the-fly algorithm, distinguishing it from the method utilized in [154]. DMD-EMPIC has the potential to significantly expedite the EMPIC simulations for plasma systems showing long oscillations (e.g. limit-cycle behavior).

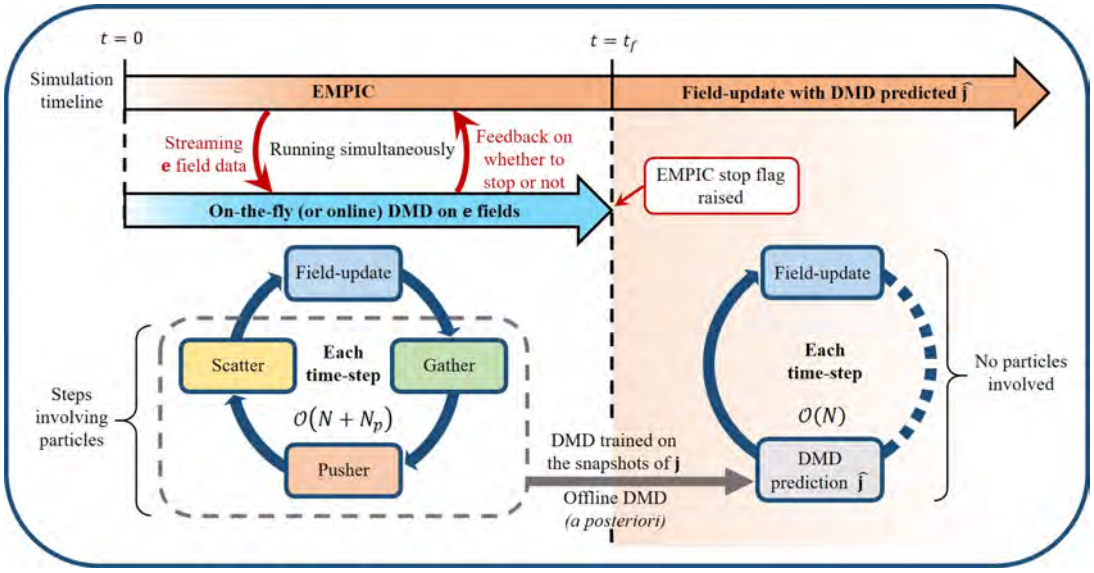


Figure 6.1: DMD-EMPIC algorithm for accelerating EMPIC simulations. Note that for $t > t_f$, the stages are not exactly cyclic since fields no longer have any effect on the DMD predicted current density (illustrated by the broken line).

6.1 DMD-EMPIC Algorithm

As described in Section 5.3, the primary computational bottleneck comes from the *gather*, *pusher* and *scatter* stages as they involve each and every particle in the solution domain. However, these steps are necessary as they dictate the time evolution of the current density which subsequently helps update the electromagnetic fields. We try to address this issue by directly modeling the time evolution of current density $\mathbf{j}$ using DMD. An analytical equation of the type (2.24) for time variation of current density in EMPIC serves the following purpose,

- DMD helps in rapid prediction of **j** with negligible computation cost compared to the particle operations. Also, a linear time-evolution model for current density can facilitate control theory applications.

- The *gather*, *pusher* and *scatter* stages can be replaced by the DMD predicted **j** which is then used to update the electric and magnetic fields in the *field-update* stage (Section 5.1.2) within the finite-element time-domain (FETD) setting. This can drastically reduce the computation cost of EMPIC.

The DMD-EMPIC algorithm is illustrated in Fig. 6.1. It consists of two main phases, the transient ($t \leq t_f$) and post-transient phase ($t > t_f$).

- **Transient phase ($t \leq t_f$):** The high-fidelity EMPIC simulation is run until the transient phase ends, denoted by the final time t_f. The on-the-fly DMD algorithm [50] is run simultaneously with the ongoing EMPIC simulation to identify t_f on the fly. The electric field data ($\mathbf{e}$) is fed to the Hankel DMD algorithm and it provides feedback in real-time regarding whether to stop the EMPIC simulation or not. In order to maximize computational gains, it is desirable to terminate the time-consuming EMPIC simulation at the earliest opportunity. At the same time, if the simulation is terminated too early (i.e. before the transient ends), DMD will not be able to make accurate time-extrapolation due to the lack of quality training data [154]. Thus, a real-time algorithm for timely termination of high-fidelity EMPIC simulation is necessary. In the DMD-EMPIC algorithm, this is achieved by the sliding-window

on-the-fly DMD algorithm developed in [50] which analyzes the time evolution of $\mathbf{e}$ in order to detect the end of transience at $t = t_f$. However, in order to handle the repetitive execution of DMD for large datasets, we make modifications to the algorithm described in [50]. Instead of using the standard version of DMD, we perform randomized DMD to reduce the computational load. The on-the-fly algorithm is described in Algorithm 3. [6]

- **Post-transient phase ($t > t_f$):** Following the detection of end-of-transience (t_f), Hankel DMD is performed in offline or *a posteriori* fashion on the snapshots of $\mathbf{j}$ collected inside the final DMD window. The purpose of the offline DMD is to predict the current density beyond t_f, denoted by $\hat{\mathbf{j}}$. Typically, the *gather*, *pusher* and *scatter* steps are required for the time-update of $\mathbf{j}$. Here, since an analytical expression (similar to (2.23)) for the time evolution of $\mathbf{j}$ is available from the offline DMD, we avoid these steps by using the DMD predicted $\hat{\mathbf{j}}$. The predicted current $\hat{\mathbf{j}}$ is then used in consecutive time steps to update the self electric and magnetic fields in the *field-update* stage (Section 5.1.2). Note that for $t > t_f$, the relation between the self-fields and current density is not exactly cyclic. Beyond $t = t_f$, the fields do not have any effect on the time-evolution of the current density $\hat{\mathbf{j}}$.

As shown in Fig. 6.1, before the end of transience is detected ($t < t_f$), the computation cost for each time step of the simulation is same as the cost of typical

[6]Note that we use a slight variant of Algorithm 3, a slightly different convergence criterion [52].

EMPIC ($\mathcal{O}(N + N_p)$) with the added cost of on-the-fly DMD, where N is aggregate mesh dimension and N_p is the total number of charged particles. However, beyond t_f the computation cost per time step reduces to $\mathcal{O}(N)$, which is a significant reduction in computation time given that $N_p \gg N$ in typical EMPIC settings [7]. The computational gain of the proposed DMD-EMPIC is discussed in details in Section 6.3.

6.2 DMD Applied to EMPIC Current Density

6.2.1 Oscillating Electron Beam

A 2-D electron beam propagating along the positive y direction and oscillating under the influence of an external transverse magnetic flux is shown in Fig. 6.2a. The solution domain (xy plane) is a square cavity of dimension 1 cm $\times$ 1 cm, which is discretized via an unstructured mesh composed of triangular elements. The mesh consists of $N_0 = 844$ nodes, $N_1 = 2447$ edges and $N_2 = 1604$ elements (triangles). Superparticles (blue dots in Fig. 6.2a), are injected at the bottom of the cavity in the $+y$ direction with a velocity of 5×10^6 m/s. The superparticles are injected at the rate of 50 per time step in a random fashion uniformly in the range [0.45 cm, 0.55 cm]. The superparticles discretize the phase-space of the electrons assuming a delta distribution in both position and velocity space. Superparticles are treated as point

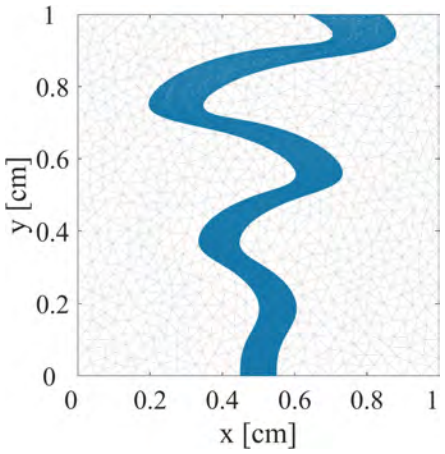
[7]Note that the contemporary EMPIC algorithms such as geometric electromagnetic particle-n-cell algorithm or GEMPIC [15] and conformal EMPIC algorithms [155] involves the "grid to particle" and "particle to grid" interpolation stages along with the particle pusher stage. Our DMD-EMPIC framework can be applied to these algorithms as well replacing the particle stages with DMD model of current density.

charges with mass $m_{sp} = r_{sp} m_e$ and charge $q_{sp} = r_{sp} q_e$, where m_e and q_e are respectively the mass and charge of an electron, and $r_{sp} = 5000$ is the number of actual electrons represented by each superparticle (superparticle ratio). An external oscillating magnetic flux $\mathcal{B}_{ext} = \mathcal{B}_0 \sin(2\pi/T_b)\ \hat{z}$ is applied in the z-direction, where $\mathcal{B}_0 = 2.5 \times 10^{-2}$ T, and $T_b = 0.8$ ns. The simulation is run until $n = 320000$ time steps or $t = 64$ ns with the time step interval $\Delta t = 0.2$ ps.

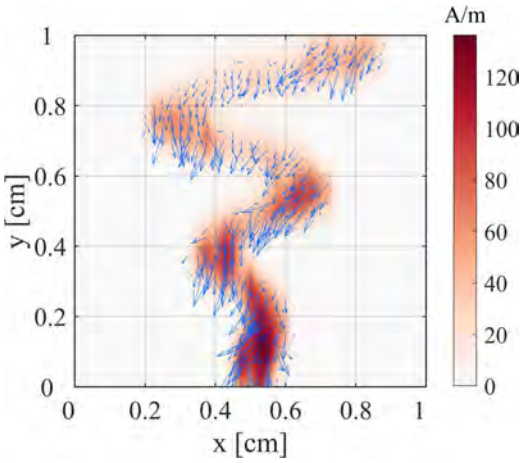
The post-transient snapshot of the current density $\mathbf{j}$ is shown in Fig. 6.2b at $t = 16$ ns ($n = 80000$). The goal is to model the time evolution of such snapshots inside the cavity using DMD. Unlike electromagnetic fields, $\mathbf{j}$ is restricted to only the mesh elements interacting with the particles. In other words, the number of active edges N_{1a} over which $\mathbf{j}$ is nonzero (within the DMD window span) is less than total number of mesh edges N_1. We only consider those active edges for DMD modeling with $\mathbf{x} = \mathbf{j}_a$, where $\mathbf{j}_a$ is the current density vector with active edges. After performing DMD, we revert back to the original state space with zero padding.

On-the-fly DMD on fields

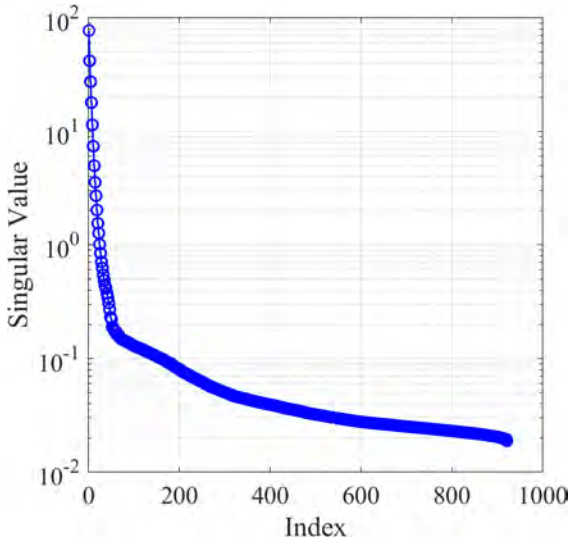
The on-the-fly DMD is carried out on the electric field data to detect the end of transience or onset of the periodic behavior as described in Appendix 3.2 [50]. Approximate prior knowledge about the time-scale is required to choose the DMD window width Δt_w accordingly, ensuring that it covers multiple oscillation cycles. We select $\Delta t_w = 8$ ns. Once the EMPIC simulation reaches $t = \Delta t_w$, fast Fourier



(a) Electron beam snapshot at $t = 64$ ns.



(b) Current density snapshot at $t = 64$ ns.



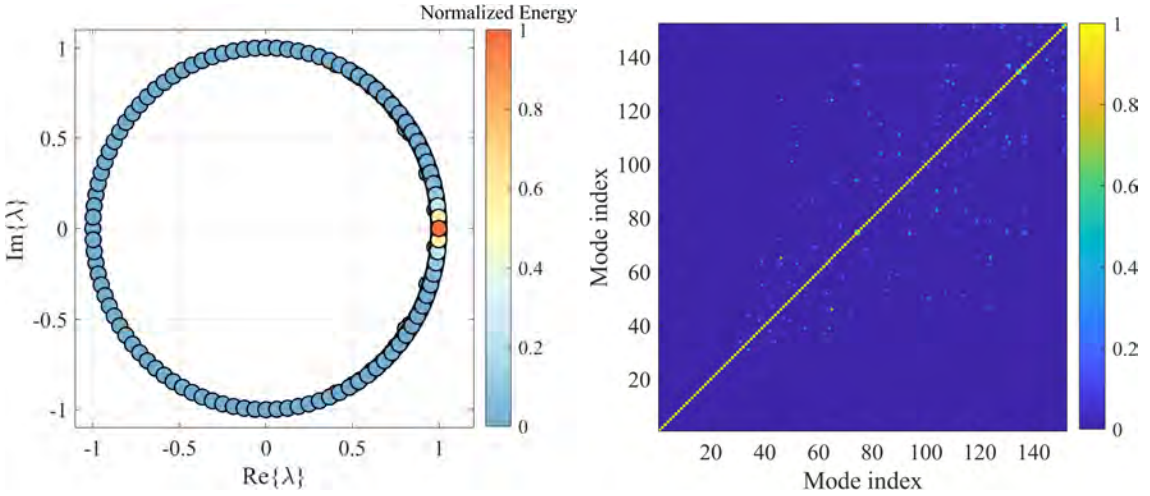
(c) Singular values for oscillating electron beam.

Figure 6.2: (a) Snapshot of the electron beam at $t = 64$ ns. The beam is propagating along the $+y$ direction, and oscillating under the influence of a z-directed transverse magnetic flux. The blue dots represent superparticles and grey lines show the finite element mesh edges. (b) Snapshot of the current density at $t = 16$ ns. The magnitude is shown by a colormap [1], whereas the direction is denoted by the blue arrows. Note that in all the current density plots, the magnitude colormap is smoothed for visualization purpose. (c) Singular values after performing SVD on the snapshots of current density.

Table 6.1: DMD parameters for modeling current density.

Parameters	**Osc. Beam**	**Vircator**	**BWO**
t_{st}	15.60 ns	22.40 ns	91.20 ns
t_{en}	23.60 ns	30.40 ns	107.20 ns
Δt_w	8 ns	8 ns	16 ns
Δ_t	8 ps	40 ps	2 ps
d	80	50	20
r	302	42	1999
M	152	22	1006

transform (FFT) is performed on a randomly chosen set of 20 points in space. Averaged FFT allows us to select the DMD sampling interval $\Delta_t = 8$ ps. We select the target rank $r = 200$, and number of Hankel stacks $d = 10$. The shift in consecutive sliding DMD windows is $\delta t_w = 0.4$ ns. The on-the-fly DMD parameters are mentioned in details in the Table 6.2. on The onset of equilibrium (end of transience) is detected at $t = t_f = 23.06$ ns.



(a) DMD eigenvalues. (b) Spatial correlation among DMD modes.

Figure 6.3: (a) The DMD eigenvalues on the complex plane wrt. the unit circle. The DMD eigenvalues are color-mapped according to their normalized energy. (b) Spatial correlation (ρ) among different DMD modes.

