Reduced-Order Modeling with Deep Learning: Flow Reconstruction for External Aerodynamics and Natural Convection

Tyler Jones

University of Wisconsin-Madison

B.S. Applied Mathematics, Engineering, and Physics

May 4, 2025

Abstract

In this paper, we present a workflow that combines the finite difference method in computational fluid dynamics, proper orthogonal decomposition for model reduction, and a dense neural network to build surrogates for turbulent external aerodynamics and natural convection. High-resolution – $\mathcal{O}(10^4)$ degrees of freedom – simulations of a circular cylinder wake, NACA 2412 airfoil, and Rayleigh–Bénard convection are first compressed to $\mathcal{O}(10^3)$ POD modes that are enforced to retain 99% of the flow energy. A multilayer perceptron is then trained to map velocity–magnitude snapshots to modal amplitudes; projected back onto the POD basis, these amplitudes reconstruct full velocity and temperature fields with under 2% mean-squared error and reduce the wall-clock time from the original CFD by a factor of $T_{reduced}$ (still needs to be timed). Although the project needs further analyses, preliminary results have demonstrated that POD-constrained deep learning can deliver accurate, low-cost digital twins.

Contents

1 Introduction

Advancements in aerospace engineering increasingly require rapid, yet physically accurate, computational tools for iterative design and real-time analysis. While highly accurate with delicate treatment, traditional computational fluid dynamics (CFD) methods often incur prohibitive computational costs—as stressed in courses like ME 573 and CS 412. This study integrates CFD simulations with machine learning-based reduced-order modeling techniques to address this challenge. Specifically, we employ Proper Orthogonal Decomposition (POD) to compress simulation data from a few canonical flow problems—external flow past a circular cylinder, aerodynamic behavior of a NACA 2412 airfoil, and buoyancy-driven Rayleigh–Bénard convection—into a reduced set of modes. Subsequently, a multilayer perceptron neural network reconstructs the original velocity and temperature fields from these reduced representations. Our results demonstrate that this POD-constrained deep learning framework achieves high reconstruction accuracy while significantly reducing computational time, thus emphasizing its potential for practical aerospace applications, such as rapid parameter optimization and active flow control.

2 Governing Equations

In this section, we introduce the fundamental equations governing Newtonian fluid motion, which form the basis for understanding turbulent—yet **deterministic**—chaotic behavior. Starting from first principles, we discuss the conservation laws of mass, momentum, and energy. Specifically, we present the incompressible Navier–Stokes equations for fluid dynamics, explore their inviscid Euler limit, and derive the vorticity transport formulation. Finally, we incorporate thermal effects by deriving the Rayleigh–Bénard convection model under the Boussinesq approximation. These equations collectively describe the analysis and simulation of the canonical flow problems investigated in this study.

2.1 Incompressible Navier–Stokes Equations

Let $\mathbf{u}(\mathbf{x}, t) = [u, v, w]^\mathsf{T}$ denote the velocity field and $p(\mathbf{x}, t)$ the *static* pressure of a Newtonian fluid with constant density ρ and dynamic viscosity μ. Under the incompressibility assumption, mass conservation reduces to

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

implying that fluid parcels neither accumulate nor deplete mass.

Momentum conservation balances inertial, pressure, viscous and body–force effects:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{f}, \tag{2}$$

where $\mathbf{f}$ is a prescribed body force per unit volume (e.g. buoyancy).

- $\rho\,\partial\mathbf{u}/\partial t$ — local (unsteady) inertia,
- $\rho(\mathbf{u}\cdot\nabla)\mathbf{u}$ — convective inertia,
- $-\nabla p$ — pressure gradient that enforces the incompressibility constraint,
- $\mu\nabla^2\mathbf{u}$ — viscous diffusion of momentum,
- $\mathbf{f}$ — external body forces.

Together, (1)–(2) constitute a closed, highly nonlinear system governing single-phase, Newtonian incompressible flow. All subsequent simplifications—such as the Euler limit, the vorticity transport form, and the Boussinesq approximation—are derived from this formulation.

2.2 Euler Limit for Inviscid Flow

When viscous stresses are negligible ($\mu \to 0$)—a somewhat fair approximation at extremely high (infinite) Reynolds numbers outside boundary and shear layers—the incompressible Navier–Stokes equations reduce to the *Euler equations*. The continuity constraint remains

$$\nabla\cdot\mathbf{u} = 0, \tag{3}$$

while momentum simplifies to

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = -\nabla p + \mathbf{f}, \tag{4}$$

with $\mathbf{f}$ a body force per unit volume.

- With $\mu\nabla^2\mathbf{u}$ absent, momentum transport occurs exclusively through convection and pressure gradients.
- For steady flows in which $\nabla\times\mathbf{f} = \mathbf{0}$ and the boundaries perform no work, the Euler system conserves total mechanical energy; viscous dissipation is identically zero.

- In any *steady* Euler flow, the relation $p + \frac{1}{2}\rho|\mathbf{u}|^2 = \text{const}$ holds *along each streamline*. If the flow is also irrotational ($\boldsymbol{\omega} = \mathbf{0}$), the Bernoulli constant is uniform throughout the domain.

In cylinder and airfoil cases, we approximate the outer flow with the inviscid Euler equations, recognizing that boundary-layer physics must be handled by separate, viscous models. The resulting simplification offers a favorable trade-off between computational speed and the accuracy required for reduced-order modeling and machine-learning reconstruction. Viscous solvers tailored to complex geometries—such as the finite volume method—are more appropriate in many practical settings, but their implementation lies beyond the scope of ME 573 and is therefore not pursued here.

2.3 Vorticity Transport Formulation

The vorticity field,

$$\boldsymbol{\omega} \;=\; \nabla \times \mathbf{u},$$

represents the local rotation of fluid parcels and is especially useful for analyzing shear-layer instabilities and coherent structures. Taking the curl of the incompressible Navier–Stokes equations yields the *vorticity transport equation*

$$\frac{\partial \boldsymbol{\omega}}{\partial t} \;+\; (\mathbf{u}\cdot\nabla)\,\boldsymbol{\omega} \;=\; (\boldsymbol{\omega}\cdot\nabla)\,\mathbf{u} \;+\; \nu\,\nabla^2\boldsymbol{\omega} \;+\; \nabla\times\mathbf{f}, \tag{5}$$

whose right-hand-side terms admit a clear physical decomposition:

- $(\boldsymbol{\omega}\cdot\nabla)\mathbf{u}$ — *vortex stretching*, responsible for amplifying or re-orienting vortical tubes,
- $\nu\nabla^2\boldsymbol{\omega}$ — *viscous diffusion* that smooths sharp vorticity gradients (vanishes in the Euler limit $\nu \to 0$),
- $\nabla \times \mathbf{f}$ — *baroclinic or body-force torque* that can generate vorticity (e.g. buoyancy in Rayleigh–Bénard convection).

For the two-dimensional cylinder benchmark, $\mathbf{u} = (u, v, 0)$ and only the out-of-plane component ω_z remains; the stretching term then drops out, simplifying (5) to

$$\frac{\partial \omega_z}{\partial t} + u\frac{\partial \omega_z}{\partial x} + v\frac{\partial \omega_z}{\partial y} \;=\; \nu\nabla^2\omega_z + \big(\nabla\times\mathbf{f}\big)_z.$$

Although vorticity is not explicitly advanced in the semi-Lagrangian solver, equation (5) supports the derived diagnostics computed in this study—namely, vortex identification and visualization.

2.4 Pressure–Poisson Equation and Projection Method

Time marching proceeds in two stages: *(i)* an explicit semi–Lagrangian advection step produces a tentative velocity $\mathbf{u}_{tent}$ that violates incompressibility, *(ii)* a pressure correction projects this field onto the divergence–free manifold according to the Helmholtz–Hodge (vector) decomposition theorem.

Pressure–Poisson Equation. Taking the divergence of the momentum equation,

$$\rho \frac{\mathbf{u}_{tent} - \mathbf{u}^n}{\Delta t} = -\nabla p^{n+\frac{1}{2}},$$

and invoking the discrete continuity constraint $\nabla \cdot \mathbf{u}^{n+1} = 0$ gives the elliptic problem

$$\nabla^2 p^{n+\frac{1}{2}} = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}_{tent}. \tag{6}$$

Equation (6) is the *pressure–Poisson equation*; its right–hand side quantifies how much the tentative velocity field violates mass conservation.

Velocity projection. Once $p^{n+\frac{1}{2}}$ is obtained, the corrected velocity is

$$\mathbf{u}^{n+1} = \mathbf{u}_{tent} - \frac{\Delta t}{\rho} \nabla p^{n+\frac{1}{2}}, \tag{7}$$

which satisfies $\nabla \cdot \mathbf{u}^{n+1} = 0$.

Numerical treatment.

- The Laplacian and gradient in (6)–(7) are discretized with second–order central differencing.
- Equation (6) is solved by Jacobi relaxation using the five-point stencil

$$p_{i,j}^{k+1} = \tfrac{1}{4}\big(p_{i+1,j}^{k} + p_{i-1,j}^{k} + p_{i,j+1}^{k} + p_{i,j-1}^{k} - h^2 R_{i,j}\big) \tag{8}$$

  where $R_{i,j}$ is the residual and h the uniform grid spacing.
- Homogeneous Neumann conditions ($\partial p/\partial n = 0$) are applied on the far-field inflow/outflow boundaries, while Dirichlet conditions enforce prescribed pressure differences near the cylinder mask. In dedicated CFD solvers, it is best practice to alternate boundary conditions on the pressure and velocity fields for stability—in hopes of preventing pressure and velocity decoupling.

