Trustworthy Operator Learning for Incompressible Flows

Ramansh Sharma[1], Matthew Lowery[1], Houman Owhadi[2], Varun Shankar[1]

[1]Kahlert School of Computing, University of Utah,
[2]Department of Computing and Mathematical Sciences, California Institute of Technology

2025 SIAM Conference on Analysis of Partial Differential Equations
November 18, 2025

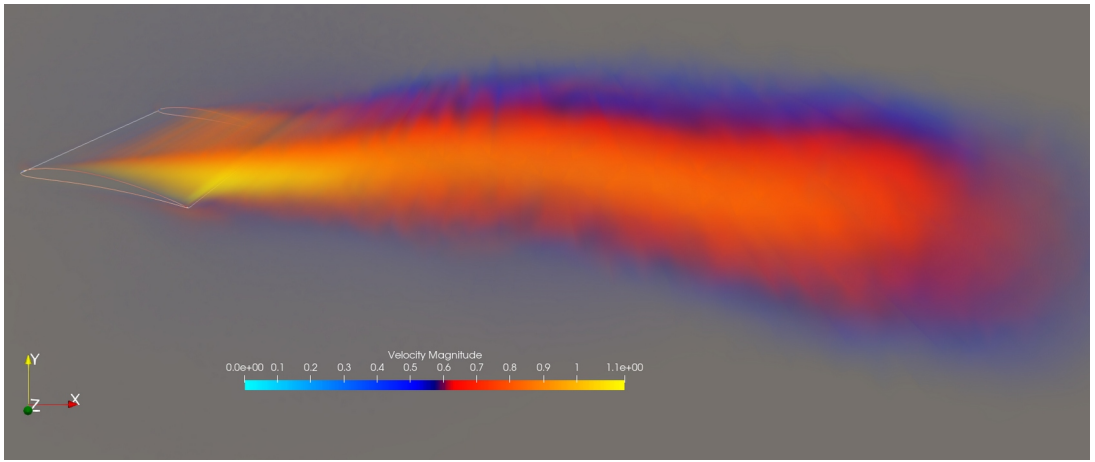
THE UNIVERSITY of UTAH

Outline

1. Motivation – Incompressible flow
2. Surrogate modeling
3. Operator Learning
4. A Property Preserving Kernel Method
5. Results
6. Conclusion & Future work

Motivation

- Incompressible fluid flow is of huge interest which arise in a range of applications.
  - Aerodynamics: Flow past airfoils, F1 cars, etc.
  - Hydrodynamics: flow around ships/submarines, circulation of water in ocean basins.
  - Medicine: blood flow in heart valves, arteries, etc.



Flow past a 3D airfoil

Motivation

- A large corpus of numerical solvers for incompressible fluid flow exist. These methods achieve state-of-the-art accuracy!
- But they are often costly and nontrivial to implement.
- **Solution**: surrogate modeling!

Surrogate modeling

Machine learning based surrogate models have recently become popular.

- Deep Operator Networks (DeepONets)[1]
- Fourier Neural Operators (FNOs)[2]

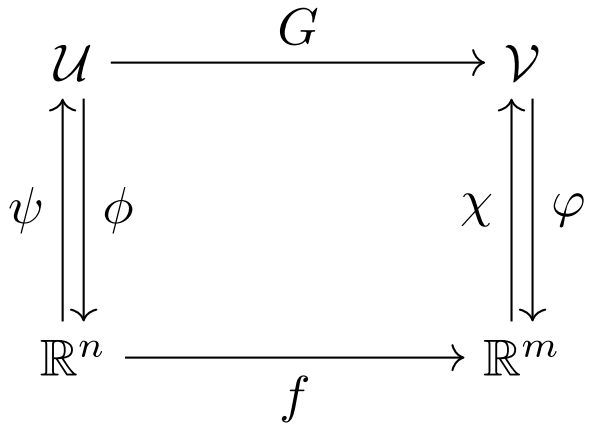
1. Lu, L., Jin, P., & Karniadakis, G. E. (2019). Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators.

2. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations.

Operator Learning

- Function approximation is between vector spaces - $\mathbb{R}^n \xrightarrow{f} \mathbb{R}^m$
- Operators are maps between function spaces - $\mathcal{U} \xrightarrow{G} \mathcal{V}$
- Example operators
  - Derivatives
  - Integrals
  - Green's functions
  - **PDE solution operators!**

Operator Learning



- Discretize the function spaces: $(u_i, v_i)_{i=1}^N,\ u_i \in \mathcal{U},\ v_i \in \mathcal{V}$
- Discretize the functions on their respective domains: $u(X) \in \mathbb{R}^n,\ v(Y) \in \mathbb{R}^m$
- The vector map f needs to be learned/approximated.
- As a learning problem, approximate G by minimizing:

$$\|\tilde{G}(u_i(\boldsymbol{x}))(\boldsymbol{y})\big|_{X,Y} - v_i(\boldsymbol{y})\big|_Y\|_2^2.$$

Operator Learning

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nu \nabla^2 \mathbf{u} - \frac{1}{\rho}\nabla p + \frac{1}{\rho}\mathbf{f}, \quad \text{on } \Omega \times (0, T],$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{on } \Omega \times (0, T],$$

$$\mathcal{B}\mathbf{u} = \mathbf{g}, \quad \text{on } \partial\Omega \times (0, T],$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \text{on } \Omega.$$