Offline DMD on current density

The primary contribution of this chapter is to accelerate EMPIC simulations by DMD modeling of the plasma current density. EMPIC stops at $t = t_f$ (detected end of transience), and offline DMD on current density $\mathbf{j}$ is performed in the window $t \in [t_f - \Delta t_w, t_f]$ for time-extrapolation. We first identify the active edges to construct $\mathbf{j}_a$, perform DMD on the snapshots of $\mathbf{j}_a$ to get the predictions $\hat{\mathbf{j}}_a$, and then revert back to $\hat{\mathbf{j}}$. The DMD parameters are summarized in Table 6.1. The t_{st}, t_{en}, i.e. the location of the DMD window for current density is already determined from the on-the-fly DMD with $t_{st} = t_f - \Delta t_w$ and $t_{en} = t_f$. FFT is performed in $[t_{st}, t_{en}]$

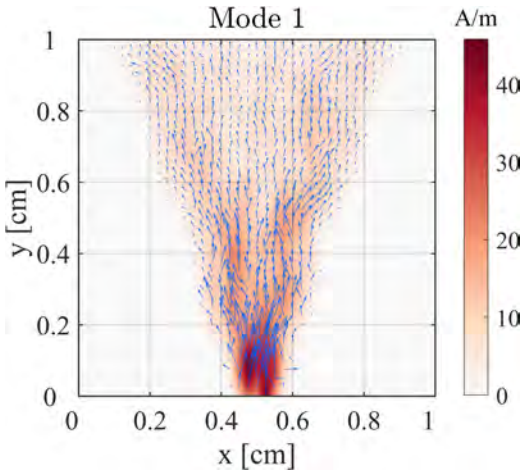
to determine the DMD sampling interval Δ_t. As a rule of thumb, we choose DMD sampling frequency to be four times the Nyquist frequency.

The sharp decay in the singular values (Fig. 6.2c) indicates the existence of a low-dimensional structure in the plasma current dynamics. Fig. 6.4 shows the first four most energetic DMD modes and corresponding DMD eigenvalues, as well as the spatial cross-correlation matrix. The DMD modes are indexed according to their energy [8], with mode 1 (Fig. 6.4a) being the most energetic one which is essentially a DC mode. Mode 2 (Fig. 6.4b) captures the oscillation frequency of the external magnetic flux with DMD frequency of $f_2 = 1.25$ GHz ($f_m = |\frac{\mathrm{Im}\{\omega_m\}}{2\pi}|$). Mode 3 (Fig. 6.4c) indicates the first harmonic. Together these two modes capture $> 90\%$ of the total energy (Fig. 6.3a). As the mode index increases, the spatial pattern becomes less structured due to the effect of numerical noise. The frequencies associated with mode 3 (Fig. 6.4c) and mode 4 (Fig. 6.4d) indicate that those are essentially the harmonics of mode 2, generated due to the nonlinear wave-particle interaction. The correlation ρ between different spatial patterns of DMD modes indicates their extent of orthogonality. We use the absolute value of modal assurance criterion (MAC) [74, 154] to compute the spatial correlation among DMD modes,

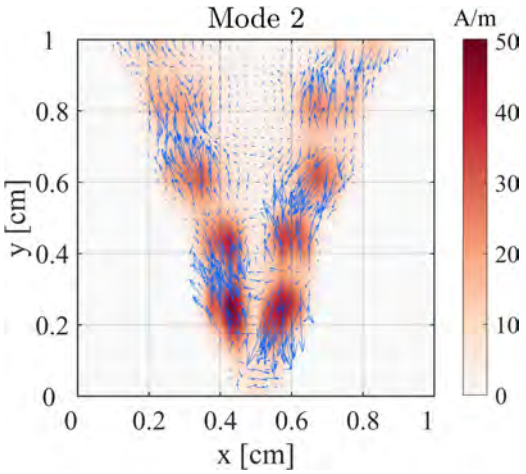
$$\rho(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j) = |\mathrm{MAC}(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j)| = |\frac{\left|\boldsymbol{\psi}_i^{\mathrm{T}}\ \overline{\boldsymbol{\psi}}_j\right|^2}{(\boldsymbol{\psi}_i^{\mathrm{T}}\overline{\boldsymbol{\psi}}_i)\cdot(\boldsymbol{\psi}_j^{\mathrm{T}}\overline{\boldsymbol{\psi}}_j)}|. \tag{6.1}$$

We reiterate that by m^{th} DMD mode, we refer to the $\{\boldsymbol{\phi}_m, \overline{\boldsymbol{\phi}}_m\}$ pair in (2.24). While plotting the modes (Fig. 6.4) as well as calculating ρ, we use $\boldsymbol{\psi}_m = (\boldsymbol{\phi}_m +$

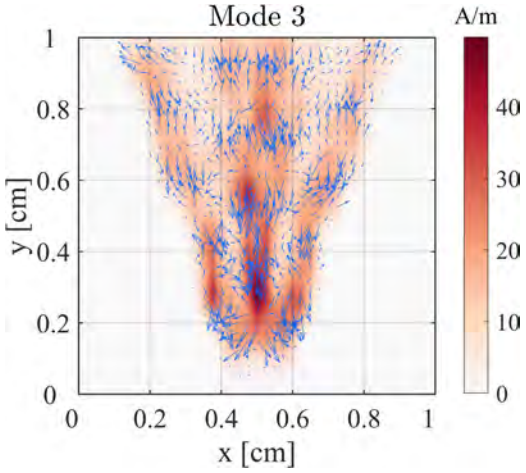
[8]The energy of a DMD mode is calculated by averaging the 2-norm square of that particular DMD mode over all the time-samples inside the DMD training window.



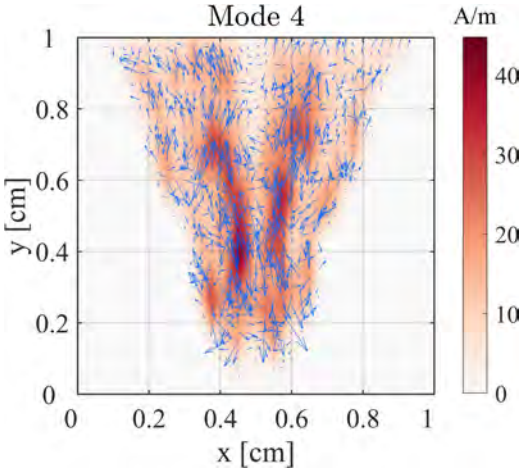
(a) DMD mode 1 with $f_1 = 0$ (DC).



(b) DMD Mode 2 with $f_2 = 1.25$ GHz.



(c) DMD Mode 3 with $f_3 = 2.50$ GHz.



(d) DMD mode 4 with $f_4 = 3.75$ GHz.

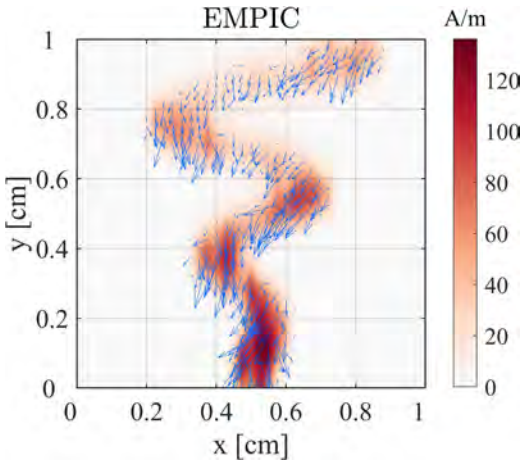
Figure 6.4: Extracted DMD features for oscillating electron beam. (a,b,c,d) The DMD modes ($\boldsymbol{\phi}_m + \overline{\boldsymbol{\phi}}_m$) and their corresponding frequencies (f_m) for the current density modeling.

$\overline{\boldsymbol{\phi}}_m) = 2\text{Re}\{\boldsymbol{\phi}_m\}$, where $\text{Re}\{\cdot\}$ represents the real part. The spatial correlation matrix shows that the dominant DMD modes are orthogonal to each other with off-diagonal elements close to zero. However, it is important to note that unlike POD, DMD does not ensure orthogonality in space, but guarantees orthogonality in time.

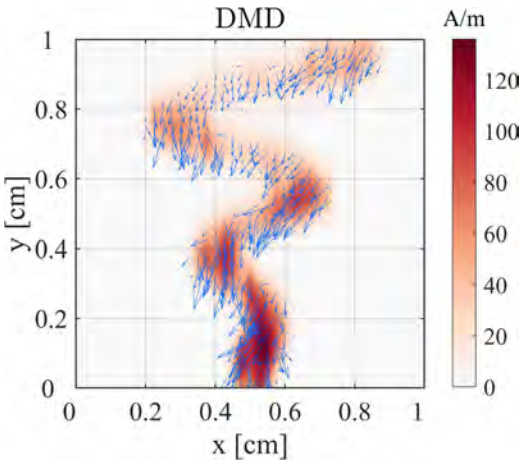
The DMD predicted current density (Fig. 6.5b) deep into the prediction (extrapolation) region is plotted against the current density from high-fidelity EMPIC simulation (Fig. 6.5a) for side-to-side comparison. The relative error given by (6.2), is also plotted in Fig. 6.5c.

$$\delta(t) = \frac{||\hat{\mathbf{j}}(t) - \mathbf{j}(t)||_2}{||\mathbf{j}(t)||_2}, \tag{6.2}$$

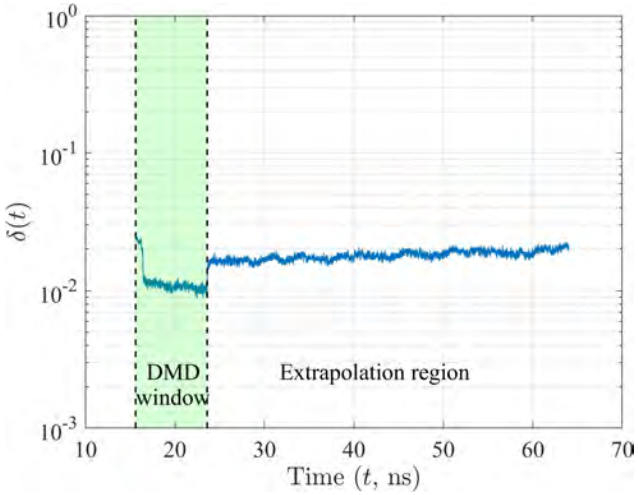
where $||\cdot||_2$ indicates the 2-norm. The average relative error in the extrapolation region is 1.80%. The *gather, pusher* and *scatter* stages are replaced by the DMD prediction $\hat{\mathbf{j}}$ for $t > t_f$ ($\equiv n > n_f$) in the EMPIC simulation, as illustrated in Fig. 6.1. The self-fields $\mathbf{e}$ and $\mathbf{b}$ generated beyond t_f is compared against the self-fields generated from EMPIC simulation. The relative error is calculated in a similar manner as in (6.2). The electric field patterns from EMPIC and DMD-EMPIC in the extrapolation region at $t = 64$ ns are shown in Fig. 6.6a, and Fig. 6.6b showing good agreement. The relative errors in the self electric and magnetic field are 0.94% and 2.60% respectively (Fig. 6.6c). The gain in runtime is discussed in Section 6.3.



(a) EMPIC simulation of current density at $t = 64$ ns.

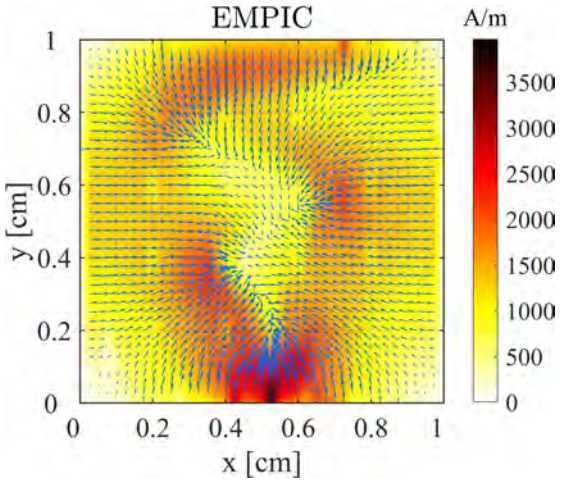


(b) DMD prediction of current density at $t = 64$ ns.

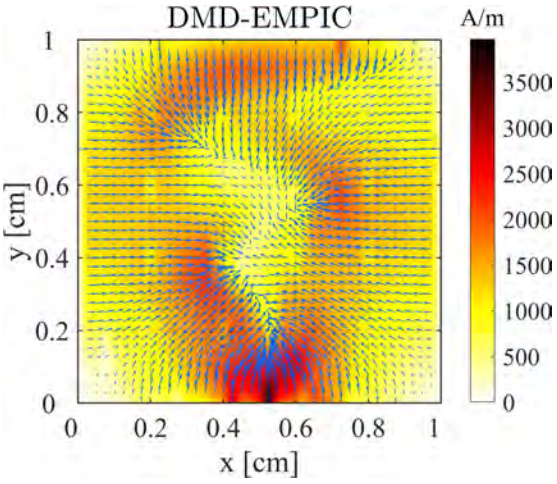


(c) Relative error between EMPIC and DMD predicted current density according to (6.2).

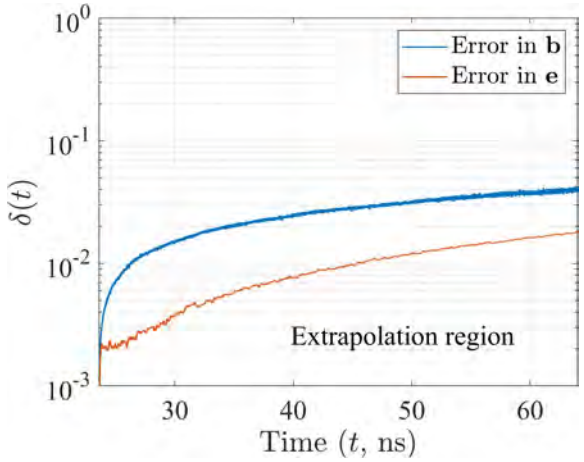
Figure 6.5: Comparison between EMPIC and DMD predicted current density. The shaded green region in (c) denotes the DMD training window. Note that the gap at the end of DMD window is due to the time-delayed stacking.



(a) EMPIC simulation of electric field at $t = 64$ ns.



(b) DMD-EMPIC simulation of electric field at $t = 64$ ns.



(c) Relative errors between EMPIC and DMD-EMPIC simulated self electric (**e**) and magnetic (**b**) fields.

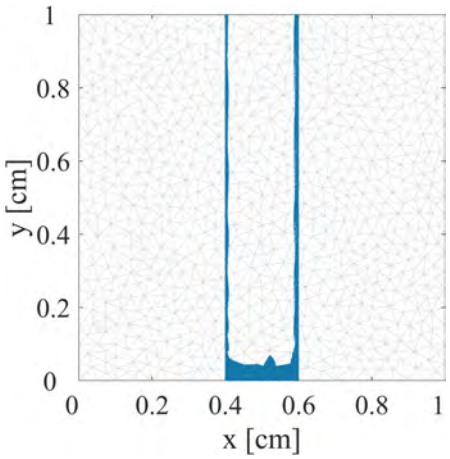
Figure 6.6: Comparison between EMPIC and DMD-EMPIC simulated self-fields in the extrapolation region.

6.2.2 Virtual Cathode Oscillations

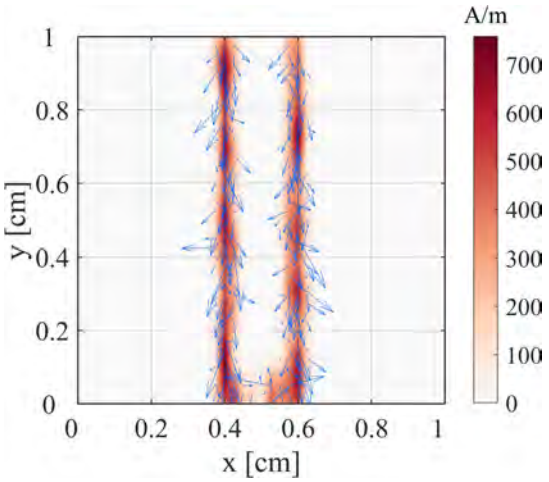
Next we consider a more challenging example of virtual cathode oscillation. We use the same setup as in Section 6.2.1, however with the following modifications. We increase the current injection by both increasing the superparticle ratio to $r_{sp} = 75000$ and the superparticle injection rate to 200 per time step. The superparticles are injected at the bottom in the region [0.4 cm, 0.6 cm]. Instead of a transverse oscillating magnetic flux, we apply a strong confining magnetic field, $\mathbf{B} = B_y\hat{y}$ along the y direction, with $B_y = 100$ T. The simulation is run until $n = 320000$ time steps or $t = 64$ ns with time step $\Delta t = 0.2$ ps. The snapshot of the beam after virtual cathode formation at $t = 16$ ns, and the corresponding current density plot are shown in Fig. 6.7a and Fig. 6.7b respectively. Modeling the current density for virtual cathode oscillations is particularly challenging because there are no external forces dictating a clear oscillation pattern of the electrons. The majority of oscillations are limited to a small region (near the bottom) causing possible rank deficiency, and the leakage from the sides makes variation of $\mathbf{j}$ more prone to the particle noise.