This projection step attempts to remove compressibility errors introduced by the semi–Lagrangian advection, ensuring that each snapshot stored for POD and machine-learning reconstruction respects the incompressible flow constraint.

2.5 Energy Equation: Scalar Advection–Diffusion of Temperature

For an incompressible fluid with constant specific heat c_p and thermal conductivity k, the temperature field $T(\mathbf{x}, t)$ evolves according to the advection–diffusion (energy) equation

$$\rho c_p \Big(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \Big) \;=\; k \nabla^2 T + q \,, \tag{9}$$

where $q(\mathbf{x}, t)$ denotes a volumetric heat source (taken $q = 0$ in this project). Dividing by ρc_p and introducing the thermal diffusivity $\alpha = k/(\rho c_p)$ gives

$$\frac{\partial T}{\partial t} \;+\; \mathbf{u} \cdot \nabla T \;=\; \alpha \, \nabla^2 T.$$

- $\partial T/\partial t$ — local, unsteady storage of thermal energy,
- $\mathbf{u} \cdot \nabla T$ — convective transport by the flow,
- $\alpha \nabla^2 T$ — diffusion that smooths temperature gradients.

In the Rayleigh–Bénard case, temperature differences across the horizontal plates drive buoyancy and are coupled back into the momentum equation via the Boussinesq term (see subsection 2.6). Boundary conditions are Dirichlet on the hot ($T = T_{\mathrm{h}}$) and cold ($T = T_{\mathrm{c}}$) walls. The east and west walls are no-slip but thermally insulated (adiabatic).

2.6 Non-Dimensionalization and Characteristic Numbers

Scaling. Let L be the cavity height, U a characteristic velocity, $\Delta T = T_{\mathrm{hot}} - T_{\mathrm{cold}}$ the imposed temperature difference, and $H \equiv L$. One can easily recover the dimensionless variables in terms of the characteristic and dimensional variables:

$$\mathbf{x}^* = \frac{\mathbf{x}}{L}, \quad t^* = \frac{Ut}{L}, \quad \mathbf{u}^* = \frac{\mathbf{u}}{U}, \quad p^* = \frac{p}{\rho U^2}, \quad T^* = \frac{T - T_{\mathrm{cold}}}{\Delta T}.$$

Dimensionless equations. Substituting these scalings into the incompressible Navier–Stokes

and energy equations yields

$$\nabla^* \cdot \mathbf{u}^* = 0, \tag{10a}$$

$$\frac{\partial \mathbf{u}^*}{\partial t^*} + (\mathbf{u}^* \cdot \nabla^*)\mathbf{u}^* = -\nabla^* p^* + \frac{1}{Re}\nabla^{*2}\mathbf{u}^* + \frac{Ra}{Pr\,Re^2}\,T^*\,\mathbf{e}_y, \tag{10b}$$

$$\frac{\partial T^*}{\partial t^*} + (\mathbf{u}^* \cdot \nabla^*)T^* = \frac{1}{Re\,Pr}\nabla^{*2}T^*, \tag{10c}$$

where three characteristic numbers appear naturally:

$$Re = \frac{UL}{\nu}\,, \qquad Pr = \frac{\nu}{\alpha}\,, \qquad Ra = \frac{g\beta\Delta T\,H^3}{\nu\alpha}\,.$$

- *Reynolds number* Re measures the ratio of inertial to viscous forces. The Euler limit adopted in the outer-flow solver corresponds to $Re \to \infty$.
- *Prandtl number* Pr compares momentum diffusivity ν with thermal diffusivity α. Air at room temperature has $Pr \approx 0.7$.
- *Rayleigh number* Ra measures buoyancy-driven destabilization against viscous and thermal damping.

Taken together, the continuity, momentum, vorticity, and temperature equations in their non-dimensional forms provide the complete thermo–fluid framework for this study. Each relation depends only on the characteristic parameters Re, Pr, and Ra, which therefore span the input space for the forthcoming semi-Lagrangian CFD solver and the POD-based, deep-learning model. Having established the governing physics, we next describe the numerical methodology employed to integrate these equations with both stability and computational efficiency.

3 Computational Fluid Dynamics

In this section, the governing equations are discretized and numerically integrated to generate the data that fuels the subsequent reduced-order and machine-learning stages. Two flow classes are treated: (i) inviscid, incompressible external aerodynamics around a circular cylinder and a NACA 2412 airfoil, and (ii) viscid, buoyancy-driven Rayleigh–Bénard convection in a cavity. Although the physics differ, both solvers share second-order finite differences on a Cartesian mesh, semi-Lagrangian advection for stability at large time steps, and a pressure or stream-function Poisson solver to enforce incompressibility. The following subsections summarize the numerical algorithms, boundary conditions, data-handling strategies, and post-processing results specific to each configuration.

3.1 Inviscid and Incompressible Flow Around a Cylinder/Airfoil

The cylinder and NACA 2412 airfoil wind tunnel simulations employ an explicit *semi-Lagrangian finite-difference projection scheme* closely related to Stam's "Stable-Fluids" algorithm, adapted here for incompressible, inviscid aerodynamics on a uniform Cartesian grid. The key algorithmic steps are summarized as follows.

1. **Spatial discretization.** Velocity components (v_x, v_y) and kinematic pressure p are stored at the collocated cell centers of an $(AR \times N_y) \times N_y$ grid ($AR = 3$ in this project to capture vortex shedding downstream). Spatial derivatives are approximated via central differences, and the nonlinear advection term utilizes an upwind scheme to preserve stability—as stressed in ME 573.

2. **Geometric immersion.** The bluff (cylinder) and streamlined (airfoil) obstacles are inserted via a Boolean mask $\chi(\mathbf{x})$: $\chi = 1$ inside the body and 0 elsewhere. Velocities are reset to zero wherever $\chi = 1$, enforcing an impermeable, no-slip surface without body-fitted meshing. For the airfoil case, the mask is generated from an analytic NACA 2412 profile for various angles of attack.

3. **Semi-Lagrangian advection.** Each time step traces grid points backwards along particle paths using a fourth-order Runge–Kutta integration scheme,

$$\mathbf{x}^* = \mathbf{x}^n - \int_{t^n}^{t^n - \Delta t} \mathbf{u}(\mathbf{x}, t)\, dt,$$

   followed by bilinear interpolation (via `RegularGridInterpolator`) to obtain the tentative field $\mathbf{u}_{tent}$. This procedure is unconditionally stable with respect to the Courant number, allowing for relatively large time steps.

4. **Pressure–Poisson projection.** Divergence of the tentative velocity defines the right-hand side of the discrete Poisson problem. The five-point Laplacian is solved by Jacobi relaxation (100 sweeps per step). As a side note for future work, a successive overrelaxation (SOR) algorithm with residual monitoring (adding a short-circuit monitor) can be employed to speed up convergence and thus more accurate results are maintainable with similar computational efficiency—as practiced in ME 573.

5. **Tracer integration and diagnostics.** Massless particles are advanced with the same RK4 scheme to visualize trajectories.

6. **Data management.** At defined time step intervals, we store (v_x, v_y, p) snapshots in compressed HDF5 format and appends time histories. These snapshots form the dataset for subsequent Proper Orthogonal Decomposition and neural-network training.