Incompressible Navier-Stokes PDE

Operator Learning

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nu \nabla^2 \mathbf{u} - \frac{1}{\rho}\nabla p + \frac{1}{\rho}\mathbf{f}, \quad \text{on } \Omega \times (0, T],$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{on } \Omega \times (0, T],$$

$$\mathcal{B}\mathbf{u} = \mathbf{g}, \quad \text{on } \partial\Omega \times (0, T],$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \text{on } \Omega.$$

Incompressible Navier-Stokes PDE

Useful operators:

- $G : \mathbf{u}(\mathbf{x}, t = 0) \longrightarrow \mathbf{u}(\mathbf{x}, t = \mathrm{T})$
- $G : \mathbf{u}(\mathbf{x}, t = 0) \longrightarrow \mathbf{u}(\mathbf{x}, t \in [\mathrm{T}_1, \mathrm{T}_2])$
- $G : \mathbf{f} \longrightarrow \mathbf{u}$

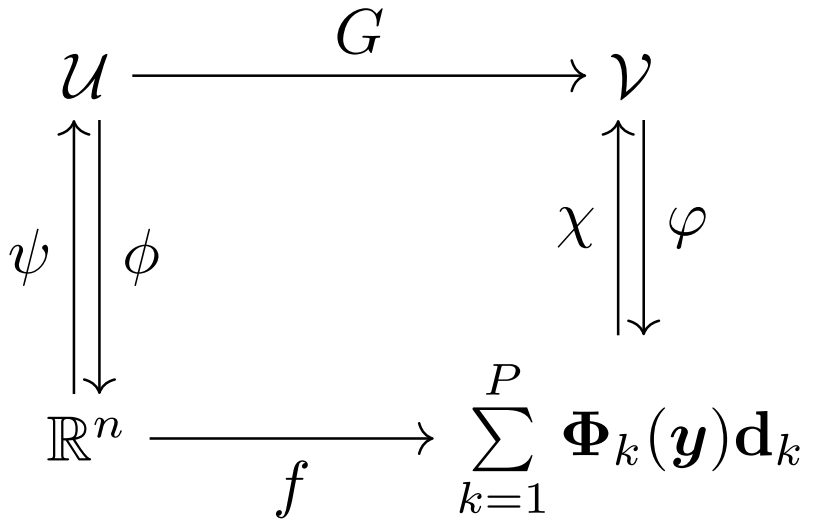
Operator Learning

- We are interested in operator learning architectures for incompressible flows.
- There has been some work in using neural networks for conservation laws[2], and **physics-guided FNOs** in solid mechanics to model divergence-free fields[1], but challenges remain.
- A more interpretable way to bake in incompressibility (and other properties) is **kernels**!

1. Khorrami, M. S., Goyal, P., Mianroodi, J. R., Svendsen, B., Benner, P., & Raabe, D. (2024). A physics-encoded Fourier neural operator approach for surrogate modeling of divergence-free stress fields in solids.

2. Richter-Powell, J., Lipman, Y., & Chen, R. T. (2022). Neural conservation laws: A divergence-free perspective. *Advances in Neural Information Processing Systems*, 35, 38075-38088.

A Property Preserving Kernel Method



- We want physical constraints to be preserved in the predicted output functions.
- A natural way is to express output functions with a property-preserving kernel basis.
- The operator learning problem then maps from input functions to the expansion coefficients $\mathbf{d}$ in that basis.
- Why use a kernel basis? It's easy to *analytically* bake in desirable properties!

A Property Preserving Kernel Method – Incompressibility

$$\phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R},$$

$$\boldsymbol{\Phi}^{df}(\boldsymbol{x}, \boldsymbol{y}) = \nabla_{\boldsymbol{x}} \times \nabla_{\boldsymbol{y}} \times \phi(\boldsymbol{x}, \boldsymbol{y}), \quad \boldsymbol{\Phi}^{df} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d \times d},$$

$$s(\boldsymbol{x}) = \sum_{k=1}^{P} \boldsymbol{\Phi}^{df}(\boldsymbol{x}, \boldsymbol{x}_k)\, \mathbf{d}_k.$$

- We construct an *analytically* divergence-free *matrix-valued* kernel (whose columns are divergence-free) by applying differential operators on a scalar-valued kernel.

Fuselier, E. J., Shankar, V., & Wright, G. B. (2016). A high-order radial basis function (RBF) Leray projection method for the solution of the incompressible unsteady Stokes equations. *Computers & Fluids*, 128, 41-52.

Fuselier, E. J., & Wright, G. B. (2017). A radial basis function method for computing Helmholtz–Hodge decompositions. *IMA Journal of Numerical Analysis*, 37(2), 774-797.

A Property Preserving Kernel Method – Periodicity

$$h(\boldsymbol{x}) = (\cos(x_1), \sin(x_1), \ldots, \cos(x_d), \sin(x_d)),$$
$$\phi^{\pi}(\boldsymbol{x}, \boldsymbol{y}) = \phi(h(\boldsymbol{x}), h(\boldsymbol{y})),$$
$$s(\boldsymbol{x}) = \sum_{k=1}^{P} \boldsymbol{\Phi}^{df,\pi}(\boldsymbol{x}, \boldsymbol{x}_k)\ \mathbf{d}_k.$$

- To make the kernel spatially periodic, we compute the basis functions on periodic inputs $h(\boldsymbol{x}), h(\boldsymbol{y})$.

Shankar, V., Wright, G. B., Kirby, R. M., & Fogelson, A. L. (2015). Augmenting the immersed boundary method with Radial Basis Functions (RBFs) for the modeling of platelets in hemodynamic flows. *International Journal for Numerical Methods in Fluids*, 79(10), 536-557.