On-the-fly DMD on fields

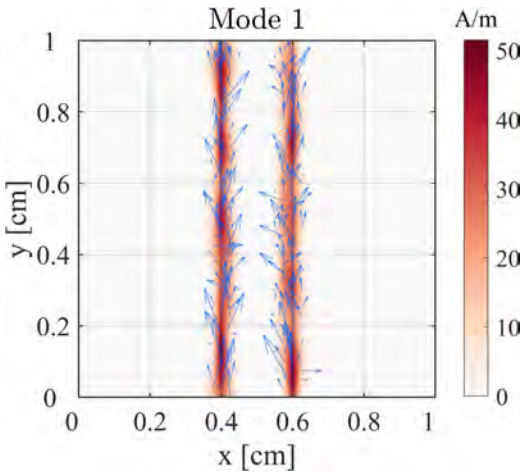
Similar to the oscillating beam case, the on-the-fly DMD is carried out to detect the end of transience indicating the onset of virtual cathode oscillations. We select $\Delta t_w = 8$ ns, $\Delta_t = 8$ ps, $\delta t_w = 0.4$ ns, $r = 200$, and number of Hankel stacks $d = 10$. The onset of the virtual cathode oscillations is detected at $t_f = 30.40$ ns. The on-the-fly DMD parameters are provided in details in Table 6.2.



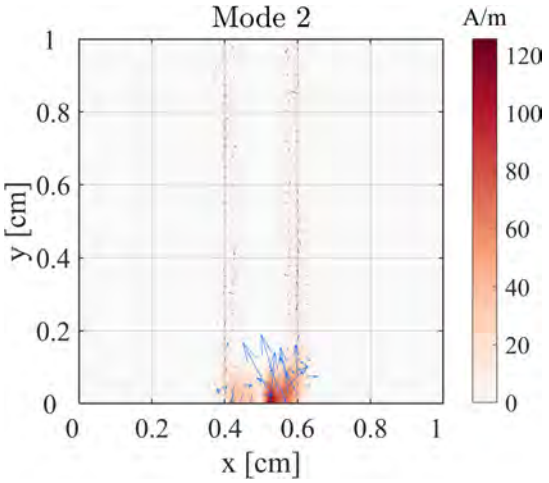
(a) Virtual cathode snapshot at $t = 64$ ns.



(b) Current density snapshot at $t = 64$ ns.

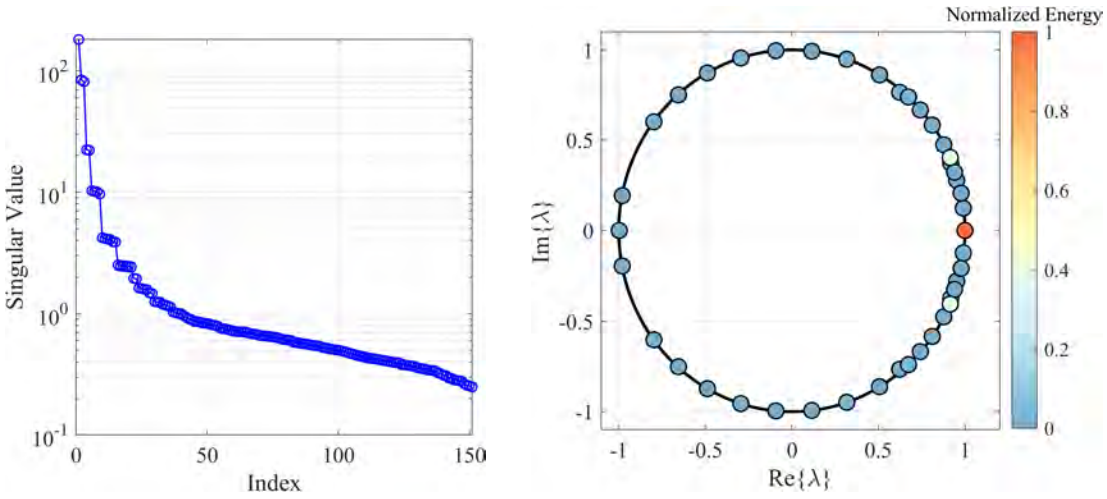


(c) DMD mode 1 with $f_1 = 0$ (DC).



(d) DMD Mode 2 with $f_2 = 1.66$ GHz.

Figure 6.7: (a) Snapshot of the virtual cathode formation at $t = 64$ ns. (b) Snapshot of the current density at $t = 64$ ns. (c,d) Extracted DMD modes for current density modeling in virtual cathode oscillations.



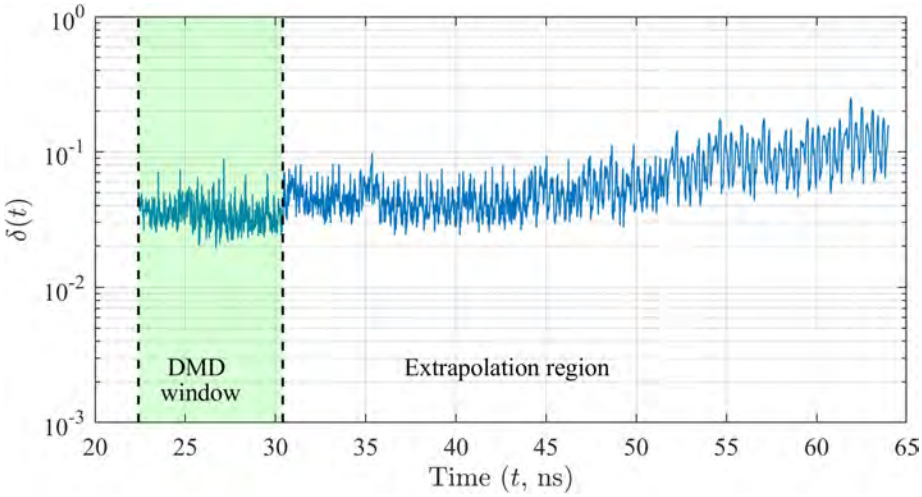
(a) Singular values for virtual cathode oscillations.

(b) DMD eigenvalues.

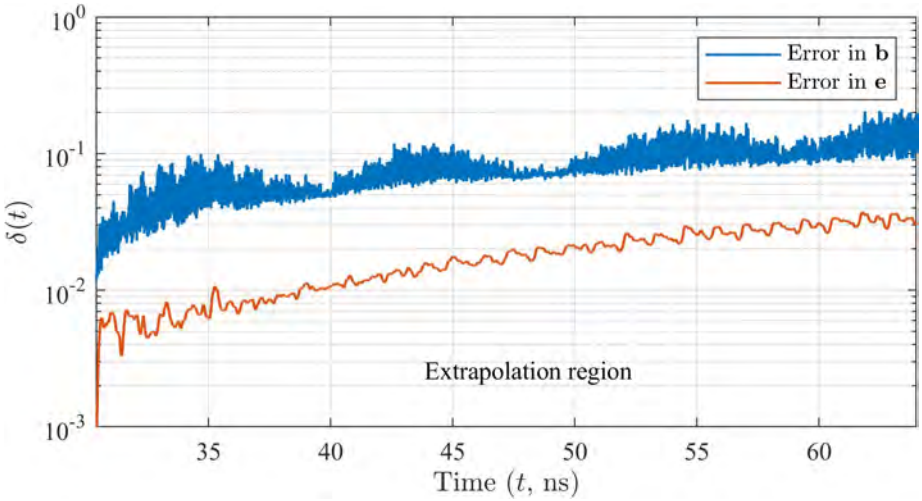
Figure 6.8: Extracted DMD modes and eigenvalues for current density modeling in virtual cathode oscillations.

Offline DMD on current density

The training parameters for the Offline DMD is summarized in Table 6.1. Fast decay of the singular values in Fig. 6.8a indicates that most of the energy is concentrated in small number of modes. First two most energetic DMD modes carry more than 95% of the total energy, and are plotted in Fig. 6.7. Mode 1 (Fig. 6.7c) represent the DC leakage current from sides of the virtual cathode, whereas mode 2 (Fig. 6.7d) captures the oscillations near the root of the beam, having frequency of 1.65 GHz. The DMD eigenvalues (Fig. 6.8b) also indicate the dominance of the DC mode in terms of energy.



(a) Relative error in DMD predicted current density.



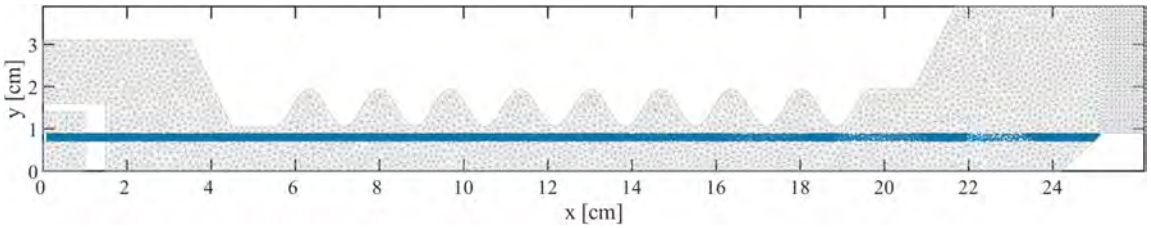
(b) Relative errors in the DMD-EMPIC simulated self-fields.

Figure 6.9: Relative errors for the virtual cathode oscillations.

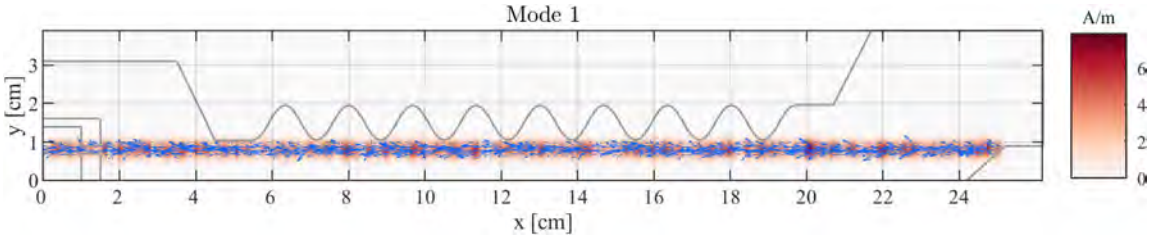
With the help of extracted DMD modes, frequencies and modal amplitudes, current density is extrapolated for $t > t_f$ using DMD. The relative error in predicted current is shown in Fig. 6.9a. The average error in extrapolation region around 6.65%, which is higher compared to the oscillating beam case (1.80%). This is expected because the leakage from both sides of the virtual cathode results in non-smooth current variation due to high-particle noise. Also, the localized nature of the oscillation contributes to a possible rank-deficiency resulting in higher error. The relative error in self electric field **e** and magnetic flux **b** is shown in Fig. 6.9b. The average relative error in **e** is around 1.81% whereas in **b** is around 8.10%. Error in **b** is typically higher for both the test-cases since self magnetic field is generally very low in magnitude, and more susceptible to particle and numerical noise.

6.2.3 Backward Wave Oscillator

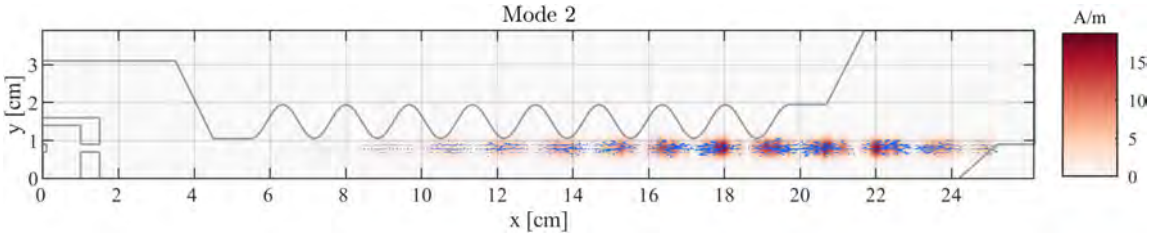
Now we consider a more challenging and practical 2.5D case of a backward wave oscillator (BWO) [156]. The finite element discretization of the longitudinl cross-section of a sinusoidally corrugated slow-wave structure (SCSWS) is depicted Fig. 6.10a. The SCWS has boundary profile $R(z) = \frac{1}{2}(A + B) + \frac{1}{2}(A - B)\cos(\frac{2\pi}{C}z)$. Based on an eigenmode analysis, the SCSWS is designed to have $A = 19.5$ mm, $B = 10.5$ mm, $C = 16.7$ mm, and $N_{\text{crg}} = 8.5$. Each superparticle in the EMPIC model represents $r_{sp} = 1.25 \times 10^8$ electrons, with injection rate of 20 superparticles per time step ($\Delta t = 0.5$ ps). The superparticles are injected in a random fashion with uniform distribution within the region centered around $y = 0.008$ m with beam



(a) Snapshot of electron beam inside the BWO at $t = 140$ ns.



(b) DMD mode 1 (DC mode)



(c) DMD mode 2 with $f_2 = 8.32$ GHz.

Figure 6.10: Offline DMD on current density of a backward wave oscillator (BWO).

width 0.0018 m. Superparticles are injected with a velocity of 2.5×10^8 m/s in the x-direction. The simulation is run for $440,001$ time steps or $t = 220.005$ ns. The fundamental frequency of the BWO is 8.31 GHz.

The on-the-fly algorithm detects end of transience at $t_f = 107.20$ ns with $\Delta t_w = 16$ ns, $\Delta_t = 2$ ps and $\delta t_w = 1.6$ ns (see Table 6.2 for details). The training parameters

Table 6.2: On-the-fly DMD parameters

Parameters	**Osc. Beam**	**Vircator**	**BWO**
Δt_w	8 ns	8 ns	16 ns
δt_w	0.4 ns	0.4 ns	1.6 ns
η	10	10	10
q	5	5	5
δ_0	0.05	0.05	0.05
r	200	200	200
Δ_t	8 ps	8 ps	2 ps
d	10	10	5

for offline DMD on current density are provided in Table 6.1. The average extrapolation error in the current density is 13.52% (Fig. 6.11). The higher error can be attributed to particle noise which plays a significant role in the absence of a strong external force dictating the oscillations (oscillating beam). The most energetic mode is the DC mode (Fig. 6.10b) followed by the mode (Fig. 6.10c) oscillating with fundamental frequency at $f_2 = 8.32$ GHz. As expected, the oscillations are concentrated towards the end of BWO where the bunching of electrons (superparticles) occurs. The error in the self electric field from the DMD-EMPIC framework is 5.58% (Fig. 6.11).

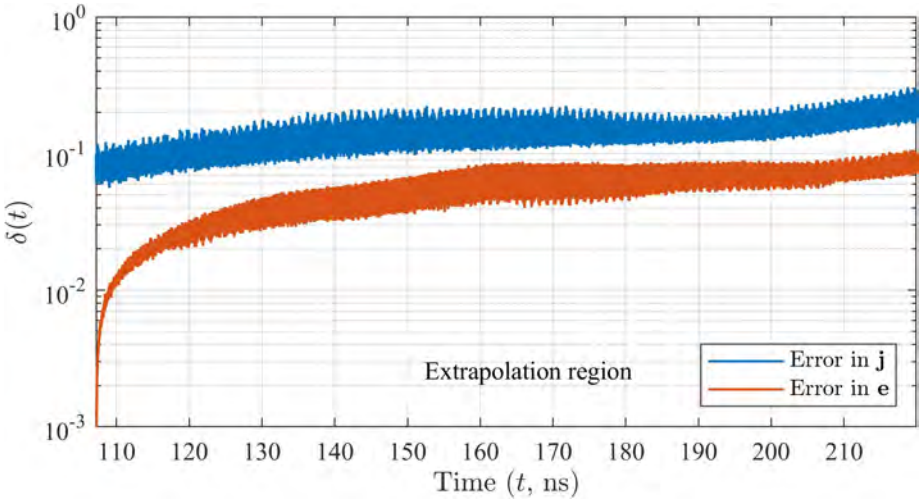


Figure 6.11: Relative 2-norm error in current density **j** (DMD extrapolation) and self electric field **e** (DMD-EMPIC framework) for the backward wave oscillator.

6.3 Computational gain

6.3.1 Computational complexity

As discussed in Section 5.3, the time-complexity of EMPIC simulation to run up to n_q time steps is given by $\mathcal{O}(N^s n_q + N_p n_q)$. For explicit solver, $s = 1$, and for implicit solvers $s > 1.5$. Since we are using explicit solver, we will proceed with $s = 1$.