3.1.1 Post-Processing: Cylinder

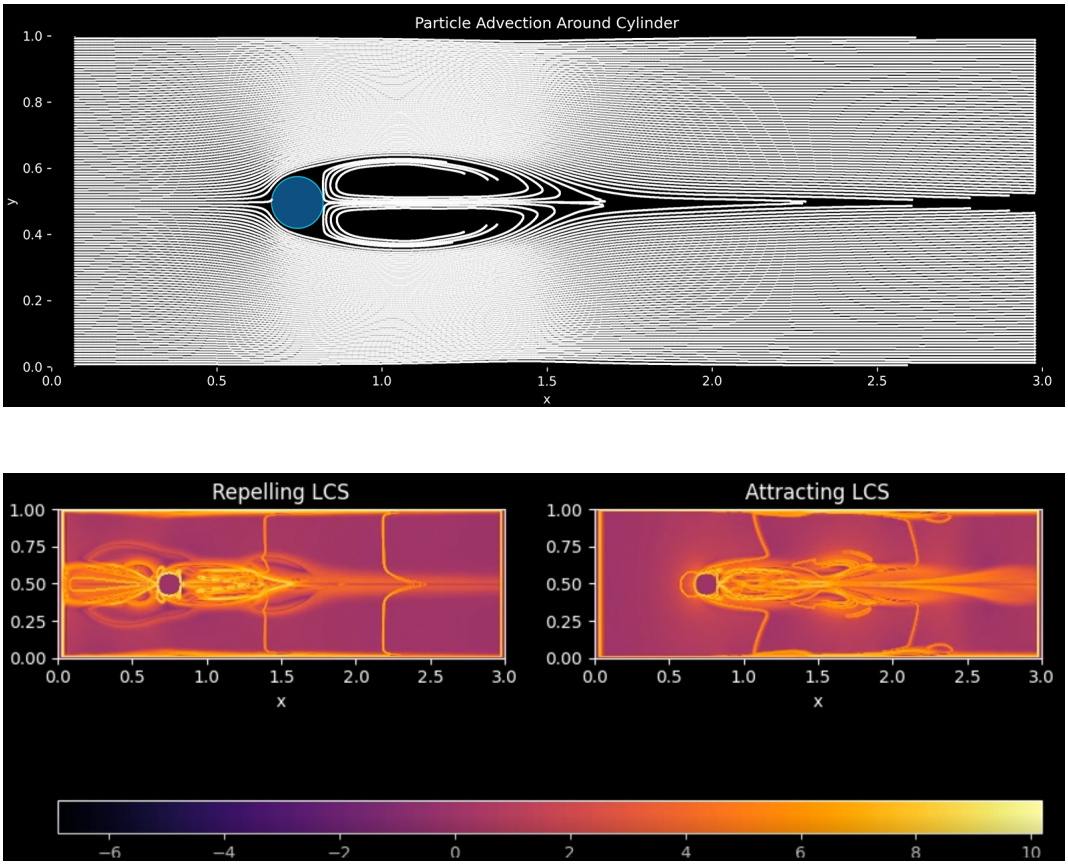

Figure 1: Instantaneous flow topology around a circular cylinder with the onset of flow separation. **Top**: particle-advection map from the inviscid, incompressible simulation, showing shear-layer separation, recirculation, and wake jet. **Bottom**: Finite-Time Lyapunov Exponent fields for the forward-time FTLE (left) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (right) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

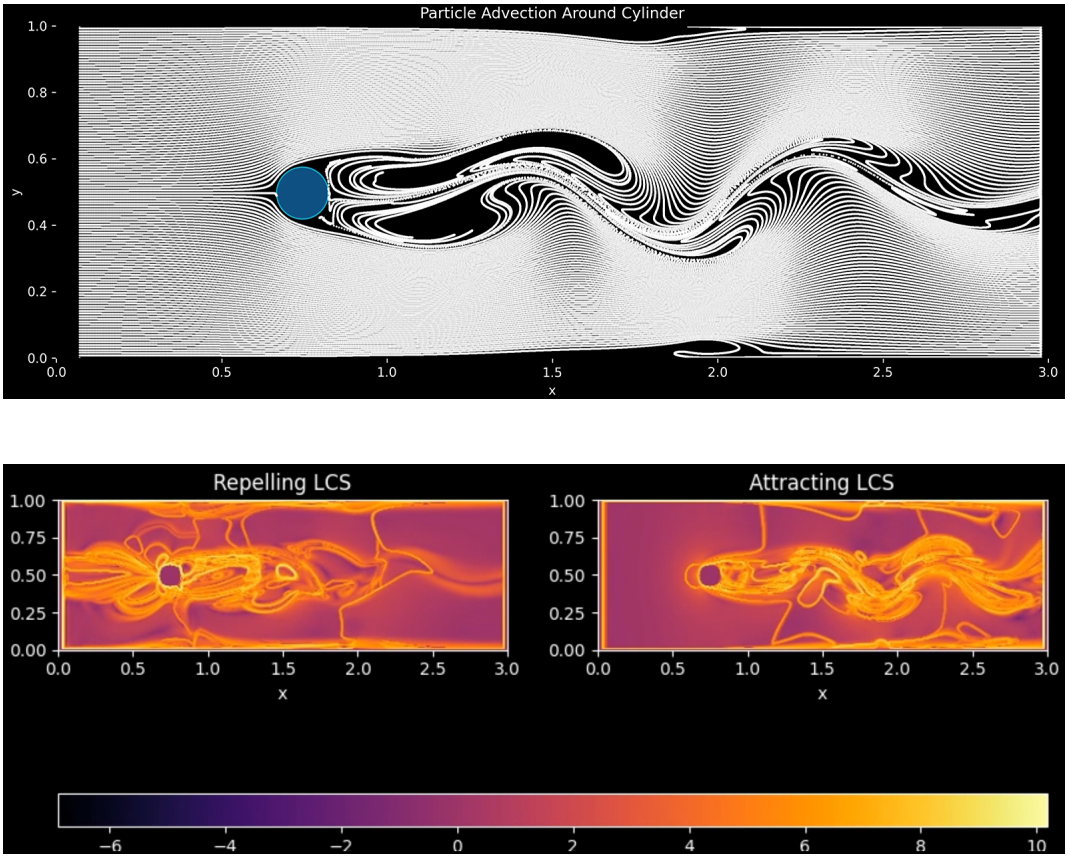

Figure 2: Instantaneous flow topology around a circular cylinder with the onset of vortex shedding. **Top**: particle-advection map from the inviscid, incompressible simulation. **Bottom**: Finite-Time Lyapunov Exponent fields for the forward-time FTLE (left) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (right) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

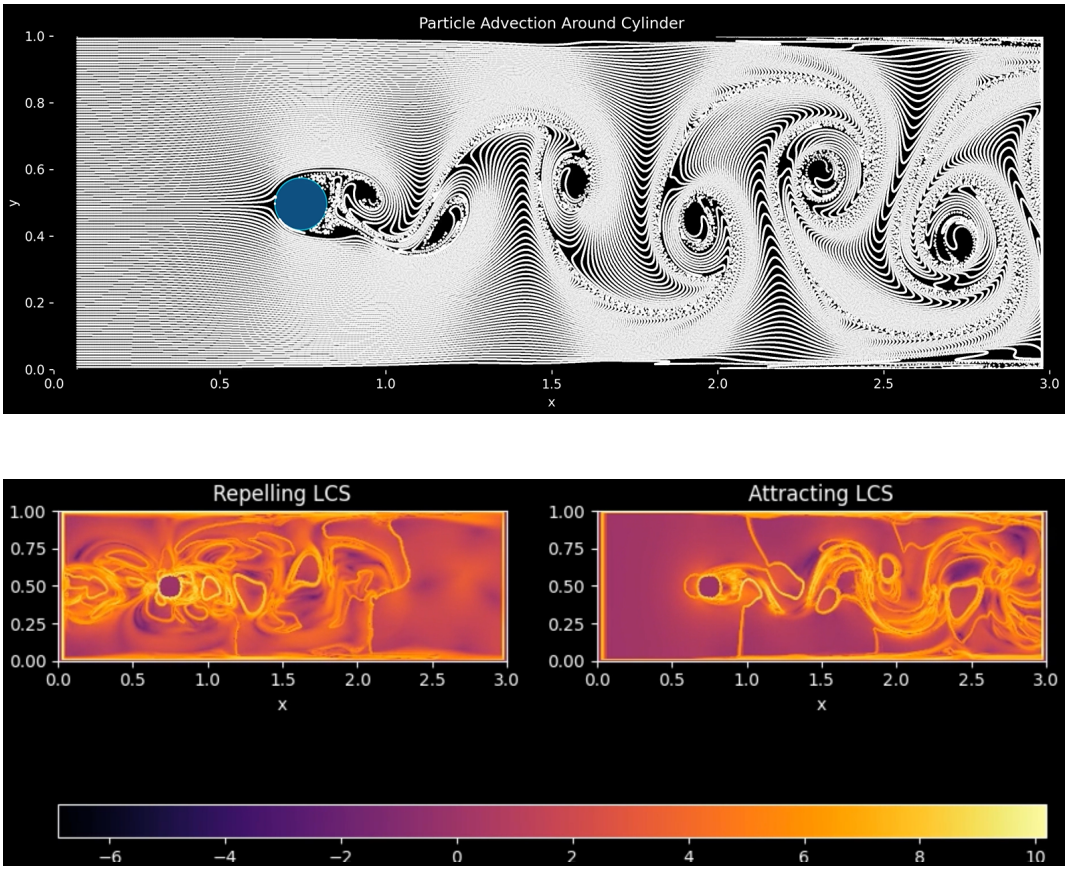

Figure 3: Instantaneous flow topology around a circular cylinder displaying the Kármán vortex street. **Top**: particle-advection map from the inviscid, incompressible simulation. **Bottom**: Finite-Time Lyapunov Exponent fields for the forward-time FTLE (left) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (right) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

3.1.2 Post-Processing: NACA 2412 Airfoil

Due to time constraints and unsatisfactory FTLE visualizations in the cylinder case, only particle-advection snapshots were captured. For future work, a more rigorous FTLE computation will be implemented; see Brunton and Kutz [2] for recommended algorithms.