Owhadi, H. (2023). Gaussian process hydrodynamics. *Applied Mathematics and Mechanics*, 44(7), 1175-1198.

A Property Preserving Kernel Method – Power laws

$$\hat{\mathbf{\Phi}} = \sum_{j=0}^{m} \alpha_j \mathbf{\Phi}_j^{df,\pi},$$

$$s(\boldsymbol{x}) = \sum_{k=1}^{P} \hat{\mathbf{\Phi}}(\boldsymbol{x}, \boldsymbol{x}_k)\, \mathbf{d}_k.$$

- The kernel can be informed with velocity-increments scaling/power laws arising from Kolmogorov-41 laws in the form of additive kernels.

Owhadi, H. (2023). Gaussian process hydrodynamics. *Applied Mathematics and Mechanics*, 44(7), 1175-1198.

A Property Preserving Kernel Method

$$v_i(Y) = s(Y), i = 1, \ldots, N.$$

- Now that we have a kernel basis with desirable properties, we solve for the expansion coefficients $\underline{\mathbf{d}}$ for each output function over some points Y .

A Property Preserving Kernel Method

- How do we map from input functions to the output expansions coefficients?
- Another kernel map!

A Property Preserving Kernel Method

Let $\underline{\boldsymbol{u}} = \begin{bmatrix} u_1(X) \\ u_2(X) \\ \vdots \\ u_N(X) \end{bmatrix}$ and $\underline{\boldsymbol{\delta}} = \begin{bmatrix} \underline{\mathbf{d}}_1 \\ \underline{\mathbf{d}}_2 \\ \vdots \\ \underline{\mathbf{d}}_N \end{bmatrix}$ be block vectors of “stacked” input functions and output expansion coefficients respectively.

Batlle, P., Darcy, M., Hosseini, B., & Owhadi, H. (2024). Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496, 112549.

A Property Preserving Kernel Method

Let $\underline{\boldsymbol{u}} = \begin{bmatrix} u_1(X) \\ u_2(X) \\ \vdots \\ u_N(X) \end{bmatrix}$ and $\underline{\boldsymbol{\delta}} = \begin{bmatrix} \underline{\mathbf{d}}_1 \\ \underline{\mathbf{d}}_2 \\ \vdots \\ \underline{\mathbf{d}}_N \end{bmatrix}$ be block vectors of "stacked" input functions and output expansion coefficients respectively.

We then write the interpolation problem as

$$\underline{\mathbf{d}}_j = \sum_{i=1}^{N} \boldsymbol{k}(u_j, u_i) c_i, \;\; j = 1, \ldots, N,$$

where $\boldsymbol{k}$ is a positive-definite matrix-valued kernel.

The key insight here is treating $u(X)$ as a point in $\mathbb{R}^n$!

Batlle, P., Darcy, M., Hosseini, B., & Owhadi, H. (2024). Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496, 112549.

Results - Setup

- All problems have **10,000 training** and **200 test** input/output functions.
- We report relative l_2 errors averaged over the **test** output functions.
- We experimented using three different kernels for $\boldsymbol{\Phi}$ and $\boldsymbol{k}$:
  - Gaussian
  - Compactly-supported C^4 Wendland
  - C^4 Matérn
- It is crucially important that we pick positive-definite kernels for the operator map!

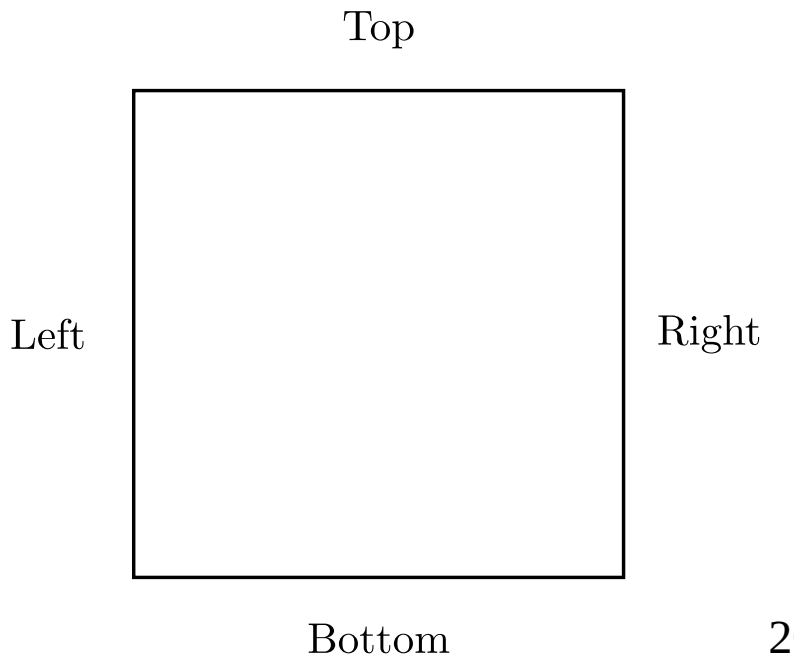
Results – 2D Taylor-Green vortices

- $\Omega = [0, 2\pi]^2$
- Initial condition:

$$u(\mathbf{x}) = A\ \sin(x)\ \cos(y)\ \exp(-2\nu t),$$
$$v(\mathbf{x}) = -A\ \cos(x)\ \sin(y)\ \exp(-2\nu t).$$