The computation cost of repeated calculation of the DMD features for the on-the-fly application also adds to the typical EMPIC cost for $n \leq n_f$ ($\equiv t \leq t_f$) in the DMD-EMPIC algorithm. The computation complexity of DMD is dominated by the SVD step, resulting in time-complexity of $\mathcal{O}(l_d^2 N d)$, where l_d is the number of DMD

snapshots after d number of Hankel stacking. Time complexity for randomized DMD with target rank $r(< l_d)$, is typically $\mathcal{O}(rl_dNd)$.Note that as mentioned in [70], SVD need not be recalculated every time we shift the DMD window. The resulting DMD features can be calculated in an incremental manner. In the worst case scenario, let us consider that the SVD is performed for each sliding window. Also, in the worst case scenario, the DMD window is shifted by only one snapshot, i.e. the amount of shift in terms of time steps $\delta n_w = \Delta_n$. With these considerations, the cost of the sliding-window on-the-fly DMD (randomized) is approximately $\mathcal{O}(\frac{rl_dNdn_f}{\Delta_n})$, where n_f is the time step at which EMPIC is stopped, and Δ_n is the number of time steps between two consecutive DMD snapshots.

The time-complexity of the DMD-EMPIC algorithm for $n < n_f$ $(\equiv t < t_f)$ is $\mathcal{O}(Nn_f + N_pn_f + \frac{rl_dNdn_f}{\Delta_n})$, whereas for $n > n_f$ $(\equiv t > t_f)$ is $\mathcal{O}(N(n_q - n_f) + l_d^2Nd)$ considering one-time cost of offline DMD with d Hankel stacks. Since there are no particles involved in case of DMD-EMPIC for $n > n_f$, N_p does not appear in the time complexity. For a simulation run until $n = n_q$, the time-complexity of the DMD-EMPIC is $\mathcal{O}(Nn_q + N_pn_f + \frac{rl_dNdn_f}{\Delta_n} + l_d^2Nd)$. The time-complexities for EMPIC and DMD-EMPIC are summarized in Table 6.3.

6.3.2 Runtime comparison

As mentioned earlier, in typical EMPIC setting $N_p \gg N$. As long as n_q is moderately larger than n_f, there will be significant gain in the runtime. For late-time queries $(n_q \gg n_f)$, the ratio of runtimes for EMPIC (T_q) and DMD-EMPIC

Table 6.3: EMPIC and DMD-EMPIC complexities with explicit field solver.

EMPIC	$\mathcal{O}(Nn_q + N_p n_q)$
DMD-EMPIC	$\mathcal{O}(Nn_q + N_p n_f + \frac{rl_d N d n_f}{\Delta_n} + l_d^2 N d)$

($\widehat{T}_q$) can be roughly given by, $\widehat{T}_q/T_q \approx n_f/n_q$, given field solver takes negligible time compared to the entire simulation. Note that for large scale problems such as BWO, this approximation may not necessarily hold.

The simulation runtime depends on several factors including the specific computational platform and hardware, and code optimization. In this work, the numerical experiments are run on Intel Xeon E5-2680 v4 (Broadwell) compute CPU with 2.4 GHz and 14 cores per processor. Each node has 128 GB of memory. The interconnect used is Mellanox EDR Infiniband Networking (100Gbps). Each simulation job was allocated 1 node and 5 cores. The node runtimes for all the test cases are listed in Table 6.4. Note that the CPU runtime is approximately 5 times the node runtime, exhibiting good shared-memory parallelization across cores.

6.3.3 Effect of Parallelization

It is important to note that particle-in-cell (PIC) algorithms are highly parallelizable and an appropriate parallel computing architecture can be employed to accelerate EMPIC simulations [157]. Single nodes with multiple CPUs (shared memory)

Table 6.4: Node runtime in days.

	EMPIC	**DMD-EMPIC**
Osc. Beam	4.43	1.66
Vircator	5.96	2.93
BWO	7.04	4.08

or multiple nodes (each with one or more CPUs and distributed memory) can reduce the simulation time from days to several hours. Fortunately, the DMD-EMPIC algorithm should also achieve comparable acceleration from parallelization for the following reason: let the computation time for performing DMD (online + offline) be T_{DMD}. Let the runtime for the original EMPIC simulation up to the desired query time be T_q. Suppose the on-the-fly (online) DMD raises the flag to stop the EMPIC simulation at $T_f (< T_q)$. The relative gain in computation time G_T is,

$$G_T = \frac{T_q}{T_f + T_{DMD} + T_{FS}} \approx \frac{T_q}{T_f}, \tag{6.3}$$

where T_{FS} is the time taken by the field solver beyond T_f. This approximation holds if T_{DMD} and T_{FS} are negligible compared to T_f. Depending on the scale of the problem, type of the field solver and parallelization, T_{FS} can be negligible compared to T_f. T_{DMD} is usually much less than T_f even with parallelization, because parallelization not only helps accelerate EMPIC but also the DMD computation. For example, authors in [158] utilize a parallel Tall and Skinny QR (TSQR) algorithm

for parallelizing the SVD computation. Construction of the Koopman operator, eigendecomposition, and the construction of DMD modes are also done in a highly parallel fashion. Recently, authors in [159] used a communication-optimal parallel TSQR algorithm for reduced-communication parallel DMD. These parallel DMD methods are reported to scale well. A detailed study of how parallel EMPIC scales compared to parallel DMD in the presence of parallel computing architecture is beyond the scope of this paper. Since both DMD and EMPIC can benefit from parallel computation, the overall acceleration by DMD-EMPIC depends on how early the system reaches equilibrium or *pseudo* equilibrium, i.e., how small T_f is compared to T_q. Consequently, Eq. (6.3) is independent of the hardware platform or the type of PIC algorithms used, as both T_f and T_q are similarly affected.

6.4 Summary

In this section, we discussed how charged particle dynamics can be indirectly modeled through current density to accelerate EMPIC simulations. We employed Hankel DMD, a Koopman ROM, to model the nonlinear time evolution of current density. We introduced DMD-EMPIC, a physics-informed hybrid ROM framework that integrates with parts of the EMPIC algorithm to enhance its efficiency. The DMD-EMPIC framework partly adheres to the underlying physics in that the self-fields and current density strictly obey Maxwell's equations. However, it does not enforce energy conservation. The DMD-EMPIC approach presents a viable method for accelerating particle-based simulations. Although we utilized Hankel DMD, other

ROMs could also be implemented within the same framework. To address large-scale practical problems, we proposed using randomized Hankel DMD for the on-the-fly components of the framework. We demonstrated the effectiveness of our method through a large-scale example involving a backward wave oscillator.

Chapter 7: Koopman Autoencoders for Modeling Kinetic Plasmas

In order to understand the motivation behind adopting a Koopman reduced-order model such as Koopman autoencoder or KAE for plasma problems, let us revisit the problem we are trying to solve. We are essentially trying to devise a reduced-order model which will help us perform the time-update of dynamic variables such as $\mathbf{E}, \mathbf{B}$ and $\mathbf{J}$ with a reduced memory requirement, lower time complexity, and sufficient accuracy by capturing the dominant physical features of a nonlinear kinetic plasma problem. As noted before, large memory/storage requirement is the direct result of the large number of mesh elements and particles that need to be stored for executing an EMPIC algorithm. A data-driven reduced-order model helps us tackle this issue by extracting the dominant characteristic features from high-fidelity data and modeling the dynamics governed by such reduced number of features or modes, and thus reducing the size of the original problem. Apart from memory usage, another aspect contributing to the high computational load is the long simulation time required by EMPIC simulations. As already discussed in Section 6.3, typical

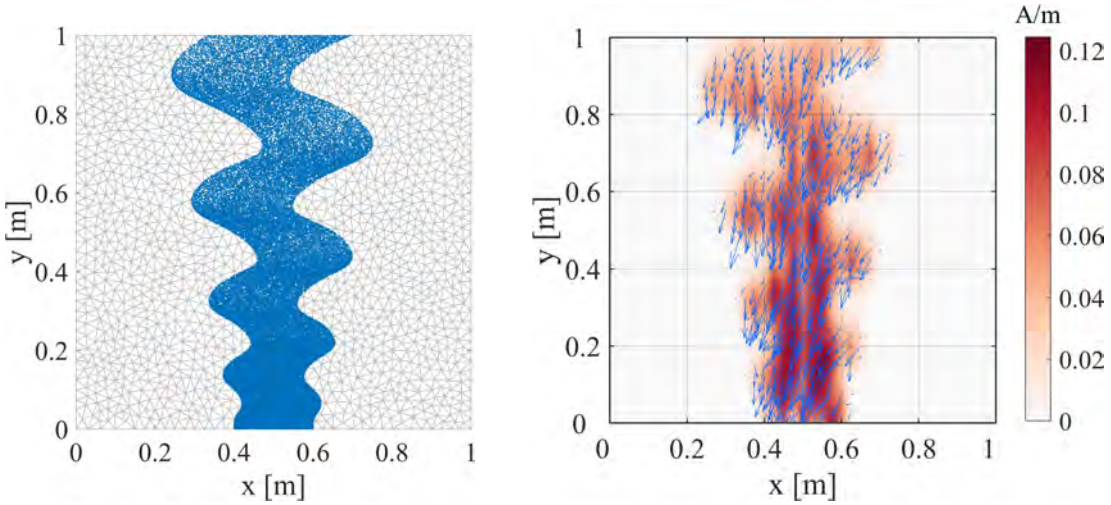
EMPIC algorithms have time complexity of $\mathcal{O}(N^s + N_p)$ at each time step. As a result, modeling time evolution of current density is essential for avoiding steps involving particles.

The primary contribution of this chapter is to study the effectiveness of KAE as a reduced-order time extrapolation model for current density modeling in EMPIC simulations. We previously studied the effectiveness of DMD in Section 6 for modeling the time evolution of $\mathbf{j}$. In this chapter, we will investigate whether nonlinear models such as KAE can provide any benefit over linear models like DMD for modeling plasma systems with oscillatory behavior. Note that here we will explore the usual two test cases: an oscillating electron beam and a virtual cathode oscillator (with a slightly different configuration compared to what was used in Chapter **??**). Thus, this is not a comprehensive comparison between the two methods but rather aims to give readers some idea about the potential advantages of each method. The primary motivation for using a nonlinear modes stems from the inherent nonlinearity in the plasma dynamics. The equations (5.3a) and (5.3b) are just the discretized version of Maxwell's equations which are linear in nature. However, in the EMPIC setting, the time variation of $\mathbf{j}^{(n+\frac{1}{2})}$ in (5.3b) is dependent on the fields as they dictate the particle motion. The gather, pusher, and the scatter stages takes into account this wave-particle interaction and governs the time update of $\mathbf{j}$. This makes the time evolution of $\mathbf{j}$ and the overall dynamics of the problem nonlinear, leading to nonlinear time variation of $\mathbf{e}$ and $\mathbf{b}$. In some cases, such as for physics-informed learning,

we might need to explicitly model $\mathbf{e}$ and $\mathbf{b}$ as well. We will briefly discuss those in Section 7.3. This chapter is based on [49].

7.1 KAE for Modeling Oscillating Electron Beam

We will first revisit and address the problem discussed in [2], which deals with a two-dimensional (2D) oscillating electron beam. Consider a 2D square domain of size 1m $\times$ 1m as shown in Fig. 7.1a. The solution domain is discretized by an unstructured mesh with $N_0 = 1647$ nodes, $N_1 = 4788$ edges, and $N_2 = 3142$ triangular cells for the EMPIC simulation. Perfect electric conductor (PEC) boundary conditions are assumed for the fields. Superparticles representing $r_{sp} = 2 \times 10^5$ electrons, are injected at random points along the x coordinate in the bottom of the domain following a uniform distribution in the range $[0.4 \text{ m}, 0.6 \text{ m}]$. The injection rate is 10 superparticles per time step ($\Delta_t = 0.01$ ns). The superparticles have an initial $+y$ directional velocity of 5×10^6 m/s. They are absorbed as they hit the opposite boundary. An external voltage bias of $V_{ext} = 2$ kV is set to accelerate the electrons in positive y direction. An external, oscillating transverse magnetic flux density field $B_{ext} = B_0 \ \sin(2\pi t/T_b)\ \hat{\mathbf{z}}$ is applied where $B_0 = 10^{-3}$ T and $T_b = 20$ ns. The EMPIC simulation is run for $n = 240,000$ time steps and the current density DoFs ($\mathbf{j}^{(n+1/2)}$) are sampled from $n = 16,000$ to $n = 240,000$ with sampling interval $\Delta_n = 40$ (in terms of time steps), providing total $5,601$ time samples (or snapshots) for training and testing.



(a) Electron beam snapshot at $n = 40000$.

(b) Current density snapshot at $n = 40000$.

Figure 7.1: (a) Snapshot of an oscillating 2D electron beam propagating along the $+y$ direction at $n = 40000$. The blue grey dots represent the superparticles in the backdrop of the unstructured mesh shown by the light grey. This figure is adapted from [2]. (b) The quiver plot of current density at $n = 40000$. The blue arrows indicate the direction and their magnitude is shown in the background.

As discussed earlier, the various stages of the EMPIC algorithm involving particles act as computational bottlenecks, restricting our ability to speed up the simulations. The steps involving particles are necessary for the time update of the current density $\mathbf{j}^{(n+1/2)}$, which is an integral part of the full Maxwell's time update equations (5.3). If we can otherwise model and predict the time evolution of $\mathbf{j}$, we can bypass the time consuming steps involving particles in the EMPIC algorithm. Note, however, that the inherent nonlinearity of the problem makes it challenging to accurately model the evolution of $\mathbf{j}$. In addition, $\mathbf{j}$ is not guaranteed to have a smooth

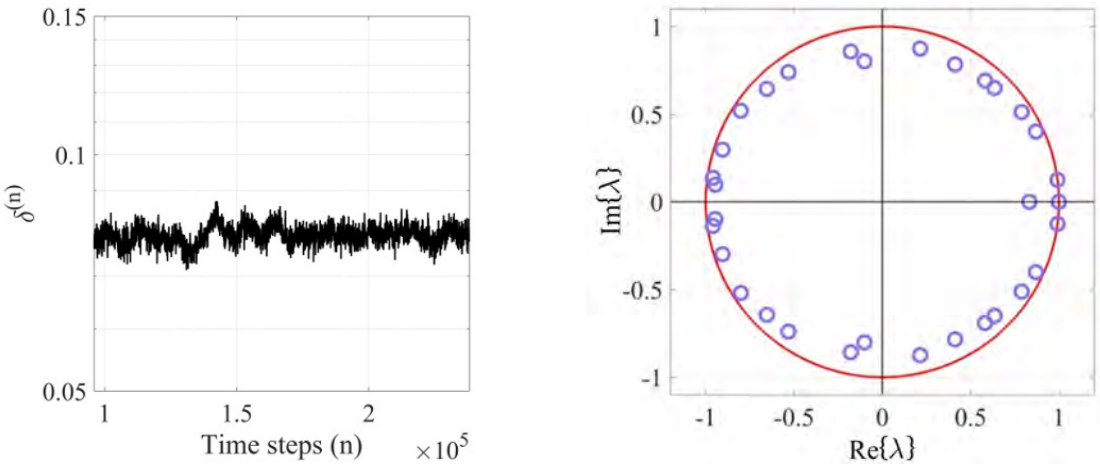


Figure 7.2: (a) Relative 2-norm error for $\mathbf{j}_{nz}^{(n+1/2)}$ in the extrapolation region. (b) The Koopman eigenvalues (small circles) corresponding to C, the forward dynamics. The large circle represents the unit circle in the complex plane. This figure is adapted from [2].

behavior in the spatial or in the temporal domain. As noted before, when dealing with nonlinear problems, it makes sense to exploit nonlinear transformations which are realizable by nonlinear activation layers in KAE [2].