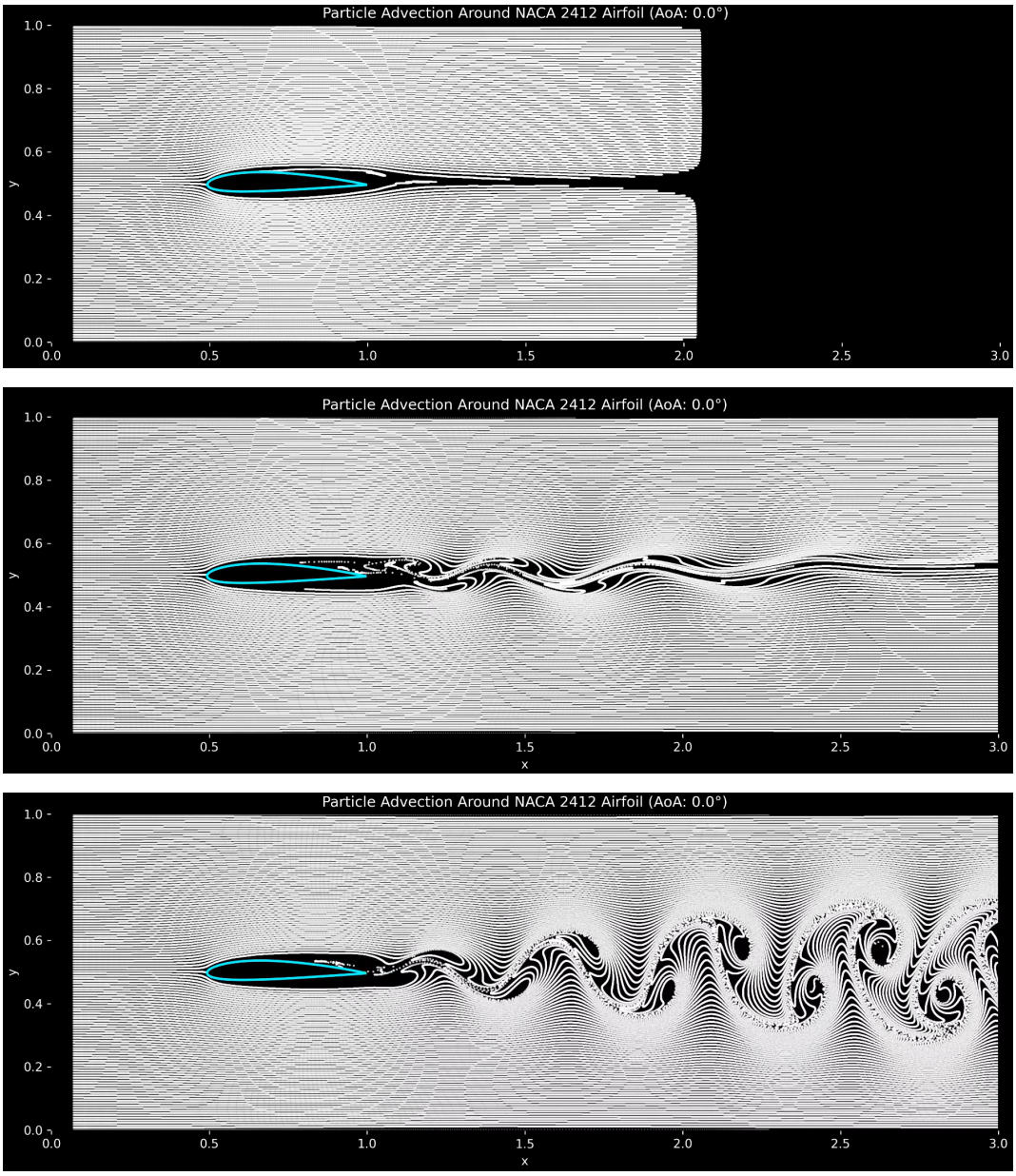

Figure 4: Instantaneous particle-advection maps around a NACA 2412 airfoil (AoA=0°) illustrating the wake evolution. **Top**: early stage showing nearly attached shear layers and a potential-flow-like field. **Middle**: transitional regime in which the shear layers roll up and alternating vortex patches begin to form. **Bottom**: fully developed Kármán-type vortex street with periodic shedding and pronounced entrainment.

3.2 Rayleigh–Bénard Convection

The thermally driven cavity is solved with the vorticity stream-function formulation that mirrors the physics outlined in subsection 2.6. The algorithm–covered in Glatzmaier [3]–proceeds as follows:

1. **Spatial Discretization.** All fields are stored on a uniform $(AR \times N_y) \times (N_y)$ Cartesian mesh ($AR = 3$ to capture more cell formations). Standard finite differencing is used to approximate derivatives.

2. **Poisson solve for the stream-function.** The incompressibility constraint $\nabla^2 \psi = -\omega$ is discretized with a five-point stencil; the corresponding sparse matrix is assembled *once* (`build_poisson_matrix`) and factorized on the fly by `spsolve`. This direct sparse solve is $\mathcal{O}(N)$ per step.

3. **Explicit time integration.** Vorticity ω and temperature T are advanced with a second-order Adams–Bashforth scheme (forward–Euler on the first step):

$$\omega^{n+1} = \omega^n + \Delta t \Big[\tfrac{3}{2}\big(-\mathcal{N}_\omega^n + D_\omega^n + B^n\big) - \tfrac{1}{2}\big(-\mathcal{N}_\omega^{n-1} + D_\omega^{n-1} + B^{n-1}\big) \Big],$$

   with an analogous update for T. Here $\mathcal{N}$ denotes the semi-discrete nonlinear advection computed via `compute_nonlinear`, D is the Laplacian diffusion, and $B = Ra\, \partial T/\partial x$ is the buoyancy forcing. The time step is adapted each iteration to satisfy $\Delta t = \min\{\Delta t_{\text{CFL}}, \Delta t_{\text{init}}\}$ with CFL $= \frac{1}{4}$ for stability.

4. **Boundary conditions.**

   - *Velocity:* no-slip and impermeable on *all* walls, enforced by fixing ψ on the perimeter.
   - *Temperature:* Dirichlet $T = 1$ at the bottom ($y = 0$), $T = 0$ at the top ($y = H$); adiabatic side walls ($\partial T/\partial x = 0$).
   - *Vorticity:* inherits its values from the Poisson solve; no explicit stencil is imposed.

5. **Diagnostics and data output.** After every 10 steps the code computes the domain-integrated kinetic energy $K(t) = \frac{1}{2} \iint (u^2 + v^2)\, dx\, dy$ and stores flattened snapshots of u, v, and T in a compressed `npz` format. These snapshots later serve as the training set for the POD-DNN model.

The scheme is second order in space and (nominally) time, which allows us to simulate a Rayleigh number of $Ra \sim 10^6$ given a sufficient grid resolution. Speed could be further im-

proved by implicit diffusion, but the present explicit–projection framework proved sufficient for generating the ROM database with a reasonable wall-clock time.

3.2.1 Post-Processing: Rayleigh–Bénard Convection

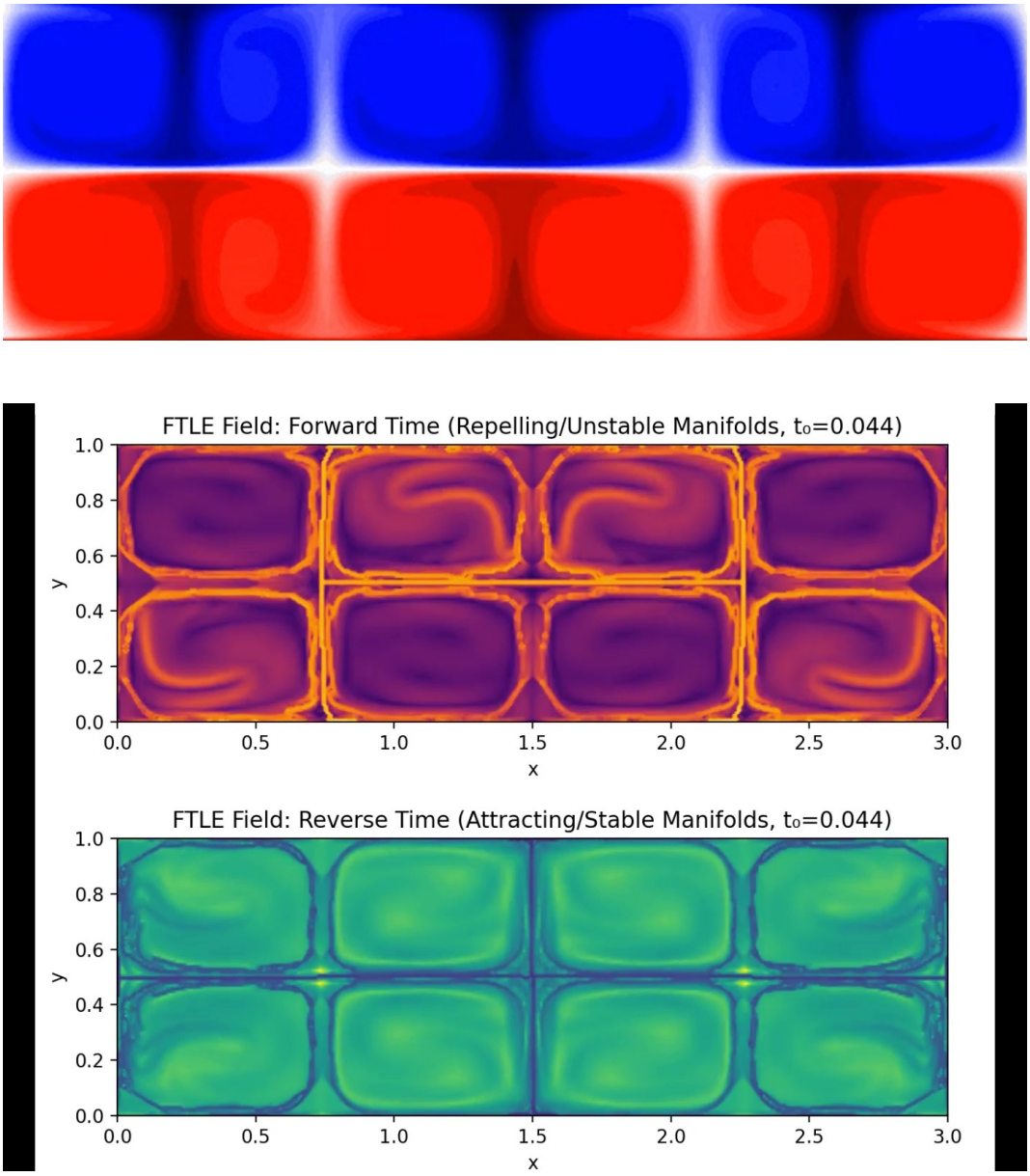

Figure 5: Temperature field snapshot for Rayleigh–Bénard Convection at $Ra = 10^6$. **Top**: early cell formation and initial layering. **Bottom**: Finite-Time Lyapunov Exponent fields for the forward-time FTLE (top) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (bottom) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