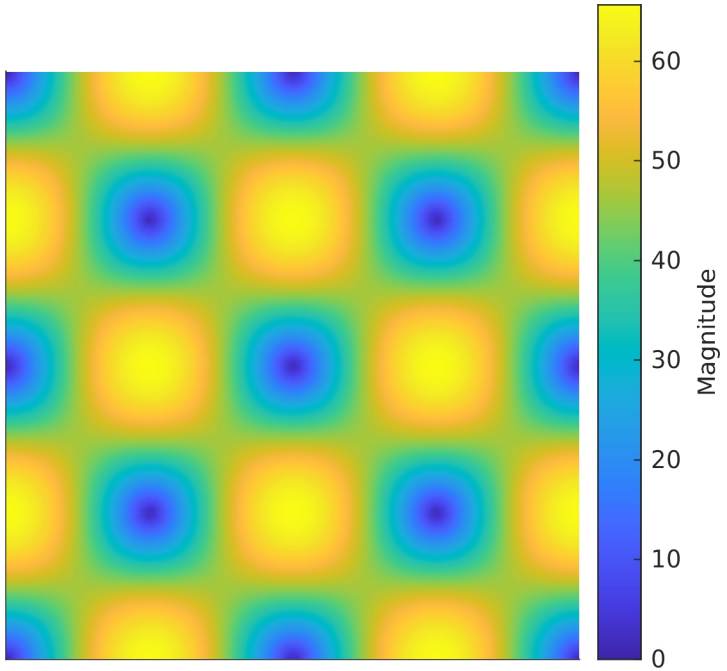
$$G : \mathbf{u}(\mathbf{x}, 0) \longrightarrow \mathbf{u}(\mathbf{x}, 1)$$

- Periodic boundary conditions in both directions.
- Functions vary by randomly sampling:
  - A : 0.1 – 80
  - ν : 0.0001 – 1

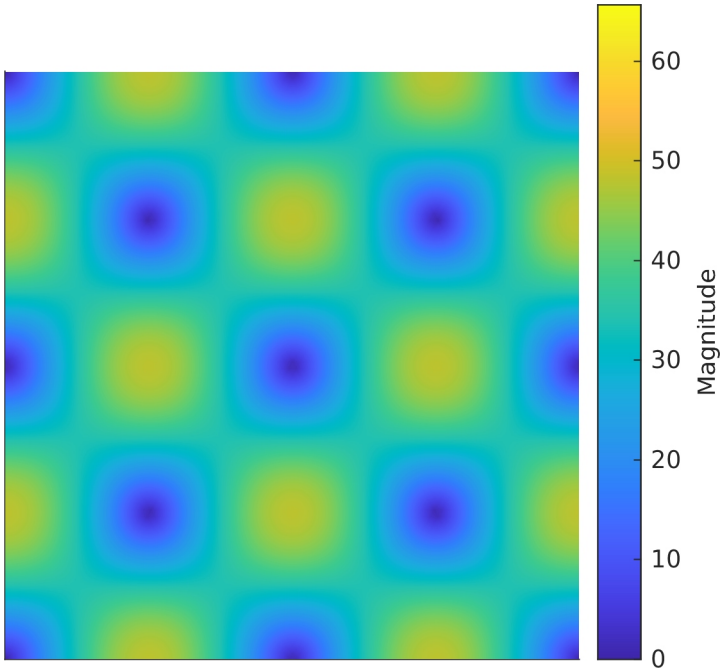


Results – 2D Taylor-Green vortices

- Triangular mesh with 7477 points.
- Example input and output functions.