As can be seen in Fig. 7.1a, the charged particles are confined to a limited region of space along the horizontal direction centered around the midpoint $x = 0.5$ m. As a result, the corresponding current density (Fig. 7.1b) is zero outside this region of space. So, a new vector $\mathbf{j}_{nz}^{(n+1/2)}$ is formed with the edges (i) where current density DoF is nonzero, i.e. $j_i^{(n+1/2)} \geq 10^{-20}$ A/m for all the training time samples. Note that in the context of Fig. 2.3, $\mathbf{x}^{(n)} = \mathbf{j}_{nz}^{(n+1/2)}$, since we want to model the

time-evolution of snapshots of current density (Fig. 7.1b) using KAE. This results in $N_{in} = N_{out} = N = 2,072$. The other hyperparameters are chosen as follows: $N_h = 256$ and $N_b = 32$ with $\gamma_{id} = 1$, $\gamma_{fwd} = 2$, $\gamma_{bwd} = 0.1$ and $\gamma_{con} = 0.01$. A total of $2,000$ training samples are used spanning from $n = 16,000$ to $95,960$ with consecutive time samples sampled at 40 time steps ($\Delta_n = 40$) apart. The model is trained for 100 steps ($k_m = 100$) in the forward and the backward directions, and tested on $3,500$ time samples in the extrapolation region. The key training parameters are summarized in Table 7.1.

Parameters	Δn	k_m	N_{in}, N_{out}	N_h	N_b	γ_{fwd}	γ_{id}	γ_{bwd}	γ_{con}
Value	40	100	2072	256	32	2	1	0.1	0.01

The parameters N_{in} and N_{out} depend on the number of mesh elements over which we seek to model the current density. The N_h and N_b values are commonly tuned by trial and error. However, the physics of the problem provides some insight on how to choose N_b. For example, N_b essentially denotes the number of features for modeling the dynamics in the transformed space. The Koopman eigenvalues (λ) are related to the frequencies ($\omega = \ln|\lambda|/\Delta_t$, Δ_t being the time interval between training samples) under which the Koopman modes evolve in time. Since we are dealing with real-valued data, the Koopman eigenvalues exist in complex-conjugate pairs, except for those residing on the real axis. If we consider the original state space, there are two primary frequency components for this electron beam problem: the static component (representing the net, longitudinal particle transport from the bottom to the top of the domain) and the transversal oscillatory component with oscillation frequency

matching that of the external magnetic flux. So, we can expect at the very minimum two modes (three eigenvalues) to roughly represent the dynamics in the original state space. However, in case of KAE, we are dealing with a nonlinearly transformed space and we generally do not have intution regarding how these frequency components may be transformed (through a nonlinear transformation) to the feature space. However, $N_b > 3$ acts as a good starting point. High values of N_h might result in a large number of trainable parameters which could lead to overfitting. On the other hand, while small N_h avoids overfitting, the model may not be able to capture all the intricacies of the dynamics. The weights for the loss functions also play a crucial role in determining the extrapolation accuracy of the KAE model. Intuitively, it makes more sense to assign larger weights to γ_{fwd} and γ_{id} as our primary interest is to predict (extrapolate) the solution in future time. After gaining a rough idea about the range of various weights, grid search can be implemented to determine their optimum values.

Note that in the context of this problem, (2.28a) can be re-written as

$$\mathbf{j}_{nz}^{(n+\frac{1}{2}+k\Delta n)} \approx \hat{\mathbf{j}}_{nz}^{(n+\frac{1}{2}+k\Delta n)} = \mathbf{\Psi}_d \circ \mathbf{K}_f^k \circ \mathbf{\Psi}_e(\mathbf{j}_{nz}^{(n+\frac{1}{2})}), \tag{7.1a}$$

$$\mathbf{j}_{nz}^{(n+\frac{1}{2}-k\Delta n)} \approx \hat{\mathbf{j}}_{nz}^{(n+\frac{1}{2}-k\Delta n)} = \mathbf{\Psi}_d \circ \mathbf{K}_b^k \circ \mathbf{\Psi}_e(\mathbf{j}_{nz}^{(n+\frac{1}{2})}). \tag{7.1b}$$

Fig. 7.2a shows that the relative error in the predicted $\mathbf{j}_{nz}$ is around 8%. Note that the relative error is calculated over all the mesh element edges associated with $\mathbf{j}_{nz}$. It is important to mention that we can always go back to $\mathbf{j}$ (or $\hat{\mathbf{j}}$) from $\mathbf{j}_{nz}$ (or $\hat{\mathbf{j}}_{nz}$), simply by zero padding. The expression for the relative error at n^{th} time step

is given by

$$\delta^{(n)} = \frac{||\hat{\mathbf{j}}_{nz}^{(n+\frac{1}{2})} - \mathbf{j}_{nz}{}^{(n+\frac{1}{2})}||_2}{||\mathbf{j}_{nz}{}^{(n+\frac{1}{2})}||_2}, \tag{7.2}$$

where hatted quantities represent the KAE reconstruction and $||\cdot||_2$ denotes the L^2 norm. The stability of the solution, even for long term predictions can be attributed to the Koopman eigenvalues, as depicted in Fig. 7.2b, staying inside or on the unit circle in the complex plane. This can be better understood from the perspective of theory of linear systems. Let us go back to the (2.9), (2.15) which tells us that under suitable transformation $\mathbf{g}(\cdot)$, any dynamical system can be represented through linear dynamics:

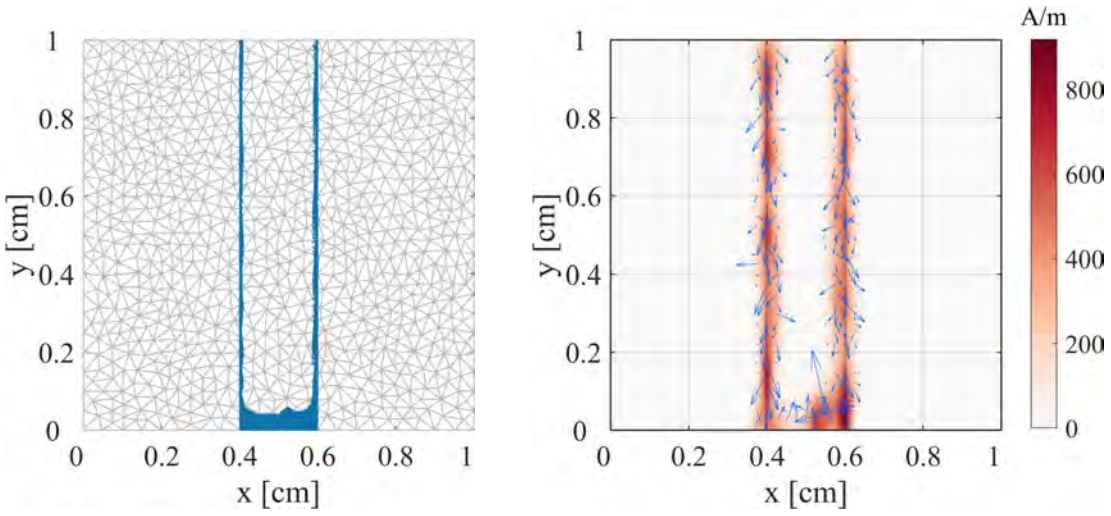
$$\mathbf{g}(\mathbf{x}^{(n+1)}) = \mathbf{K} \cdot \mathbf{g}(\mathbf{x}^{(n)}), \tag{7.3}$$

In our setting, $\mathbf{x} \equiv \mathbf{j}_{nz}$, $\mathbf{g} \equiv \mathbf{\Psi}_e$ and $\mathbf{K} \equiv \mathbf{K}_f$. From theory of linear systems, we can say that for a stable systems, eignevalues of $\mathbf{K}$, $|\lambda_i| \leq 1$, or in other words, eigenvalues of $\mathbf{K}_f$ should lie on or inside the unit circle, which is the case in Fig. 7.2b. However, we must emphasize that for neural networks with continuous activation functions, stability in the latent space will guarantee stability in the original state space if the transformation is Lipschitz continuous. In this work, we use the hyperbolic tangent (tanh) activation function, ensuring that the transformation is continuous or "smooth." However, any guarantees regarding Lipschitz continuity cannot be provided. Nevertheless, the location of eigenvalues in the latent space can offer insights into potential instability.

7.2 KAE for Modeling Virtual Cathode Oscillations

Next, we test KAE for virtual cathode formation. Fig. 7.3a shows a 2D electron beam propagating along the positive y direction and forming the virtual cathode. The solution domain (xy plane) has dimensions 1 cm $\times$ 1 cm and is discretized by an unstructured mesh consisting of $N_0 = 844$ nodes, $N_1 = 2447$ edges, and $N_2 = 1604$ triangular cells. Superparticles are injected with initial velocity $\mathbf{v}_0 = 5 \times 10^6\ \hat{\mathbf{y}}$ m/s at the bottom of the cavity with uniform random distribution in the spatial interval $[0.4, 0.6]$ cm along x. The injection number rate of superparticles is set to 30 per time step (with $r_{sp} = 5 \times 10^5$ and $\Delta t = 0.2$ ps). A strong confining magnetic field, $B_{ext} = B_y\ \hat{\mathbf{y}}$ is applied in the y direction, with $B_y = 100$ T.

The current density is sampled at every 80^{th} time steps ($\Delta_t = 80$), generating a total of $2,000$ time samples. A virtual cathode formation with stable oscillations takes place after the transient regime is over. We are essentially trying to model these oscillations in the current density (Fig. 7.3b) using the KAE. We consider a total of 800 time samples for training, starting from the 400^{th} time sample onward. We use 350 time samples for testing. An approach similar to the one considered in the previous example is taken in order to model the current density; however, unlike the previous case where the oscillations are prominent due to external magnetic flux, the oscillations in this case are more subtle and localized. The oscillations are localized in the sense that they take place at the root of the beam, close to the virtual cathode, and a lingering effect can be observed at the edges due to the lateral beam leakage.



(a) Beam snapshot at $n = 80000$. (b) Current density snapshot at $n = 80000$.

Figure 7.3: (a) Snapshot of virtual cathode formation at $n = 80,000$. The blue dots represent the superparticles, whereas the light grey lines indicate the unstructured mesh. Two leakage streams are observed, corresponding to the electrons that are able to laterally circumvent the virtual cathode repulsion. (b) The quiver plot of current density with magnitude at $n = 80,000$.

The higher concentration of superparticles results in "smoother" time variation of the current density at the root of the beam compared to boundary of the beam (Fig. 7.5). Such weak localized oscillations make the modeling task challenging which is reflected in relatively higher error in reconstruction in the extrapolation region.

The KAE training parameters in this case are chosen as follows 7.2:

Parameters	Δn	k_m	N_{in}, N_{out}	N_h	N_b	γ_{fwd}	γ_{id}	γ_{bwd}	γ_{con}
Value	80	100	459	256	32	2	1	0.1	0.01

Fig. 7.4 shows on average relative error of 15% in the predicted current density. Similar to the oscillating beam case, the Koopman eigenvalues residing inside or on

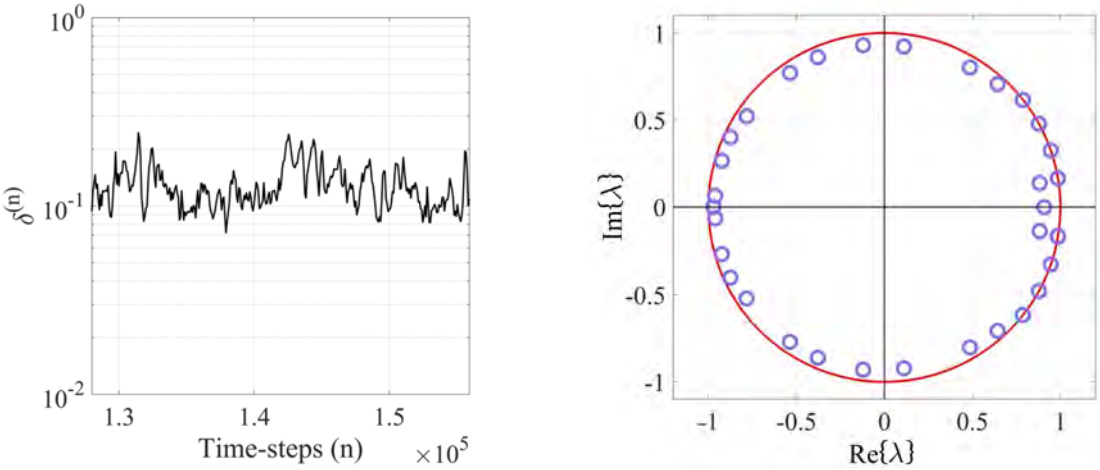


Figure 7.4: (a) Relative 2-norm error in the extrapolation region for $\mathbf{j}_{nz}^{(n+1/2)}$ (b) The relative position of the Koopman eigenvalues (corresponding to forward dynamics) w.r.t. the unit circle in the complex plane.

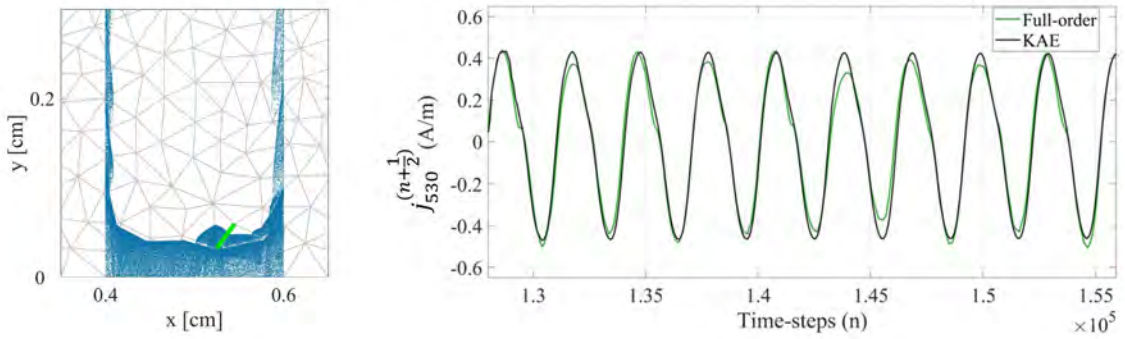
the unit circle ensures stability in the reconstruction. However, relatively high error in reconstruction can be attributed to limitations of fully connected neural networks in learning high-frequency functions. Note that KAE is realized by fully connected layers and fully connected NNs are known to have spectral bias [160] also known as "F-principle" [161]. As mentioned earlier, the time signature of current density is not so "smooth" for this particular test case, especially at the mesh edges located at two sides of the beam. Fig. 7.5 shows how smoothness of current density time signature varies based on the density of superparticles. The edge indexed 530 (indexed with respect to 2447 edges) is located at the root of the beam where superparticles are concentrated.

Consequently the time variation of j is smooth and KAE reconstruction follows the original high-order solution closely as can be seen in Fig. 7.5a. Same comments can not be made for the edge indexed 777, which encounters much lower density of superparticles resulting in non-smooth time variation in j (Fig. 7.5b). This phenomena leads to higher relative error in the KAE reconstruction.

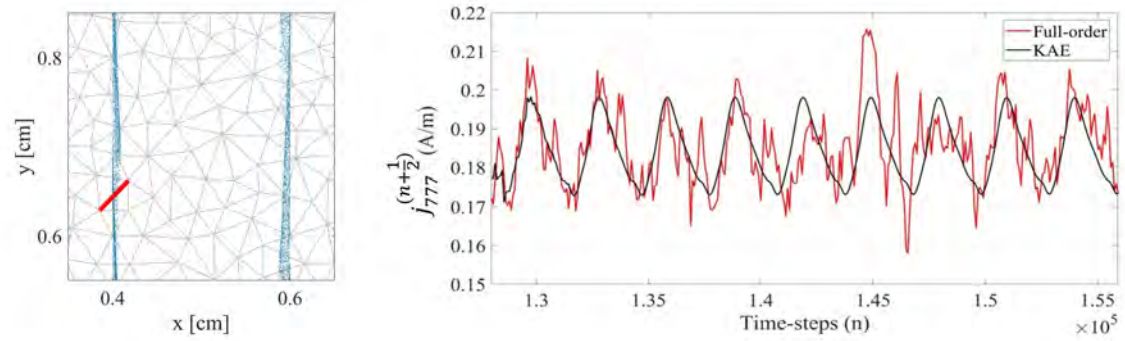
One of the primary reasons behind the non-smooth nature of the current variation is the point nature of the superparticles, i.e. representation of number density function for a superparticle through a delta distribution. Consideration of finite-size superparticles in EMPIC algorithms can mitigate this issue to some extent, but it poses additional implementation challenges. One straightforward approach would be to inject a much large number of superparticles while keeping the $N_p \times r_{sp}$ same. But recall that time step complexity of EMPIC algorithms is approximately $\mathcal{O}(N_p)$. therefore, a careful consideration should be made regarding the trade-off between smoothness and computational cost.