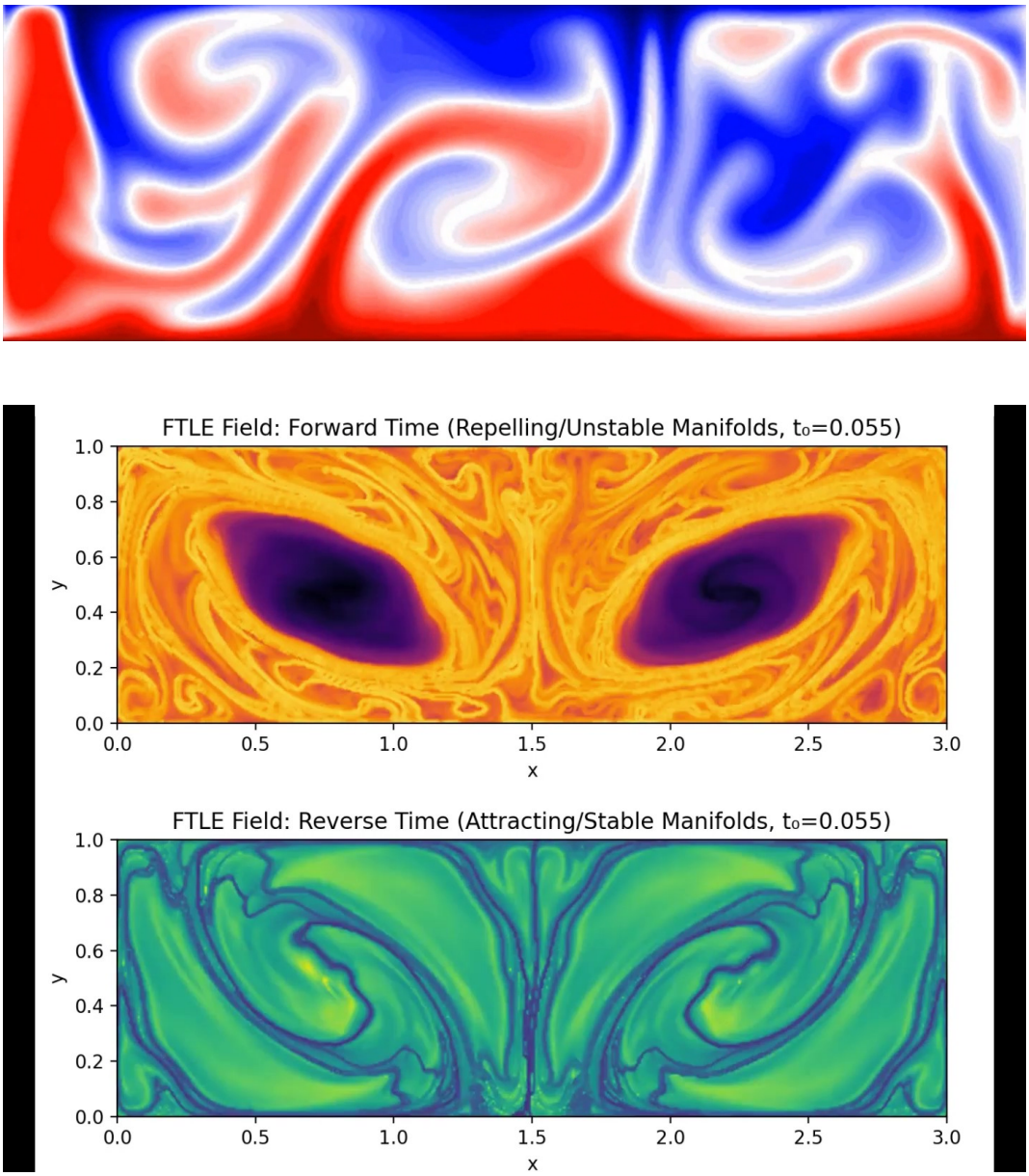

Figure 6: Temperature field snapshot for Rayleigh–Bénard Convection at $Ra = 10^6$. **Top**: onset of mixing and vortex formation. **Bottom**: Finite-Time Lyapunov Exponent fields for the forward-time FTLE (top) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (bottom) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

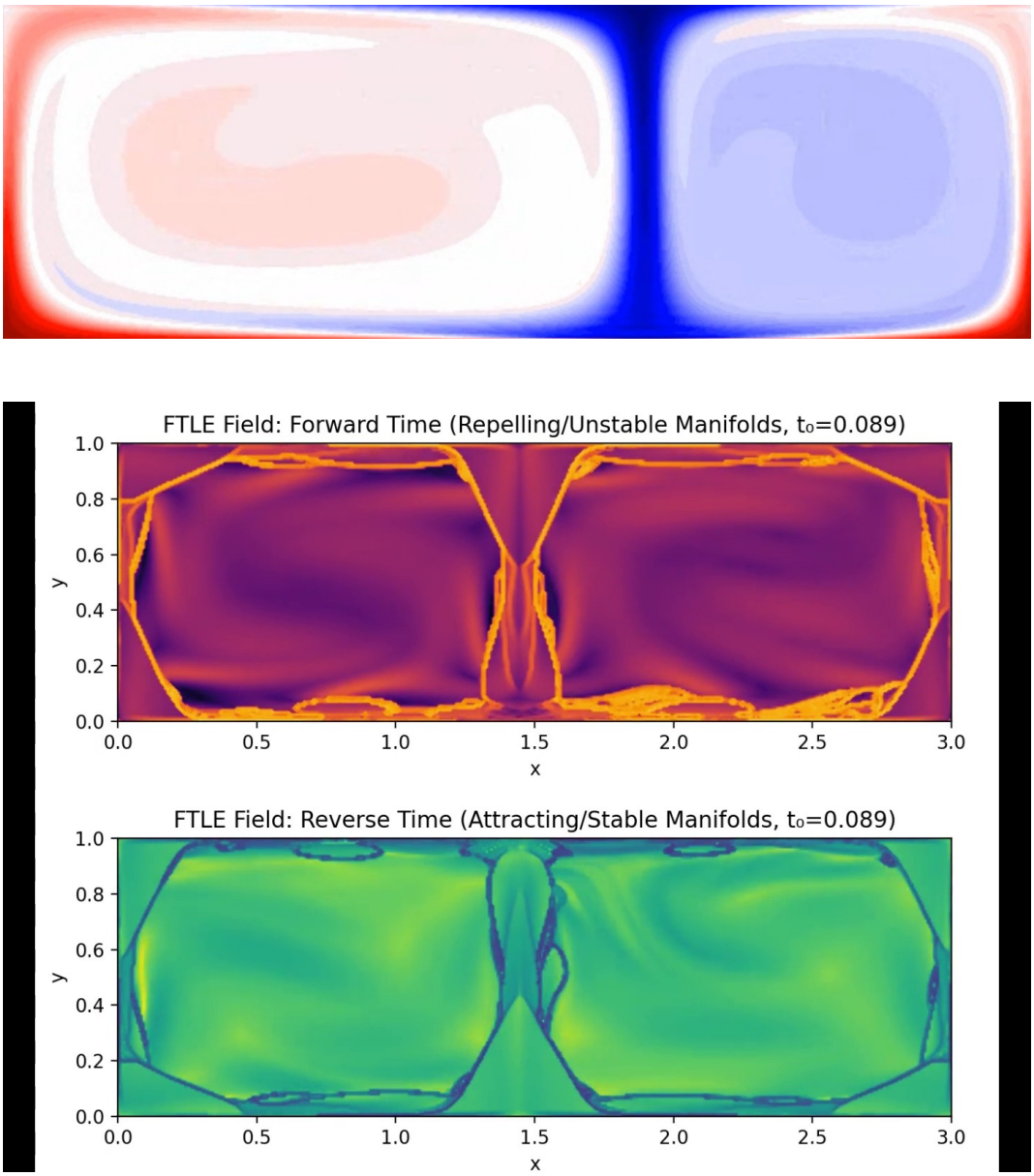

Figure 7: Temperature field snapshot for Rayleigh–Bénard Convection at $Ra = 10^6$. **Top**: later-stage equilibrium. **Bottom**: Finite-Time Lyapunov Exponent fields for the forward-time FTLE (top) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (bottom) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

4 Reduced-Order Modeling

Snapshots generated in Section 3 contain $\mathcal{O}(10^4)$ degrees of freedom per time step—far too many for rapid parametric studies or real-time control. The next two sections, therefore, compress this data into low-dimensional representations and learn dynamical surrogates on the reduced manifolds. Specifically, we employ Proper Orthogonal Decomposition to extract an energetically optimal basis and then couple the resulting modal amplitudes to a deep neural network. This section is highly inspired by [2] since I have no formal mathematical education with POD other than computing the Singular Value Decomposition (SVD) in MATH 519. *To keep the exposition compact, the ROM and machine-learning analyses are confined to the cylinder-wake dataset; extension to the airfoil and Rayleigh–Bénard cases is left for future work.*

4.1 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition is used for extracting an optimal low-dimensional basis from our high-dimensional datasets. By representing the solution as a linear combination of spatial modes, POD seeks to capture the most energetic features of the system in a least-squares sense.