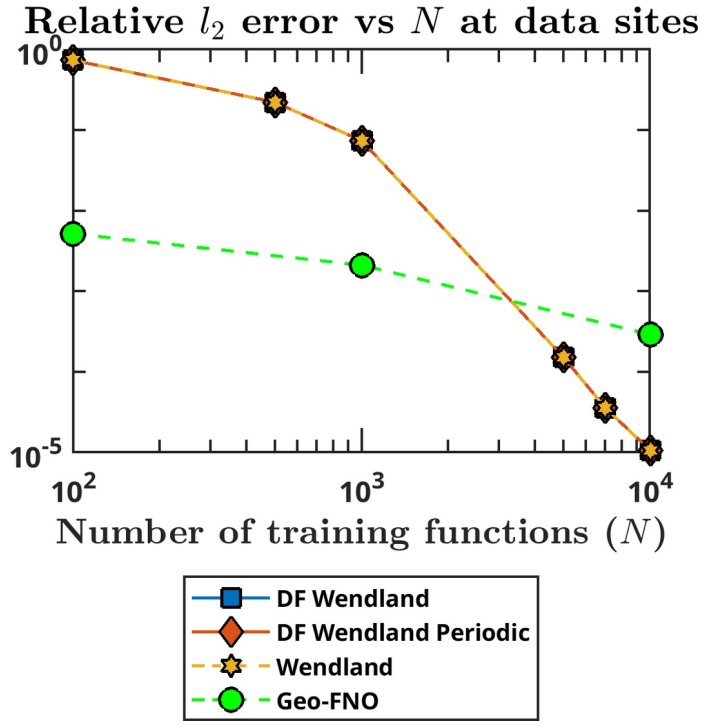


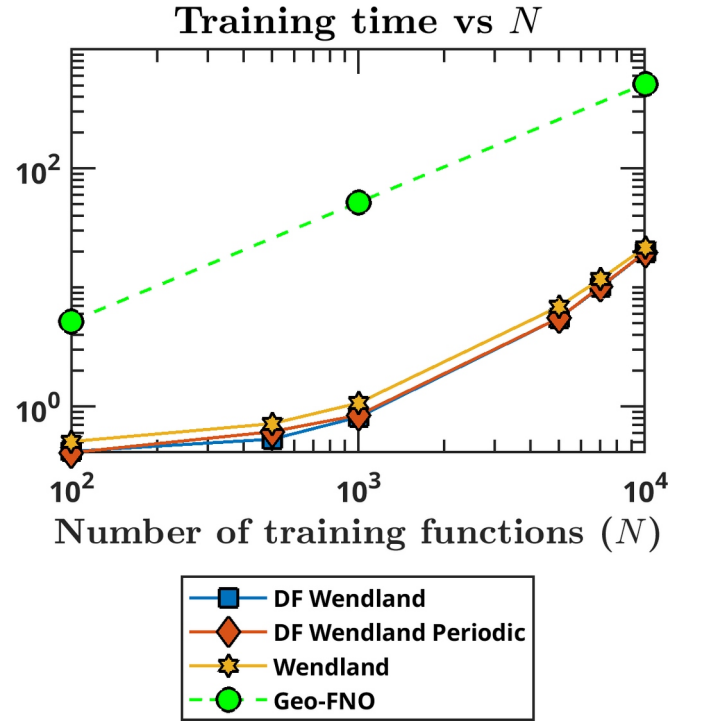
Input function

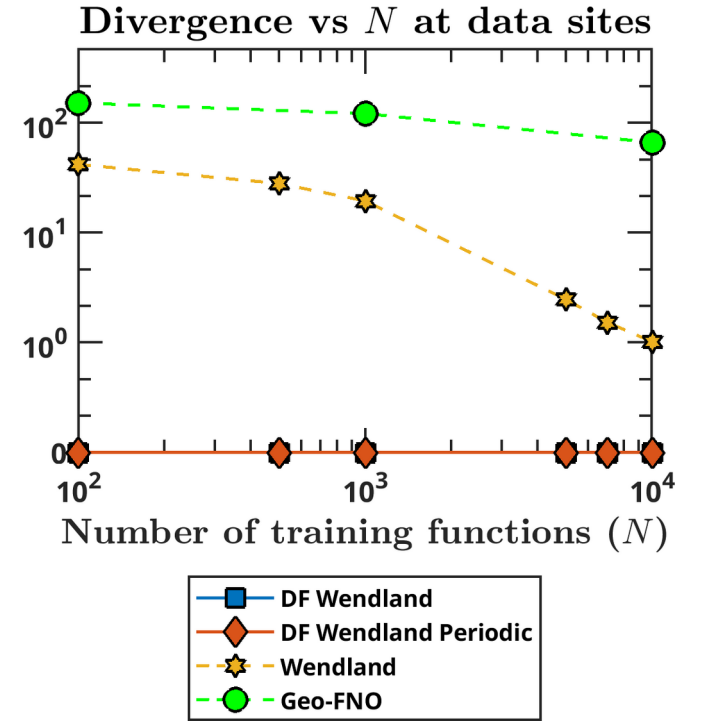


Output function

Results – 2D Taylor-Green vortices







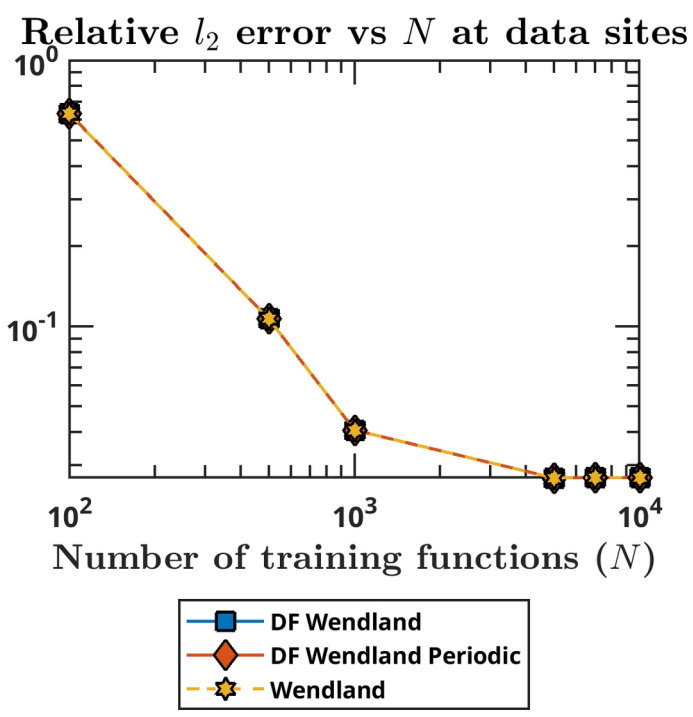
Results – 2D Spacetime Taylor-Green vortices

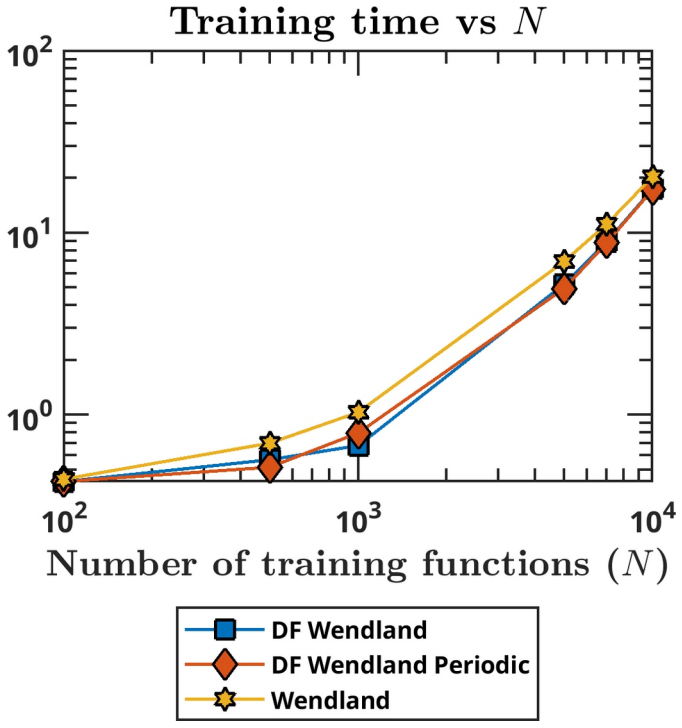
- We map from initial condition to four final timesteps.
- $T = [0.7, 0.8, 0.9, 1]$.
- We employ a spacetime product kernel for the output functions.