7.2.1 Computational Gain

The potential computational gain is similar to that in Section 6.3. Once the model is trained, the current density at any time step can be queried by $\mathcal{O}(N_1)$ or equivalently $\mathcal{O}(N)$ operations, effectively reducing the one time step cost from $\mathcal{O}(N+N_p)$ tp $\mathcal{O}(N)$. Recall that N_1 is the number of mesh element edges over which $\mathbf{j}$ evolves and N is the aggregate mesh dimension. Fig. 7.6 illustrates this approach, which can be exploited similar to DMD-EMPIC framework (Fig. 6.1) as discissed in



(a) KAE reconstruction at root of the beam at mesh edge index 530 (green edge).



(b) KAE reconstruction at edge of the beam at mesh edge index 777 (red edge).

Figure 7.5: (a) Current density sampled at the root of the beam with larger concentration of superparticles (blue dots). The top right figure shows the corresponding reconstruction of current density DoF. (b) KAE reconstruction of the current density (bottom right) at the edge of the beam experiencing leakage current (bottom left).

Chapter 6. However, the primary drawback of KAE compared to DMD is its training time. Training any neural network model requires tuning hyperparameters, which can be time-consuming. For high-fidelity simulations with very long runtimes, KAE can be beneficial as a reduced-order time extrapolation model. The primary appeal of neural network-based models lies in their applicability to parametric problems

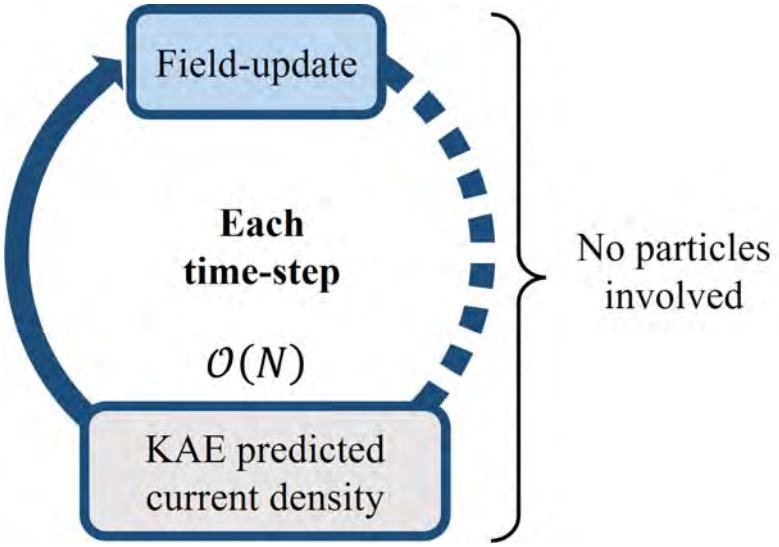
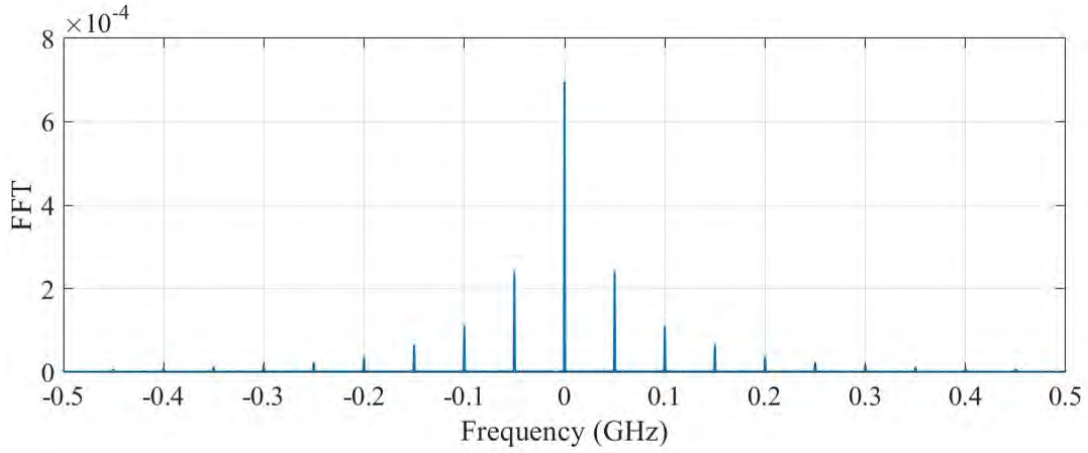


Figure 7.6: EMPIC with KAE predicted current density ($\hat{\mathbf{j}}$) only involvs field update with $\hat{\mathbf{j}}$, with time step complexity of $\mathcal{O}(N)$.

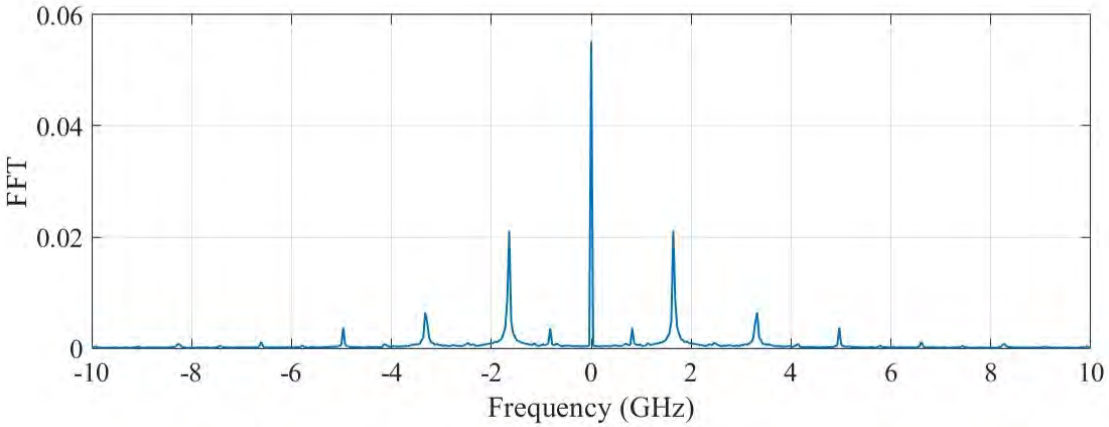
and their straightforward approach to enforcing physical constraints. This is further discussed in Section 7.3.

7.2.2 DMD vs. KAE

In order to compare KAE's performance with DMD, we applied Hankel DMD on the test cases mentioned in this chapter. The average extrapolation error is listed in Table 7.1. We can clearly see that KAE does not provide any obvious advantage over DMD for modeling the plasma beam oscillations. Although this might initially seem strange, a deeper look into the spectrum of each problem clarifies the situation. Note that a nonlinear model such as KAE will be advantageous over linear models



(a) Current density FFT for oscillating electron beam.



(b) Current density FFT for virtual cathode oscillator (vircator).

Figure 7.7: Frequency spectrum of oscillating electron beam and vircator.

like DMD for highly nonlinear problems. While it is not straightforward to quantify the extent of nonlinearity, one measure can be the spread of the spectrum.

Whether it is the oscillating electron beam or the vircator, both have a fundamental oscillation frequency. The nonlinearity in the system gives rise to harmonics.

Table 7.1: KAE vs. DMD comparison of extrapolation error.

	KAE	**DMD**
Osc. Beam	8.00 %	7.83 %
Vircator	12.79 %	10.95 %

However, if the spectrum is more or less discrete, DMD will be able to accurately model the dynamics. In such cases, linear models like DMD, especially Hankel DMD, will always have an advantage over more complicated nonlinear models. Theoretically, a dynamic system governed by a band-limited discrete spectrum can always be modeled by a finite number of harmonics, thus DMD modes and frequencies. Sharp peaks in Fig. 7.7a explain the good performance of DMD. The spectrum in Fig. 7.7b is slightly more spread, leading to less accurate DMD reconstruction. However, overall sharp distinct peaks in both cases ensure good performance for DMD. KAE performance is also close and might slightly improve with further tuning.

The primary takeaway is that the test cases are not "nonlinear" enough to establish a clear advantage of KAE over DMD. Further thorough investigation for a variety of test cases needs to be carried out to reach any solid conclusion. Note that the higher DMD error, compared to the test cases in Chapter 6 is due to the lower particle injection rate, which results in higher particle noise in the data.

7.3 Towards A Physics-Informed Approach

One key advantage of neural network-based models is the ease of incorporating physical constraints into the reduced-order model. Modeling and forecasting multiscale multiphysics systems from a purely data-driven perspective has its own limitations. This is because machine learning models, and in particular deep learning models involving neural networks, tend to lack general interpretability. Models trained purely on data might fit the observations well in the training region, but when it comes to predicting beyond the observed dataset, these models typically fail in the presence of any new underlying physical regimes not present in the training dataset (for example, beyond a phase transition) [162]. Also, the accuracy of any data-driven model depends on the quality of the data used to train that particular model. Often, the available data are corrupted with noise. Therefore, it becomes important to have some a priori information regarding the governing physics model of the system to distinguish "physical data" from "non-physical data" (noise). Another motivation for including physics information into the training process stems from the data-hungry nature of ML models. In many cases, certain parts of a physical system might be inaccessible to experiments or a large amount of time might be necessary to collect/generate sufficient amount of data. In such data-limited scenarios, one needs to incorporate physics constraints to guide the ML model through any limited-data regime. In the past, one of the strategies for the inclusion of physics knowledge in the training of machine learning models included the so-called knowledge-based neural

networks [163, 164, 165, 166]. It is of note that neural networks have been used in the past to directly solve PDEs [167] and to emulate certain types of measurement data as well [168]. However, a number of recent advances in ML techniques have shed new light [169, 170] on the topic of physics-informed machine learning (PIML). The reader is referred to [162] for a recent account of ongoing developments in this area.

Depending on the available data, there can be broadly three categories of prior information levels encountered while trying to model a physical system. First, a limited amount of data and a comprehensive knowledge of the governing physics leads to traditional approaches where physics equations are used to describe a system. On the other extreme, an absolute lack of information regarding the underlying physics and the presence of a large dataset motivates a purely data-driven blackbox approach. The more common scenario lies somewhere in the middle, where some physics information regarding the system is available (for example some of the governing equations but not the full boundary conditions, energy conservation, expected underlying symmetries, etc.), and some data is available through measurements or partial simulations. A "physics-informed" or "physics-guided" machine learning approach is suitable for such scenarios.

We have so far demonstrated the effectiveness of KAE for modeling EMPIC simulations without involving the physics of the problem (other than using high-fidelity simulations to provide the underlying data for training). Therefore, this approach

is more data-centric rather than physics-centric. As a result, there is no guarantee that reconstructed current density (or the electromagnetic field for that matter) will follow the physics relations discussed in (5.3). One way to make the model aware of the underlying physics is to introduce new loss term(s) incorporating physical constraints (laws and/or boundary conditions) so that the physics information can be included during the training process. In the context of EMPIC kinetic plasma simulations, the energy constrain or the contraints arising from the Maxwell's equations can be taken into account while training KAE. The new physics loss term will be simply equivalent to the equation error in the discretized Maxwell's equations (5.3), or energy equation. These involve the quantities $\mathbf{e}$, $\mathbf{b}$, $\mathbf{j}$, $\mathbf{C}$, $\tilde{\mathbf{C}}$ and the Hodge star matrices $[\star_{\mu^{-1}}]$ and $[\star_{\epsilon}]$. Recall that $\mathbf{C}$ and $\tilde{\mathbf{C}}$ depends only on the connectivity among mesh elements whereas $[\star_{\mu^{-1}}]$ and $[\star_{\epsilon}]$ contains the metric information from the computational mesh. In other words, for a fixed mesh, $\mathbf{C}$, $\tilde{\mathbf{C}}$, $[\star_{\mu^{-1}}]$ and $[\star_{\epsilon}]$ are fixed. For each of $\mathbf{e}$, $\mathbf{b}$ and $\mathbf{j}$ we can have separate KAE architectures, the outputs of which will be connected through the equation error from physical constraints. Note that there are several training challenges associated with physics-informed learning in general [171]. It can be challenging for neural networks to learn physical laws from loss functions derived from complex equations, such as Maxwell's discrete field update equations ((5.3)), where multiple quantities are being predicted. Additionally, introducing a new loss term only imposes the physics constraints in a "soft" manner, which does not necessarily ensure exact adherence to the underlying physical

principles. This is an area of ongoing research and further investigations need to be carried out to take full advantage of physics-informed approach.

7.4 Summary

While KAE provides a unique approach to realizing a finite-dimensional Koopman-invariant subspace, its practical use for pure time extrapolation in scientific machine learning might be limited. The main limitations stem from the training complexity of neural networks, which can hinder its applicability for real-time acceleration of high-fidelity time-domain simulations. Particularly for systems with a band-limited discrete spectrum, linear models that employ time-delay embeddings, such as Hankel DMD, are often more suitable than neural network-based models.

However, the true potential of neural network-based models like KAE can be leveraged for parametric problems and physics-informed learning. KAEs can be trained across different parameters—such as varying initial conditions, boundary conditions, and beam parameters—eliminating the need for new simulations each time. Furthermore, as discussed in Section 7.3, KAEs are well-suited for incorporating physics-informed learning approaches. This makes KAE an ideal Koopman ROM for physics-informed learning across various simulations, rather than limiting it to a case-by-case application.

Chapter 8: Conclusions and Future Work

Extrapolating periodic data, such as limit cycle behavior, might seem trivial, but traditional statistical tools like auto regressive integrated moving average (ARIMA) and Fourier-based models are not well-suited for long-term predictions of high-dimensional complex nonlinear systems, especially when the assumption of stationarity is violated. Neural network (NN) based models, such as recurrent neural networks (RNN) and long short-term memory (LSTM) networks, can effectively handle nonlinearity and perform well for short-term predictions. However, for long-term forecasting of high-dimensional complex nonlinear dynamics, Koopman operator-based methods have been shown to outperform the aforementioned methods. Additionally, Koopman reduced-order models (ROMs) are designed to learn reduced-order dynamics in a low-dimensional space, leading to better scalability. This scalability is crucial for modeling cavity resonances and EMPIC simulations, which consist of thousands to millions of mesh elements (edges, facets, etc.). The high dimensionality, inherent nonlinearities, and complex nature of plasma dynamics make Koopman ROMs an ideal choice for reduced-order modeling.

In this dissertation, we explored data-driven reduced-order models based on Koopman operator theory for spatio-temporal modeling of electromagnetic fields and currents in electromagnetic cavities exhibiting linear dynamics, as well as kinetic plasma dynamics showing nonlinear behavior. Although Koopman-based models for linear problems reduce to linear state-space models, the Hankel DMD approach provides significant insights into modeling linear systems from partial observations. For nonlinear plasma problems, the time-delayed stacking of the original state in Hankel DMD offers a "better" Koopman observable for modeling nonlinearities. The Koopman Autoencoders (KAEs) exploit the nonlinearity of neural networks and the dimension reduction capabilities of autoencoders to search for a reduced-order Koopman observable space, thereby modeling high-dimensional nonlinear plasma problems more accurately than DMD. However, this comes at the cost of neural network training, which can be time-consuming and complicated depending on the problem's complexity. Koopman ROMs, especially DMD, provide meaningful insights into the dynamics of the problem, identifying stable and unstable modes, which is crucial for control applications. Nevertheless, the primary focus of this dissertation has been on the predictive capabilities (time extrapolation) and computational efficiency of Koopman ROMs for reducing the simulation time of high-fidelity time-domain simulations based on FDTD, FETD, or EMPIC. We proposed two on-the-fly DMD algorithms for real-time termination of high-fidelity simulations, effectively reducing

the runtime of EMPIC and FDTD cavity simulations. The DMD predictions closely match the high-fidelity simulations and help expedite frequency domain analysis.

There are several exciting avenues to further build on or extend this work. The DMD mode tracking algorithm proposed in this dissertation provides insights into how DMD features evolve from transient to equilibrium, paving the way for future work on predicting equilibrium behavior by observing only transient data. Further extensions of DMD, such as extended DMD, kernel DMD, and physics-informed DMD, can be exploited to better model plasma nonlinearities. An active area of research focuses on increasing the robustness and trustworthiness of data-driven models. Establishing an error bound for DMD predictions can make these models applicable to more sensitive real-world applications. Physics-informed machine learning (PIML) is an active area of research that focuses on incorporating underlying physics to make ML models more accurate, robust, and trustworthy. KAEs can be further improved by incorporating physical constraints such as charge conservation in EMPIC simulations or energy conservation in EM wave propagation problems. Currently, most neural network models are based on multilayer perceptrons (MLPs), which do not account for spatial awareness. Emerging fields like geometric machine learning (GeoML), including graph neural networks (GNNs), hold great promise for modeling unstructured particle phase-space as well as fields and currents on unstructured meshes.