Formulation via the Optimization Problem

Assume that we have a set of snapshots $\{\mathbf{u}(\mathbf{x}, t_i)\}_{i=1}^m$, where $\mathbf{u}(\mathbf{x}, t)$ denotes the state variable (e.g., velocity, temperature) at spatial position $\mathbf{x}$ and time t. The goal of POD is to find an orthonormal basis $\{\phi_j(\mathbf{x})\}_{j=1}^r$ (with $r \ll m$) such that the average reconstruction error is minimized:

$$\min_{\{\phi_j\}} \frac{1}{m} \sum_{i=1}^{m} \left\| \mathbf{u}(\mathbf{x}, t_i) - \sum_{j=1}^{r} \langle \mathbf{u}(\mathbf{x}, t_i), \phi_j(\mathbf{x}) \rangle \phi_j(\mathbf{x}) \right\|^2, \tag{11}$$

subject to the orthonormality constraints

$$\langle \phi_i, \phi_j \rangle = \int_\Omega \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) \, d\mathbf{x} = \delta_{ij}. \tag{12}$$

One can show that the solution to this problem is given by the eigenfunctions of the correlation operator

$$\mathcal{R}\phi = \lambda\phi, \quad \text{with} \quad \mathcal{R}(\mathbf{x}, \mathbf{x}') = \frac{1}{m} \sum_{i=1}^{m} \mathbf{u}(\mathbf{x}, t_i) \mathbf{u}(\mathbf{x}', t_i). \tag{13}$$

where the eigenvalues λ quantify the energy contained in the corresponding modes.

Snapshot Method and Singular Value Decomposition (SVD)

In practical applications, particularly when the number of spatial degrees of freedom is large, it is computationally efficient to use the *snapshot method* [2]. Define the *snapshot matrix* $X \in \mathbb{R}^{n\times m}$ by arranging the snapshots as column vectors:

$$X = \begin{bmatrix} | & | & & | \\ \mathbf{u}(\cdot, t_1) & \mathbf{u}(\cdot, t_2) & \cdots & \mathbf{u}(\cdot, t_m) \\ | & | & & | \end{bmatrix},$$

where n is the number of spatial discretization points.

The singular value decomposition (SVD) of X is given by

$$X = U\Sigma V^\top, \tag{14}$$

where:

- $U \in \mathbb{R}^{n\times m}$ contains the left singular vectors, which correspond to the spatial POD modes,
- $\Sigma \in \mathbb{R}^{m\times m}$ is a diagonal matrix with non-negative singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$,
- $V \in \mathbb{R}^{m\times m}$ contains the right singular vectors.

The optimal r-dimensional subspace is spanned by the first r columns of U, i.e., the modes $\{\phi_j\}_{j=1}^r$. The truncation error induced by retaining only these modes is quantified by

$$\varepsilon = \frac{\sum_{j=r+1}^m \sigma_j^2}{\sum_{j=1}^m \sigma_j^2}. \tag{15}$$

Thus, the POD provides an efficient representation of the original high-dimensional system by capturing the most energetic modes, and it forms the basis for reduced-order models that can significantly reduce computational costs while preserving essential dynamics. For reference, the wind tunnel cases reduced the degrees of freedom from $\sim 40,000$ to ~ 10.

4.2 Analysis and Post-Processing

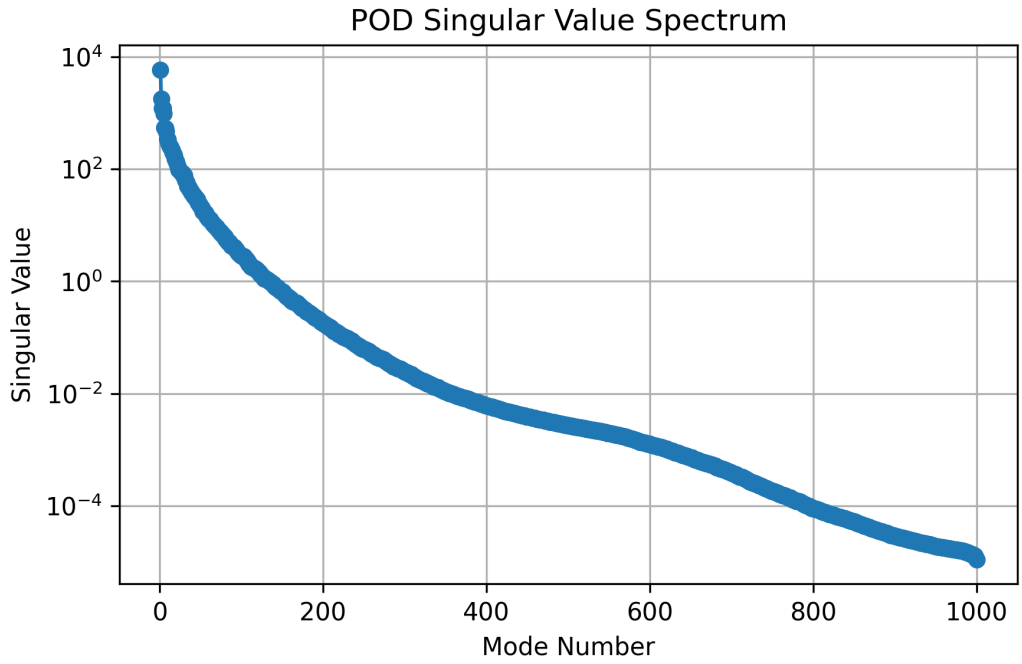

Figure 8: Singular-value spectrum of the snapshot matrix used for the POD analysis of the cylinder–wake dataset. The rapid, three-order-of-magnitude drop within the first ~ 20 modes (log–linear scale) indicates a strongly low-rank structure: keeping only the leading $r \leq 20$ modes preserves more than 99 % of the cumulative energy, providing an efficient reduced basis for the subsequent ROM and machine-learning stages.