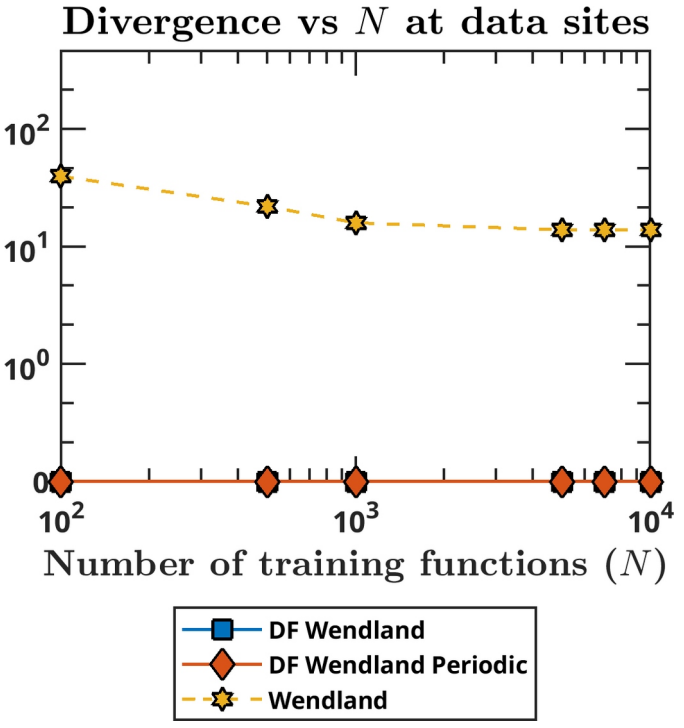
$$G : \mathbf{u}(\mathbf{x}, 0) \longrightarrow \mathbf{u}(\mathbf{x}, T)$$

$$s(\boldsymbol{x}, t) = \sum_{k=1}^{P} \left(\psi(t, t_k) \otimes \mathbf{\Phi}^{df,\pi}(\boldsymbol{x}, \boldsymbol{x}_k) \right) \mathbf{d}_k.$$

Results – 2D Spacetime Taylor-Green vortices







Results – 3D *Turbulent* Species Transport

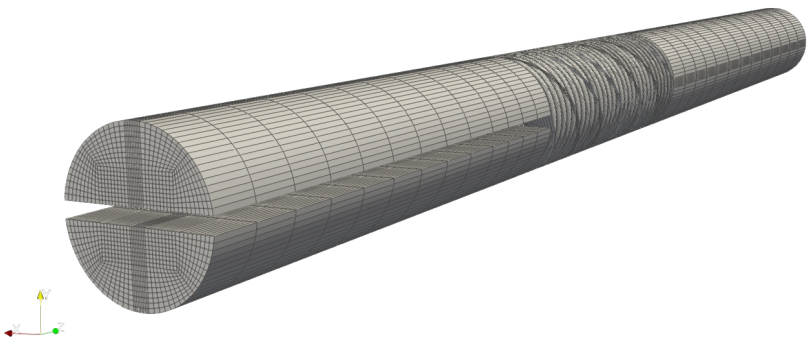
- Reynolds number range: 10^5 – 10^6.
- $\Delta t = 0.05,\ T = 0.5.$
- Initial condition:

$$\mathbf{u}^{\text{air inlet}} = \text{specified},$$

$$\mathbf{u}^{\text{gas inlet}} = \text{specified},$$

$$\mathbf{u}^{\text{everywhere else}} = 0.$$

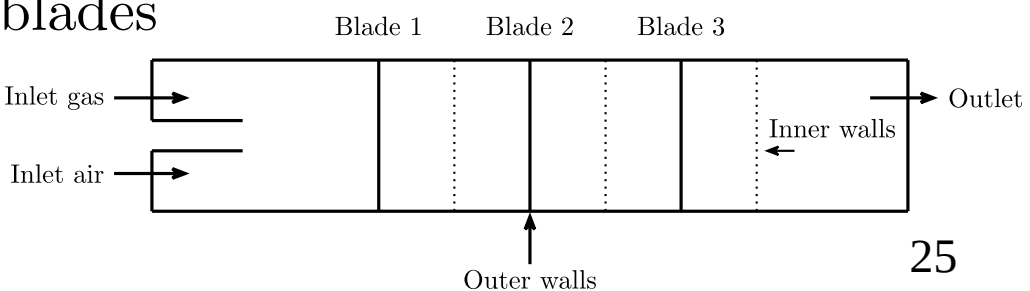
$$G : \mathbf{u}^{\text{air, gas inlet}} \longrightarrow \mathbf{u}(\mathbf{x}, 0.5)$$



- Boundary condition:

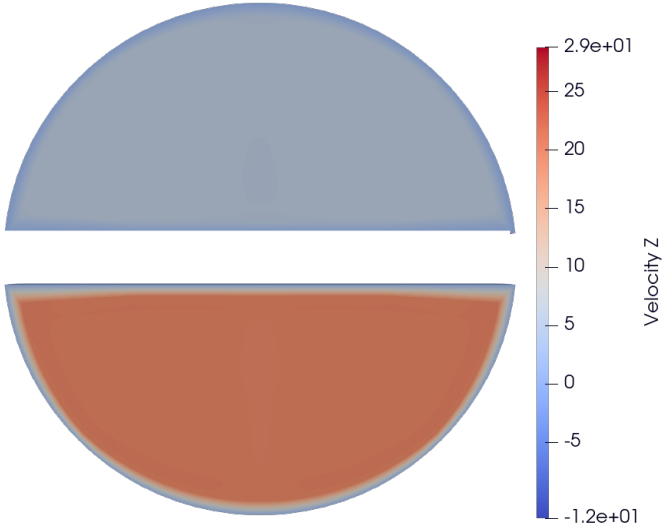
$$\mathbf{u} = 0,\ \text{inner \& outer walls, three blades}$$

$$p^{\text{right}} = 0.$$

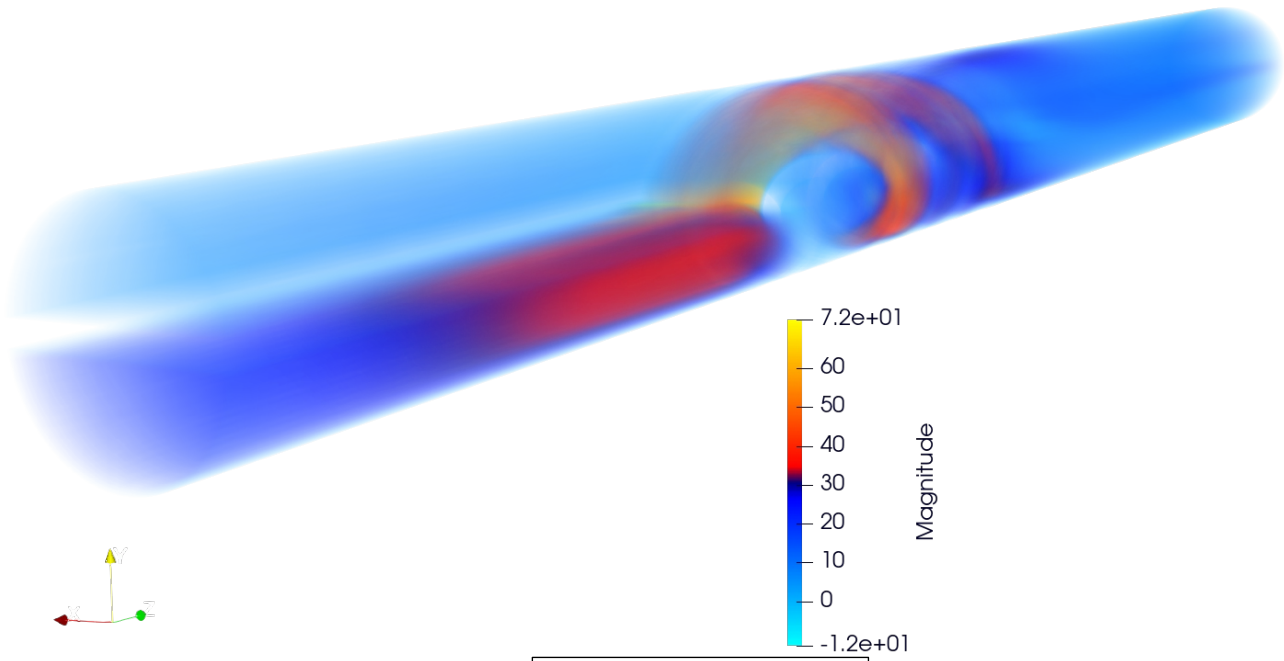


Results – 3D *Turbulent* Species Transport

- Volumetric mesh with 84,876 points.
- Example input and output functions.