Bibliography

[1] V. Martínez-Cagigal, "Custom colormap," 2023. MATLAB Central File Exchange. Retrieved February 22, 2023.

[2] K. M. I. Nayak and F. L. Teixeira, "Koopman autoencoder architecture for current density modeling in kinetic plasma simulations," in *2021 International Applied Computational Electromagnetics Society Symposium (ACES) (accepted)*, 2021.

[3] A. Bondeson, T. Rylander, and P. Ingelström, *Computational electromagnetics*. Springer, 2012.

[4] A. Taflove, S. C. Hagness, and M. Piket-May, "Computational electromagnetics: the finite-difference time-domain method," *The Electrical Engineering Handbook*, vol. 3, pp. 629–670, 2005.

[5] T. Rylander, P. Ingelström, and A. Bondeson, *Computational electromagnetics*. Springer Science & Business Media, 2012.

[6] S. H. Gold and G. S. Nusinovich, "Review of high-power microwave source research," *Rev. Sci. Instrum*, vol. 68, no. 11, pp. 3945–3974, 1997.

[7] J. H. Booske, "Plasma physics and related challenges of millimeter-wave-to-terahertz and high power microwave generation," *Phys. Plasmas*, vol. 15, no. 5, p. 055502, 2008.

[8] T. Tajima, *Computational plasma physics: with applications to fusion and astrophysics*. CRC press, 2018.

[9] S. Kutsaev, "Advanced technologies for applied particle accelerators and examples of their use," *Technical Physics*, vol. 66, pp. 161–195, 2021.

[10] E. Priest, *Magnetohydrodynamics of the Sun*. Cambridge University Press, 2014.

[11] M. L. Goldstein, D. A. Roberts, and W. H. Matthaeus, "Magnetohydrodynamic turbulence in the solar wind," *Annual Review of Astronomy and Astrophysics*, vol. 33, no. 1, pp. 283–325, 1995.

[12] C. Sovinec, A. Glasser, T. Gianakon, D. Barnes, R. Nebel, S. Kruger, D. Schnack, S. Plimpton, A. Tarditi, and M. Chu, "Nonlinear magnetohydrodynamics simulation using high-order finite elements," *Journal of Computational Physics*, vol. 195, no. 1, pp. 355–386, 2004.

[13] J. Benford, J. A. Swegle, and E. Schamiloglu, *High Power Microwaves*. New York: CRC Press, 2015.

[14] G. Chen, L. Chacón, and D. C. Barnes, "An energy-and charge-conserving, implicit, electrostatic particle-in-cell algorithm," *Journal of Computational Physics*, vol. 230, no. 18, pp. 7018–7036, 2011.

[15] M. Kraus, K. Kormann, P. J. Morrison, and E. Sonnendrücker, "GEMPIC: geometric electromagnetic particle-in-cell methods," *Journal of Plasma Physics*, vol. 83, no. 4, p. 905830401, 2017.

[16] C. K. Birdsall and A. B. Langdon, *Plasma physics via computer simulation*. CRC press, 2004.

[17] K. Eppley, "The use of electromagnetic particle-in-cell codes in accelerator applications," Tech. Rep. SLAC-PUB-4812; CONF-881049-46 ON: DE89004968, Stanford Linear Accelerator Center, Menlo Park, CA (USA), 1988.

[18] R. Lehe, M. Kirchen, B. B. Godfrey, A. R. Maier, and J.-L. Vay, "Elimination of numerical Cherenkov instability in flowing-plasma particle-in-cell simulations by using Galilean coordinates," *Phys. Rev. E*, vol. 94, p. 053305, Nov 2016.

[19] O. Shapoval, R. Lehe, M. Thévenet, E. Zoni, Y. Zhao, and J.-L. Vay, "Overcoming timestep limitations in boosted-frame particle-in-cell simulations of plasma-based acceleration," *Phys. Rev. E*, vol. 104, p. 055311, Nov 2021.

[20] G. Lapenta, J. Brackbill, and P. Ricci, "Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas," *Physics of plasmas*, vol. 13, no. 5, p. 055904, 2006.

[21] I. Nayak, D. Na, J. L. Nicolini, Y. A. Omelchenko, and F. L. Teixeira, "Progress in kinetic plasma modeling for high-power microwave devices: Analysis of multipactor mitigation in coaxial cables," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 2, pp. 501–508, 2020.

[22] H. Karimabadi, B. Loring, H. X. Vu, Y. Omelchenko, V. Roytershteyn, W. Daughton, T. Sipes, M. Tatineni, A. Majumdar, U. Catalyurek, and A. Yilmaz, "Petascale kinetic simulation of the magnetosphere," in *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*, TG '11, (New York, NY, USA), pp. 1–2, Association for Computing Machinery, 2011.

[23] Y. Chen, G. Tóth, H. Hietala, S. K. Vines, Y. Zou, Y. Nishimura, M. V. D. Silveira, Z. Guo, Y. Lin, and S. Markidis, "Magnetohydrodynamic with embedded particle-in-cell simulation of the geospace environment modeling dayside kinetic processes challenge event," *Earth and Space Science*, vol. 7, no. 11, p. e2020EA001331, 2020.

[24] R. Zheng, P. Ohlckers, and X. Chen, "Particle-in-cell simulation and optimization for a 220-ghz folded-waveguide traveling-wave tube," *IEEE Transactions on Electron Devices*, vol. 58, no. 7, pp. 2164–2171, 2011.

[25] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[26] F. J. Montáns, F. Chinesta, R. Gómez-Bombarelli, and J. N. Kutz, "Data-driven modeling and learning in science and engineering," *Comptes Rendus Mécanique*, vol. 347, no. 11, pp. 845–855, 2019.

[27] S. J. Qin, "Process data analytics in the era of big data," *AIChE Journal*, vol. 60, no. 9, pp. 3092–3100, 2014.

[28] A. Labrinidis and H. V. Jagadish, "Challenges and opportunities with big data," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, 2012.

[29] Y. Hajjaji, W. Boulila, I. R. Farah, I. Romdhani, and A. Hussain, "Big data and iot-based applications in smart environments: A systematic review," *Computer Science Review*, vol. 39, p. 100318, 2021.

[30] W. Pan, Z. Li, Y. Zhang, and C. Weng, "The new hardware development trend and the challenges in data management and analysis," *Data Science and Engineering*, vol. 3, no. 3, pp. 263–276, 2018.

[31] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, and Z. Zhang, "Hardware for machine learning: Challenges and opportunities," in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–8, IEEE, 2017.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[33] R. Iten, T. Metger, H. Wilming, L. Del Rio, and R. Renner, "Discovering physical concepts with neural networks," *Physical review letters*, vol. 124, no. 1, p. 010508, 2020.

[34] G.-H. Liu and E. A. Theodorou, "Deep learning theory review: An optimal control and dynamical systems perspective," 2019.

[35] S. Scher, "Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning," *Geophysical Research Letters*, vol. 45, no. 22, pp. 12–616, 2018.

[36] S. D. Campbell and F. X. Diebold, "Weather forecasting for weather derivatives," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 6–16, 2005.

[37] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the national academy of sciences of the united states of america*, vol. 17, no. 5, p. 315, 1931.

[38] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, Aug 2010.

[39] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," *Journal of Computational Dynamics*, vol. 1, no. 2, p. 391–421, 2014.

[40] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: Data-driven modeling of complex systems*. SIAM, Dec 2016.

[41] I. Nayak and F. L. Teixeira, "Data-driven modeling of high-Q cavity fields using dynamic mode decomposition," in *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, pp. 1118–1119, 2022.

[42] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning koopman invariant subspaces for dynamic mode decomposition," *Advances in neural information processing systems*, vol. 30, 2017.

[43] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, no. 1, pp. 1–10, 2018.

[44] O. Azencot, N. B. Erichson, V. Lin, and M. Mahoney, "Forecasting sequential data using consistent Koopman autoencoders," in *International Conference on Machine Learning*, pp. 475–485, PMLR, 2020.

[45] Y. Zhang and L. Jiang, "A novel data-driven analysis method for electromagnetic radiations based on dynamic mode decomposition," *IEEE Trans. Electromagn. Compat.*, vol. 62, no. 4, pp. 1443–1450, 2020.

[46] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, and S. L. Brunton, "Characterizing magnetized plasmas with dynamic mode decomposition," *Physics of Plasmas*, vol. 27, no. 3, p. 032108, 2020.

[47] I. Nayak, M. Kumar, and F. L. Teixeira, "Detection and prediction of equilibrium states in kinetic plasma simulations via mode tracking using reduced-order dynamic mode decomposition," *Journal of Computational Physics*, vol. 447, p. 110671, 2021.

[48] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 2009.

[49] I. Nayak, M. Kumar, and F. L. Teixeira, "Koopman autoencoders for reduced-order modeling of kinetic plasmas," *Advances in Electromagnetics Empowered by Artificial Intelligence and Deep Learning*, pp. 515–542, 2023.

[50] I. Nayak, F. L. Teixeira, and R. J. Burkholder, "On-the-fly dynamic mode decomposition for rapid time-extrapolation and analysis of cavity resonances," *IEEE Transactions on Antennas and Propagation*, 2023.

[51] I. Nayak, F. L. Teixeira, D.-Y. Na, M. Kumar, and Y. A. Omelchenko, " Accelerating Particle-in-Cell Kinetic Plasma Simulations via Reduced-Order Modeling of Space-Charge Dynamics using Dynamic Mode Decomposition," *arXiv preprint arXiv:2303.16286*, 2023.

[52] I. Nayak, F. L. Teixeira, D.-Y. Na, M. Kumar, and Y. A. Omelchenko, "Accelerating particle-in-cell kinetic plasma simulations via reduced-order modeling of space-charge dynamics using dynamic mode decomposition," *Phys. Rev. E*, vol. 109, p. 065307, Jun 2024.

[53] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLoS one*, vol. 11, no. 2, p. e0150171, 2016.

[54] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, "Sparsity-promoting dynamic mode decomposition," *Physics of Fluids*, vol. 26, p. 024103, Feb 2014.

[55] H. Arbabi and I. Mezic, "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 4, pp. 2096–2126, 2017.

[56] S. Le Clainche and J. M. Vega, "Higher order dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 2, pp. 882–925, 2017.

[57] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 37–49, JMLR Workshop and Conference Proceedings, 2012.

[58] Q. V. Le *et al.*, "A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks," *Google Brain*, vol. 20, pp. 1–20, 2015.

[59] J. Rice, W. Xu, and A. August, "Analyzing Koopman approaches to physics-informed machine learning for long-term sea-surface temperature forecasting," *arXiv preprint arXiv:2010.00399*, 2020.

[60] M. Van Der Mheen, H. A. Dijkstra, A. Gozolchiani, M. Den Toom, Q. Feng, J. Kurths, and E. Hernandez-Garcia, "Interaction network based early warning indicators for the Atlantic moc collapse," *Geophysical Research Letters*, vol. 40, no. 11, pp. 2714–2719, 2013.

[61] J. P. Feng, Q. Y.and Viebahn and H. A. Dijkstra, "Deep ocean early warning signals of an Atlantic moc collapse," *Geophysical Research Letters*, vol. 41, no. 16, pp. 6009–6015, 2014.

[62] M. D. Chekroun, J. D. Neelin, D. Kondrashov, J. C. McWilliams, and M. Ghil, "Rough parameter dependence in climate models and the role of ruelle-pollicott resonances," *Proceedings of the National Academy of Sciences*, vol. 111, no. 5, pp. 1684–1690, 2014.

[63] A. Tantet, F. R. van der Burgt, and H. A. Dijkstra, "An early warning indicator for atmospheric blocking events using transfer operators," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 3, p. 036406, 2015.

[64] G. A. Gottwald and F. Gugole, "Detecting regime transitions in time series using dynamic mode decomposition," *Journal of Statistical Physics*, pp. 1–18, 2019.

[65] A. Alessandri, P. Bagnerini, M. Gaggero, D. Lengani, and D. Simoni, "Dynamic mode decomposition for the inspection of three-regime separated transitional boundary layers using a least squares method," *Physics of Fluids*, vol. 31, no. 4, p. 044103, 2019.

[66] D. Dylewsky, M. Tao, and J. N. Kutz, "Dynamic mode decomposition for multiscale nonlinear physics," *Phys. Rev. E*, vol. 99, p. 063311, Jun 2019.

[67] A. C. Costa, T. Ahamed, and G. J. Stephens, "Adaptive, locally linear models of complex dynamics," *Proceedings of the National Academy of Sciences*, vol. 116, no. 5, pp. 1501–1510, 2019.

[68] J. M. Kunert-Graf, K. M. Eschenburg, D. J. Galas, J. N. Kutz, S. D. Rane, and B. W. Brunton, "Extracting reproducible time-resolved resting state networks using dynamic mode decomposition," *Frontiers in Computational Neuroscience*, vol. 13, p. 75, 2019.

[69] R. Taylor, J. N. Kutz, K. Morgan, and B. A. Nelson, "Dynamic mode decomposition for plasma diagnostics and validation," *Review of Scientific Instruments*, vol. 89, no. 5, p. 053501, 2018.

[70] M. S. Hemati, M. O. Williams, and C. W. Rowley, "Dynamic mode decomposition for large and streaming datasets," *Physics of Fluids*, vol. 26, no. 11, p. 111701, 2014.

[71] 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.

[72] M. Alfatlawi and V. Srivastava, "An incremental approach to online dynamic mode decomposition for time-varying systems with applications to eeg data modeling," 2020.

[73] R. Alden and F. Qureshy, "Eigenvalue tracking due to parameter variation," *IEEE Transactions on automatic control*, vol. 30, no. 9, pp. 923–925, 1985.

[74] C. Beaverstock, M. Friswell, S. Adhikari, T. Richardson, and J. Du Bois, "Automatic mode tracking for flight dynamic analysis using a spanning algorithm," *Aerospace Science and Technology*, vol. 47, pp. 54–67, 2015.

[75] E. Safin and D. Manteuffel, "Advanced eigenvalue tracking of characteristic modes," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 7, pp. 2628–2636, 2016.

[76] B. D. Raines and R. G. Rojas, "Wideband characteristic mode tracking," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 7, pp. 3537–3541, 2012.

[77] S. M. Hirsh, K. D. Harris, J. N. Kutz, and B. W. Brunton, "Centering data improves the dynamic mode decomposition," 2019.

[78] G. W. Stewart, "Error and perturbation bounds for subspaces associated with certain eigenvalue problems," *SIAM review*, vol. 15, no. 4, pp. 727–764, 1973.

[79] G. W. Stewart, "On the perturbation of pseudo-inverses, projections and linear least squares problems," *SIAM review*, vol. 19, no. 4, pp. 634–662, 1977.

[80] F. Li, H. Liu, and R. J. Vaccaro, "Performance analysis for DOA estimation algorithms: unification, simplification, and observations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 4, pp. 1170–1184, 1993.

[81] Y. Chen, C. Cheng, and J. Fan, "Asymmetry helps: Eigenvalue and eigenvector analyses of asymmetrically perturbed low-rank matrices," 2020.

[82] A. Greenbaum, R. Li, and M. L. Overton, "First-order perturbation theory for eigenvalues and eigenvectors," 2019.

[83] S. Bagheri, "Koopman-mode decomposition of the cylinder wake," *Journal of Fluid Mechanics*, vol. 726, pp. 596–623, 2013.

[84] J. Page and R. R. Kerswell, "Koopman mode expansions between simple invariant solutions," *Journal of Fluid Mechanics*, vol. 879, pp. 1–27, 2019.

[85] G. Pascarella, I. Kokkinakis, and M. Fossati, "Analysis of transition for a flow in a channel via reduced basis methods," *Fluids*, vol. 4, no. 4, p. 202, 2019.

[86] I. Nayak, M. Kumar, and F. Teixeira, "Detecting Equilibrium State of Dynamical Systems Using Sliding-Window Reduced-Order Dynamic Mode Decomposition," in *AIAA Scitech 2021 Forum*, p. 1858, 2021.