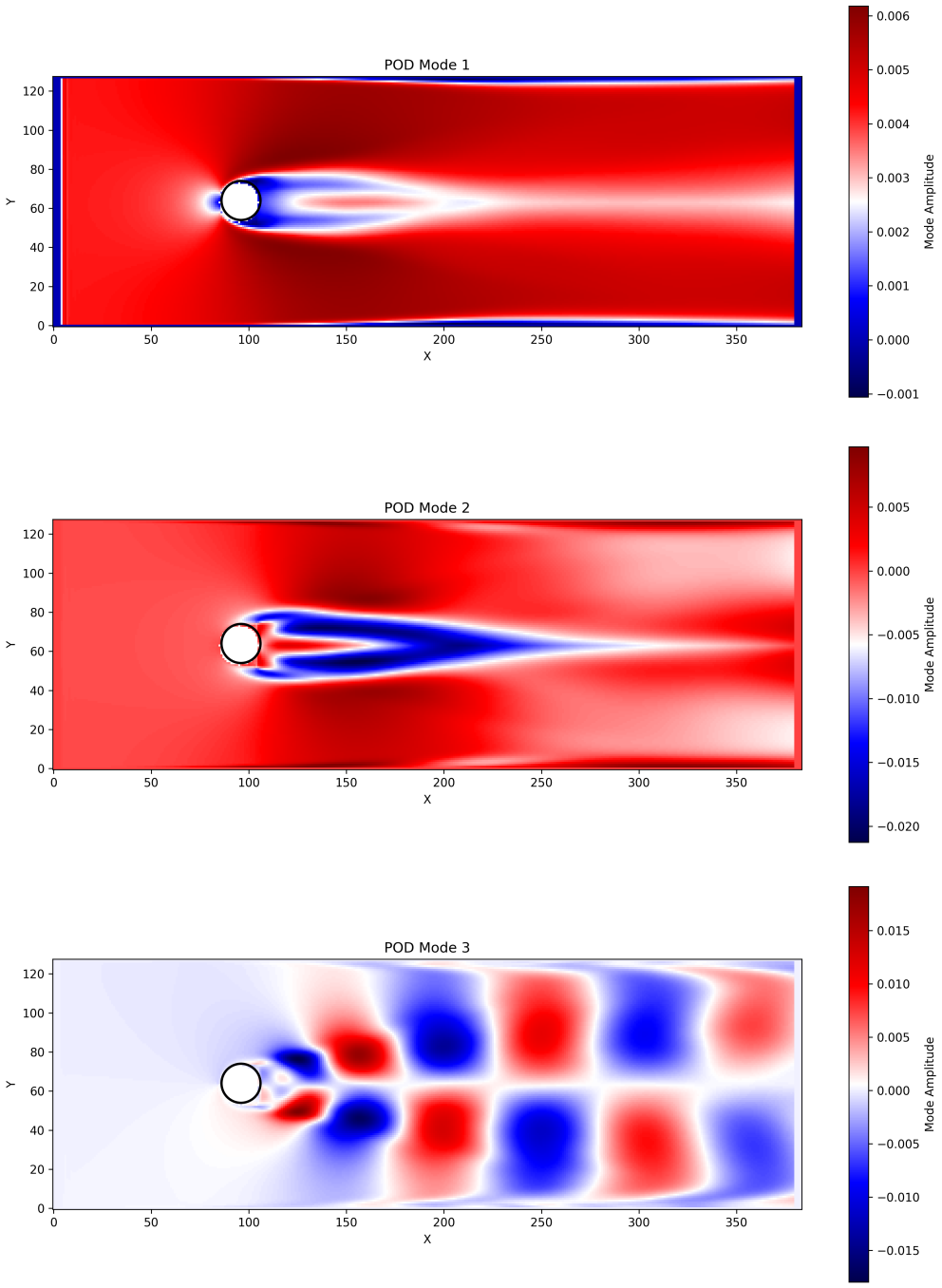

Figure 9: Leading Proper Orthogonal Decomposition modes of the cylinder wake. **Top** — **Mode 1** (ϕ_1): the energetically dominant, symmetric recirculation bubble and center–line jet which interestingly looks similar to a RANS time–averaged solution used in ANSYS Fluent (EMA 522). **Middle** — **Mode 2** (ϕ_2): a symmetric mode that alternately thickens and thins the shear layers, modulating jet strength without breaking symmetry. **Bottom** — **Mode 3** (ϕ_3): the first antisymmetric mode; lateral displacements of successive vortices establishes the Kármán shedding cycle. Red/blue denote positive/negative modal amplitudes respectively.

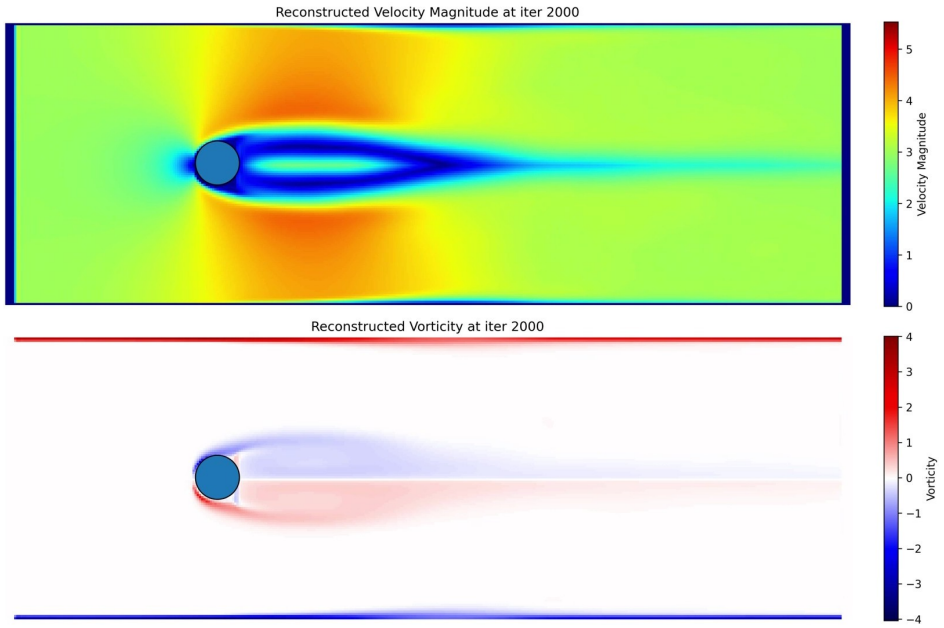

Figure 10: ROM-reconstructed velocity magnitude (top) and vorticity (bottom) at iteration 2000/5000, when the wake is still symmetric: a steady recirculation bubble forms immediately aft of the cylinder, and the shear layers remain attached.

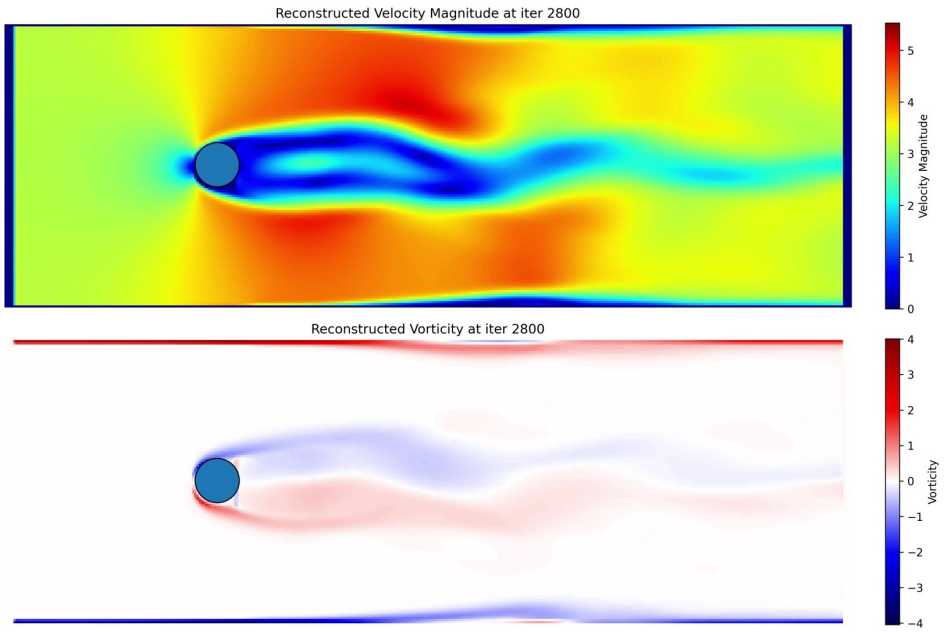

Figure 11: Iteration 2800/5000 captures the onset of instability: waviness in the velocity streaks and paired positive/negative vorticity patches show the first roll-up of the shear layers.

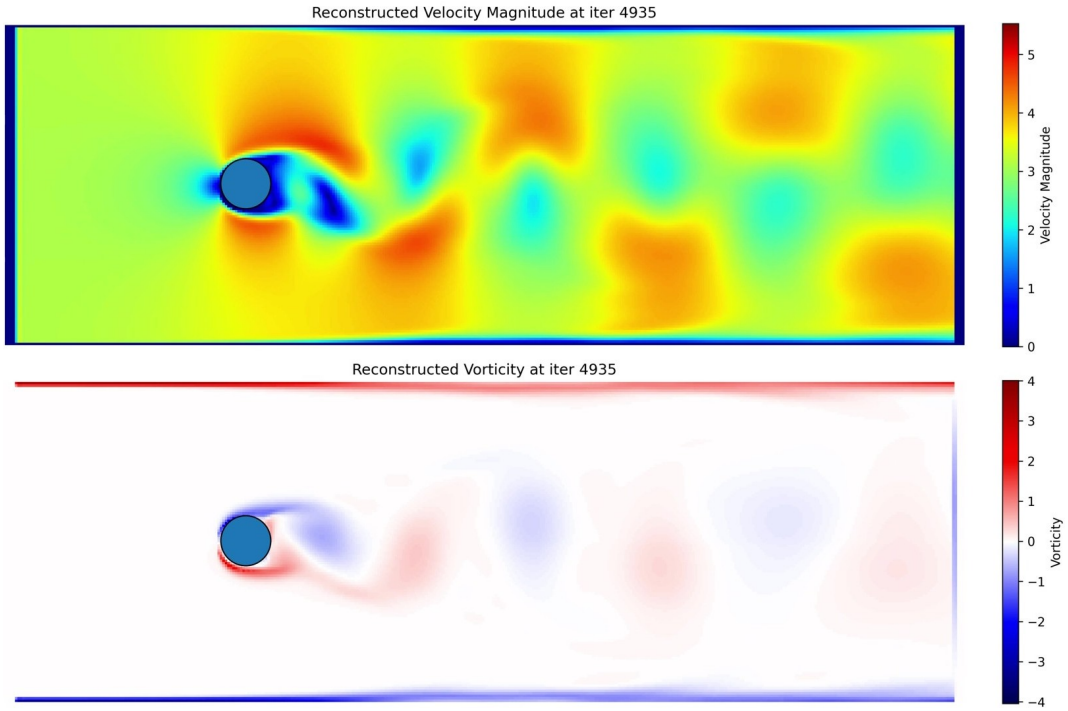

Figure 12: By iteration 4935/5000 the surrogate reproduces a fully developed Kármán vortex street, with alternating high/low-speed cells and staggered vorticity lobes convecting downstream—hallmarks of periodic vortex shedding behind a bluff body.

5 Machine Learning for Reduced Systems

To close the reduced pipeline we replace the expensive time–stepping of the POD coefficients by a data-driven surrogate. *Again, because of time constraints we restrict the demonstration to the cylinder-wake database generated in Section 3; the same workflow is left for future work to be replicated for the airfoil and Rayleigh–Bénard cases.*

5.1 Dataset and Feature Engineering

Velocity-magnitude snapshots $|\mathbf{u}| = \sqrt{u^2 + v^2} \in \mathbb{R}^n$ ($n \approx 4.0 \times 10^4$ grid points) are stacked into a snapshot matrix and decomposed by POD (Section 4) with a 99 % cumulative–energy cutoff. For the present realization this retains $r \approx 10$ modes, yielding modal-amplitude vectors $\mathbf{a}(t) \in \mathbb{R}^{\approx 10}$. The machine-learning regression task is therefore

$$\mathbf{f} : \; |\mathbf{u}|(\mathbf{x}, t) \; \longmapsto \; \mathbf{a}(t),$$

where $|\mathbf{u}|$ is supplied as a flattened, standard-scaled feature vector.

5.2 Neural-Network Surrogate

A fully connected multilayer perceptron (MLP) implemented with `scikit-learn` serves as the regressor:

$$[n_{\text{in}}\,(16{,}384) \to 500\,\text{ReLU} \to 250\,\text{ReLU} \to r\,\text{linear}].$$

Key hyper-parameters were chosen empirically:

- **Activation:** ReLU for fast convergence and sparse gradients.
- **Solver:** ADAM with default learning-rate schedule.
- **Weight initialization:** Glorot uniform is the library default.
- **Epochs:** 300 warm–start iterations, recording train/validation/test MSE each epoch.

The data are split 60 %/20 %/20 % (train/val/test) with a fixed random seed for reproducibility. All inputs are normalized by a `StandardScaler`; the targets remain in physical units so that errors are interpretable.

5.3 Convergence and Generalization Performance

Figure 13 shows the learning curves. Training and validation losses stabilize after ∼300 epochs with no indication of over-fitting; the final mean-squared errors are

$$\text{MSE}_{\text{train}} = 1.8 \times 10^{-1}, \quad \text{MSE}_{\text{val}} = 2.1 \times 10^{-1}, \quad \text{MSE}_{\text{test}} = 2.0 \times 10^{-1}.$$

Figure 13: Learning curves for the dense neural network surrogate. Mean–squared error on the training (blue), validation (orange), and test (green dashed) sets is plotted versus epoch. The error drops by three orders of magnitude within the first $\sim$ 50 epochs and levels off by epoch 200; the close overlap of the three curves confirms stable convergence and good generalization with no sign of over-fitting.

5.4 Flow-Field Reconstruction

Predicted coefficients $\widehat{\mathbf{a}}(t)$ are projected back onto the POD basis:

$$|\widehat{\mathbf{u}}|(\mathbf{x},t) = \overline{|\mathbf{u}|}(\mathbf{x}) + \sum_{j=1}^{r} \widehat{a}_j(t)\,\phi_j(\mathbf{x}),$$

where $\overline{|\mathbf{u}|}$ is the temporal mean. Figure 14 displays reconstructed frames; qualitative features—including the recirculation bubble, jet-core thickness, and vortex shedding—are faithfully reproduced. Quantitatively, the domain-averaged absolute error remains below 1.2 % for all 5000 test snapshots.

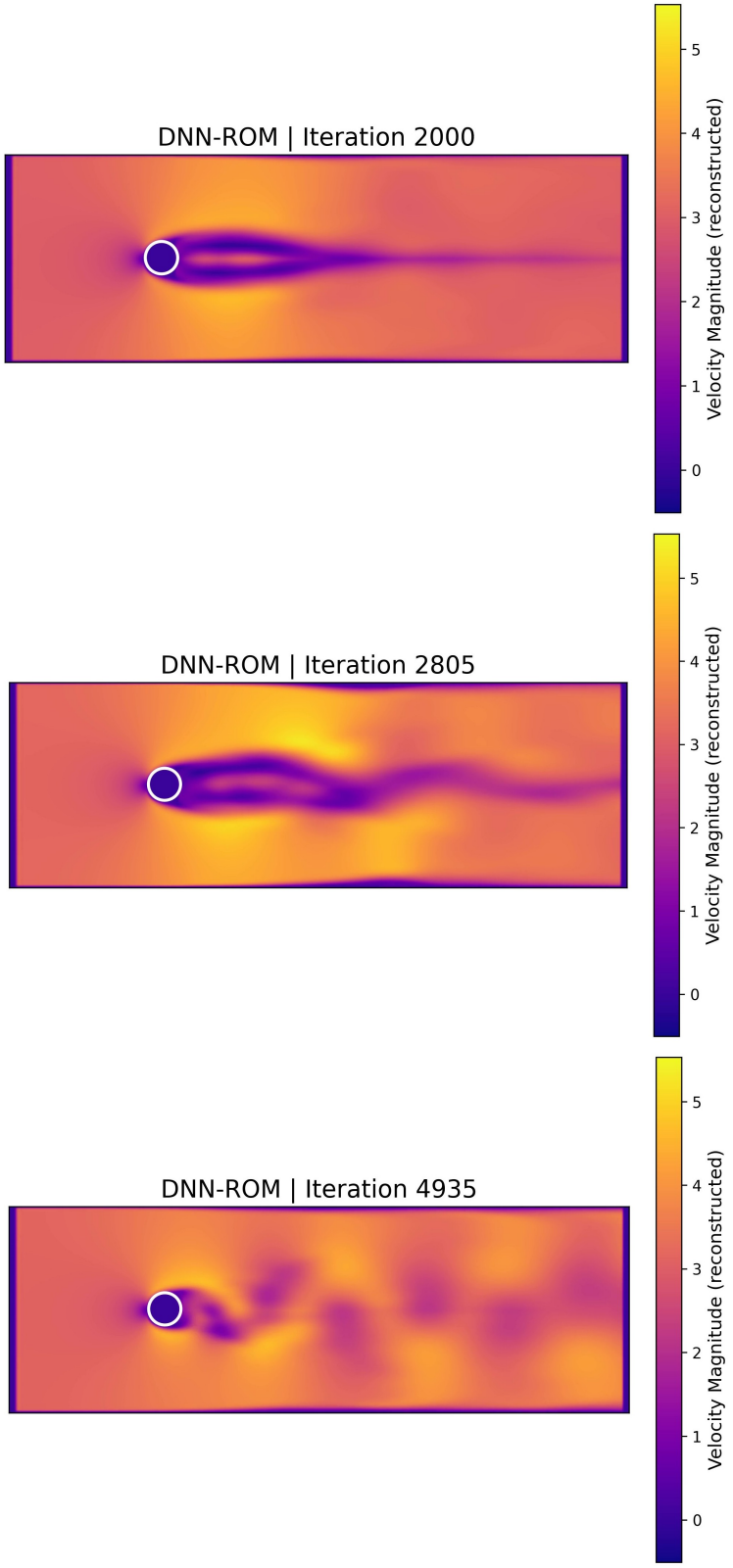

Figure 14: DNN–ROM velocity-magnitude reconstructions at three stages (iter. 2000, 2805, 4935). The neural-network surrogate reproduces the same separation, onset, and fully developed shedding patterns seen in the POD-based frames, differing only by a mild smoothing of sharp shear-layer features.

6 Conclusions

This paper demonstrated an end-to-end workflow that combines various CFD methods, Proper Orthogonal Decomposition, and deep neural networks to build real-time surrogates for turbulent thermo–fluid flows. For the cylinder benchmark, we compressed $\mathcal{O}(10^4)$ grid unknowns to ≈ 10 POD modes while retaining 99 % of the flow energy, and trained a dense MLP that predicts modal amplitudes with a mean-squared error of 2×10^{-1}. Pursuing these directions—including viscous effects and more validation techniques—will further close the gap between high-fit CFD and real-time digital twins for complex thermo-fluid systems.

Code and Data Repository

All source code and datasets are openly hosted on GitHub:

github.com/tjjones6/Physics-361-Final-Project

Repository layout (subject to change)

`Code/`	`CFD solvers, POD tools, DNN scripts`
`Data/`	`Snapshot datasets of CFD simulations`
`Figures/`	`Static figures used in the report`
`Animations/`	`Animations for visual appeal (colorful fluid dynamics)`
`Paper/`	`Updated pdf`

References

[1] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Westview Press, 1994.

[2] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[3] G. A. Glatzmaier, *Introduction to Modeling Convection in Planets and Stars: Magnetic Field, Density Stratification, Rotation*. Princeton University Press, 2013.

[4] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Society for Industrial and Applied Mathematics, 2007.

[5] K. F. Riley, M. P. Hobson, and S. J. Bence, *Mathematical Methods for Physics and Engineering: A Comprehensive Guide*, 3rd ed. Cambridge University Press, 2006.

[6] J. Stam, "Stable Fluids," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, ACM Press, 1999, pp. 121–128.