[87] M. Sasaki, Y. Kawachi, R. O. Dendy, H. Arakawa, N. Kasuya, F. Kin, K. Yamasaki, and S. Inagaki, "Using dynamical mode decomposition to extract the limit cycle dynamics of modulated turbulence in a plasma simulation," *Plasma Physics and Controlled Fusion*, vol. 61, no. 11, p. 112001, 2019.

[88] E. N. Lorenz, "Predictability: A problem partly solved," in *Proc. Seminar on predictability*, vol. 1, pp. 1–18, 1996.

[89] E. N. Lorenz and K. A. Emanuel, "Optimal sites for supplementary weather observations: Simulation with a small model," *Journal of the Atmospheric Sciences*, vol. 55, no. 3, pp. 399–414, 1998.

[90] J. Kerin and H. Engler, "On the Lorenz '96 model and some generalizations," 2020.

[91] V. Lucarini and S. Sarno, "A statistical mechanical approach for the computation of the climatic response to general forcings," *Nonlinear Processes in Geophysics*, vol. 18, no. 1, pp. 7–28, 2011.

[92] G. Gallavotti and V. Lucarini, "Equivalence of non-equilibrium ensembles and representation of friction in turbulent flows: the Lorenz 96 model," *Journal of Statistical Physics*, vol. 156, no. 6, pp. 1027–1065, 2014.

[93] C. Amor, P. Schlatter, R. Vinuesa, and S. Le Clainche, "Higher-order dynamic mode decomposition on-the-fly: A low-order algorithm for complex fluid flows," *Journal of Computational Physics*, vol. 475, p. 111849, 2023.

[94] S. M. Hirsh, K. D. Harris, J. N. Kutz, and B. W. Brunton, "Centering data improves the dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 19, no. 3, pp. 1920–1955, 2020.

[95] S. Pan and K. Duraisamy, "On the structure of time-delay embedding in linear models of non-linear dynamical systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 7, p. 073135, 2020.

[96] F. L. Teixeira, "Time-domain finite-difference and finite-element methods for Maxwell equations in complex media," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2150–2166, 2008.

[97] J.-F. Lee, R. Lee, and A. Cangellaris, "Time-domain finite-element methods," *IEEE transactions on antennas and propagation*, vol. 45, no. 3, pp. 430–442, 1997.

[98] R. S. Adve, T. K. Sarkar, O. M. Pereira-Filho, and S. M. Rao, "Extrapolation of time-domain responses from three-dimensional conducting objects utilizing the matrix pencil technique," *IEEE transactions on antennas and propagation*, vol. 45, no. 1, pp. 147–156, 1997.

[99] F. Teixeira, "A summary review on 25 years of progress and future challenges in FDTD and fetd techniques," *The Applied Computational Electromagnetics Society Journal (ACES)*, pp. 1–14, 2010.

[100] T. Namiki, "A new FDTD algorithm based on alternating-direction implicit method," *IEEE Transactions on Microwave Theory and techniques*, vol. 47, no. 10, pp. 2003–2007, 1999.

[101] F. Zheng, Z. Chen, and J. Zhang, "A finite-difference time-domain method without the courant stability conditions," *IEEE Microwave and Guided wave letters*, vol. 9, no. 11, pp. 441–443, 1999.

[102] J. Lee and B. Fornberg, "Some unconditionally stable time stepping methods for the 3d Maxwell's equations," *Journal of Computational and Applied Mathematics*, vol. 166, no. 2, pp. 497–523, 2004.

[103] E. L. Tan, "Unconditionally stable LOD–FDTD method for 3-d Maxwell's equations," *IEEE Microwave and Wireless Components Letters*, vol. 17, no. 2, pp. 85–87, 2007.

[104] G. Sun and C. Trueman, "Unconditionally-stable FDTD method based on crank-nicolson scheme for solving three-dimensional Maxwell equations," *Electronics Letters*, vol. 40, no. 10, pp. 589–590, 2004.

[105] M. Gaffar and D. Jiao, "An explicit and unconditionally stable FDTD method for electromagnetic analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 11, pp. 2538–2550, 2014.

[106] X. Zhang, F. Bekmambetova, and P. Triverio, "A stable FDTD method with embedded reduced-order models," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 2, pp. 827–837, 2017.

[107] S. B. Sayed, H. A. Ülkü, and H. Bağcı, "A stable marching on-in-time scheme for solving the time-domain electric field volume integral equation on high-contrast scatterers," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 7, pp. 3098–3110, 2015.

[108] Z. Luo and J. Gao, "A POD reduced-order finite difference time-domain extrapolating scheme for the 2D Maxwell equations in a lossy medium," *Journal of Mathematical Analysis and Applications*, vol. 444, no. 1, pp. 433–451, 2016.

[109] V. Jandhyala, E. Michielssen, and R. Mittra, "FDTD signal extrapolation using the forward-backward autoregressive (AR) model," *IEEE Microw. Guided Wave Lett.*, vol. 4, no. 6, pp. 163–165, 1994.

[110] A. K. Shaw and K. Naishadham, "ARMA-based time-signature estimator for analyzing resonant structures by the FDTD method," *IEEE Transa. Ant. Propagat.*, vol. 49, no. 3, pp. 327–339, 2001.

[111] W. L. Ko and R. Mittra, "A combination of FD-TD and Prony's methods for analyzing microwave integrated circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 39, no. 12, pp. 2176–2181, 1991.

[112] P. Kozakowski and M. Mrozowski, "Low-order models from FD-TD time samples," *IEEE Microw. Wireless Comp. Lett.*, vol. 12, no. 11, pp. 438–440, 2002.

[113] T. K. Sarkar, S. Park, J. Koh, and S. M. Rao, "Application of the matrix pencil method for estimating the sem (singularity expansion method) poles of source-free transient responses from multiple look directions," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 4, pp. 612–618, 2000.

[114] M. L. Burrows, "Two-dimensional esprit with tracking for radar imaging and feature extraction," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 2, pp. 524–532, 2004.

[115] A. N. Lemma, A.-J. Van Der Veen, and E. F. Deprettere, "Analysis of joint angle-frequency estimation using esprit," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1264–1283, 2003.

[116] S.-Y. Kung, K. S. Arun, and D. B. Rao, "State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem," *JOSA*, vol. 73, no. 12, pp. 1799–1811, 1983.

[117] K. Naishadham and J. E. Piou, "Representation of electromagnetic responses in time domain using state-space system identification method," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 4, pp. 1404–1415, 2016.

[118] J.-N. Juang and R. S. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of guidance, control, and dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[119] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, "Chaos as an intermittently forced linear system," *Nature communications*, vol. 8, no. 1, pp. 1–9, 2017.

[120] L. Pogorelyuk and C. W. Rowley, "Clustering of series via dynamic mode decomposition and the matrix pencil method," *arXiv preprint arXiv:1802.09878*, 2018.

[121] Y. Susuki, I. Mezic, F. Raak, and T. Hikihara, "Applied Koopman operator theory for power systems technology," *Nonlinear Theory and Its Applications, IEICE*, vol. 7, no. 4, pp. 430–459, 2016.

[122] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Transactions on antennas and propagation*, vol. 14, no. 3, pp. 302–307, 1966.

[123] F. L. Teixeira, W. C. Chew, M. Straka, M. Oristaglio, and T. Wang, "Finite-difference time-domain simulation of ground penetrating radar on dispersive, inhomogeneous, and conductive soils," *IEEE Transactions on Geoscience and remote sensing*, vol. 36, no. 6, pp. 1928–1937, 1998.

[124] F. L. Teixeira, C. D. Moss, W. C. Chew, and J. A. Kong, "Split-field and anisotropic-medium PML-FDTD implementations for inhomogeneous media," *IEEE transactions on microwave theory and techniques*, vol. 50, no. 1, pp. 30–35, 2002.

[125] Y.-K. Hue and F. L. Teixeira, "FDTD simulation of mwd electromagnetic tools in large-contrast geophysical formations," *IEEE transactions on magnetics*, vol. 40, no. 2, pp. 1456–1459, 2004.

[126] A. Taflove and S. C. Hagness, "Finite-difference time-domain solution of Maxwell's equations," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–33, 1999.

[127] H. O. Lee, *Numerical Modeling of Electromagnetic Well-Logging Sensors*. PhD thesis, The Ohio State University, 2010.

[128] R. E. Collin, *Electromagnetic Resonators*, pp. 481–549. 2001.

[129] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, 2014.

[130] M. Gavish and D. L. Donoho, "Matlab code: The optimal hard threshold for singular values is $4/\sqrt{3}$."

[131] H. Lu and D. M. Tartakovsky, "Prediction accuracy of dynamic mode decomposition," *SIAM Journal on Scientific Computing*, vol. 42, no. 3, pp. A1639–A1662, 2020.

[132] H. Moon, F. L. Teixeira, and Y. A. Omelchenko, "Exact charge-conserving scatter-gather algorithm for particle-in-cell simulations on unstructured grids: A geometric perspective," *Comput. Phys. Commun*, vol. 194, pp. 43–53, 2015.

[133] J. Kim and F. L. Teixeira, "Parallel and explicit finite-element time-domain method for Maxwell's equations," *IEEE Trans. Antennas Propag.*, vol. 59, pp. 2350–2356, Jun 2011.

[134] E. Evstatiev and B. Shadwick, "Variational formulation of particle algorithms for kinetic plasma simulations," *Journal of Computational Physics*, vol. 245, pp. 376–398, 2013.

[135] J. Squire, H. Qin, and W. M. Tang, "Geometric integration of the Vlasov-Maxwell system with a variational particle-in-cell scheme," *Physics of Plasmas*, vol. 19, no. 8, p. 084501, 2012.

[136] X. Jianyuan, Q. Hong, and L. Jian, "Structure-preserving geometric particle-in-cell methods for Vlasov-Maxwell systems," *Plasma Science and Technology*, vol. 20, no. 11, p. 110501, 2018.

[137] N. Meyer-Vernet, "Aspects of Debye shielding," *American journal of physics*, vol. 61, no. 3, pp. 249–257, 1993.

[138] D.-Y. Na, J. Nicolini, R. Lee, B.-H. Borges, Y. Omelchenko, and F. Teixeira, "Diagnosing numerical cherenkov instabilities in relativistic plasma simulations based on general meshes," *Journal of Computational Physics*, vol. 402, p. 108880, 2020.

[139] J. W. Eastwood, "The virtual particle electromagnetic particle-mesh method," *Computer Physics Communications*, vol. 64, no. 2, pp. 252–266, 1991.

[140] B. Marder, "A method for incorporating gauss' law into electromagnetic pic codes," *Journal of Computational Physics*, vol. 68, no. 1, pp. 48–55, 1987.

[141] M. C. Pinto, S. Jund, S. Salmon, and E. Sonnendrücker, "Charge-conserving FEM–PIC schemes on general grids," *Comptes Rendus Mecanique*, vol. 342, no. 10-11, pp. 570–582, 2014.

[142] F. L. Teixeira, "Lattice Maxwell's equations," *Prog. Electromagn. Res.*, vol. 148, pp. 113–128, 2014.

[143] D. Na, H. Moon, Y. A. Omelchenko, and F. L. Teixeira, "Local, explicit, and charge-conserving electromagnetic particle-in-cell algorithm on unstructured grids," *IEEE Transactions on Plasma Science*, vol. 44, no. 8, pp. 1353–1362, 2016.

[144] D.-Y. Na, H. Moon, Y. A. Omelchenko, and F. L. Teixeira, "Relativistic extension of a charge-conservative finite element solver for time-dependent Maxwell-Vlasov equations," *Phys. Plasmas*, vol. 25, p. 013109, Jan 2018.

[145] K. F. Warnick, R. H. Selfridge, and D. V. Arnold, "Teaching electromagnetic field theory using differential forms," *IEEE Transactions on education*, vol. 40, no. 1, pp. 53–68, 1997.

[146] F. L. Teixeira and W. C. Chew, "Lattice electromagnetic theory from a topological viewpoint," *J. Math. Phys.*, vol. 40, no. 1, pp. 169–187, 1999.

[147] A. Bossavit, "Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism," *IEE Proceedings A-Physical Science, Measurement and Instrumentation, Management and Education-Reviews*, vol. 135, no. 8, pp. 493–500, 1988.

[148] B. He and F. L. Teixeira, "Geometric finite element discretization of Maxwell equations in primal and dual spaces," *Phys. Lett. A*, vol. 349, pp. 1–14, 2006.

[149] B. He and F. L. Teixeira, "Differential forms, galerkin duality, and sparse inverse approximations in finite element solutions of Maxwell equations," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 5, pp. 1359–1368, 2007.

[150] M. Clemens and T. Weiland, "Discrete electromagnetism with the finite integration technique," *Progress In Electromagnetics Research*, vol. 32, pp. 65–87, 2001.

[151] R. Schuhmann and T. Weiland, "Conservation of discrete energy and related laws in the finite integration technique," *Progress In Electromagnetics Research*, vol. 32, pp. 301–316, 2001.

[152] E. M. Wolf, M. Causley, A. Christlieb, and M. Bettencourt, "A particle-in-cell method for the simulation of plasmas based on an unconditionally stable field solver," *Journal of Computational Physics*, vol. 326, pp. 342 – 372, 2016.

[153] M. Horký, W. J. Miloch, and V. A. Delong, "Numerical heating of electrons in particle-in-cell simulations of fully magnetized plasmas," *Phys. Rev. E*, vol. 95, p. 043302, Apr 2017.

[154] I. Nayak, M. Kumar, and F. L. Teixeira, "Detection and prediction of equilibrium states in kinetic plasma simulations via mode tracking using reduced-order dynamic mode decomposition," *Journal of Computational Physics*, vol. 447, p. 110671, 2021.

[155] Y. Wang, J. Wang, Z. Chen, G. Cheng, and P. Wang, "Three-dimensional simple conformal symplectic particle-in-cell methods for simulations of high power microwave devices," *Computer Physics Communications*, vol. 205, pp. 1–12, 2016.

[156] D.-Y. Na, Y. A. Omelchenko, H. Moon, B.-H. V. Borges, and F. L. Teixeira, "Axisymmetric charge-conservative electromagnetic particle simulation algorithm on unstructured grids: Application to microwave vacuum electronic devices," *J. Comput. Phys.*, vol. 346, pp. 295 – 317, 2017.

[157] V. K. Decyk and T. V. Singh, "Particle-in-cell algorithms for emerging computer architectures," *Computer Physics Communications*, vol. 185, no. 3, pp. 708–719, 2014.

[158] T. Sayadi and P. J. Schmid, "Parallel data-driven decomposition algorithm for large-scale datasets: with application to transitional boundary layers," *Theoretical and Computational Fluid Dynamics*, vol. 30, no. 4, pp. 415–428, 2016.

[159] K. Maryada and S. Norris, "Reduced-communication parallel dynamic mode decomposition," *Journal of Computational Science*, vol. 61, p. 101599, 2022.

[160] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International Conference on Machine Learning*, pp. 5301–5310, PMLR, 2019.

[161] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, "Frequency principle: Fourier analysis sheds light on deep neural networks," *arXiv preprint arXiv:1901.06523*, 2019.

[162] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[163] G. G. Towell and J. W. Shavlik, "Extracting refined rules from knowledge-based neural networks," *Machine learning*, vol. 13, no. 1, pp. 71–101, 1993.

[164] F. Wang and Q.-J. Zhang, "Knowledge-based neural models for microwave design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 12, pp. 2333–2343, 1997.

[165] J. Ghaboussi, J. H. Garrett Jr, and X. Wu, "Knowledge-based modeling of material behavior with neural networks," *Journal of engineering mechanics*, vol. 117, no. 1, pp. 132–153, 1991.

[166] M. O. Noordewier, G. G. Towell, and J. W. Shavlik, "Training knowledge-based neural networks to recognize genes in dna sequences," in *Advances in neural information processing systems*, pp. 530–536, 1991.

[167] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[168] Q. Marashdeh, W. Warsito, L.-S. Fan, and F. Teixeira, "Nonlinear forward problem solution for electrical capacitance tomography using feed-forward neural network," *IEEE Sensors Journal*, vol. 6, no. 2, pp. 441–449, 2006.

[169] A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided neural networks (pgnn): An application in lake temperature modeling," *arXiv preprint arXiv:1710.11431*, 2017.

[170] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[171] S. Wang, X. Yu, and P. Perdikaris, " When and why PINNs fail to train: A neural tangent kernel perspective," *Journal of Computational Physics*, vol. 449, p. 110768, 2022.