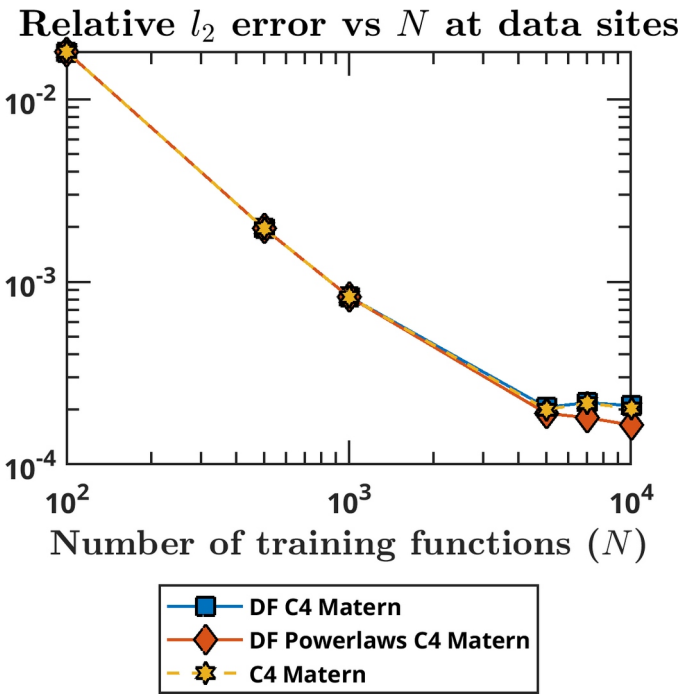


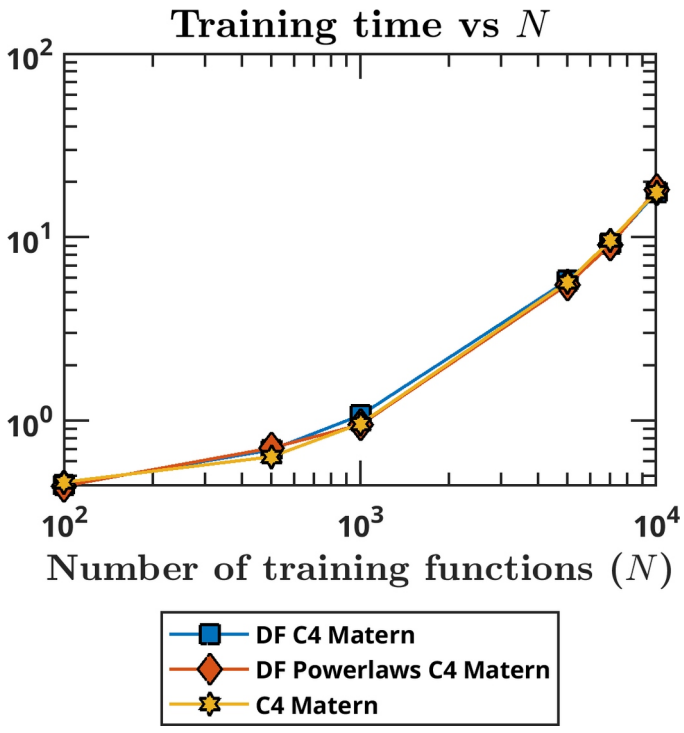
Input function

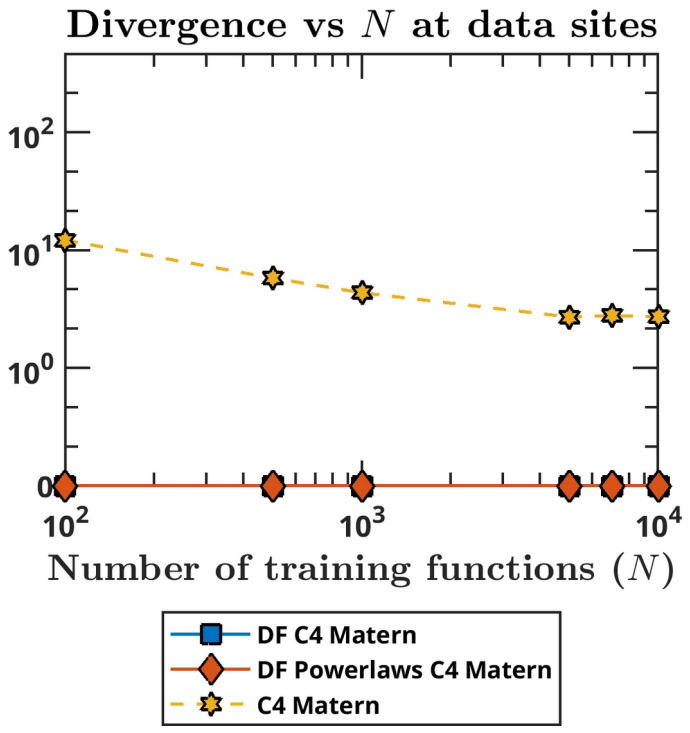


Output function

Results – 3D *Turbulent* Species Transport







Conclusion & Future work

- Property-preserving kernel methods are overall competitive against other operator learning methods for 2D and 3D fluid flow problems.
- One does not have to sacrifice accuracy for preserving physics!
- **Big challenge!** Cost of the linear solve scales cubically with N. Ways to get around this to scale to big datasets!
- Incorporating additional desirable properties in the kernels.
- Applications in magnetohydrodynamics.

Thank you! Questions?

References

- Lu, L., Jin, P., & Karniadakis, G. E. (2019). Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators.
- Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations.
- Khorrami, M. S., Goyal, P., Mianroodi, J. R., Svendsen, B., Benner, P., & Raabe, D. (2024). A physics-encoded Fourier neural operator approach for surrogate modeling of divergence-free stress fields in solids.
- Richter-Powell, J., Lipman, Y., & Chen, R. T. (2022). Neural conservation laws: A divergence-free perspective. *Advances in Neural Information Processing Systems*, 35, 38075-38088.
- Fuselier, E. J., Shankar, V., & Wright, G. B. (2016). A high-order radial basis function (RBF) Leray projection method for the solution of the incompressible unsteady Stokes equations. *Computers & Fluids*, 128, 41-52.
- Fuselier, E. J., & Wright, G. B. (2017). A radial basis function method for computing Helmholtz–Hodge decompositions. *IMA Journal of Numerical Analysis*, 37(2), 774-797.
- Shankar, V., Wright, G. B., Kirby, R. M., & Fogelson, A. L. (2015). Augmenting the immersed boundary method with Radial Basis Functions (RBFs) for the modeling of platelets in hemodynamic flows. *International Journal for Numerical Methods in Fluids*, 79(10), 536-557.
- Owhadi, H. (2023). Gaussian process hydrodynamics. *Applied Mathematics and Mechanics*, 44(7), 1175-1198.
- Batlle, P., Darcy, M., Hosseini, B., & Owhadi, H. (2024). Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496, 112549.

Results – 2D Laminar Flow past a Cylinder

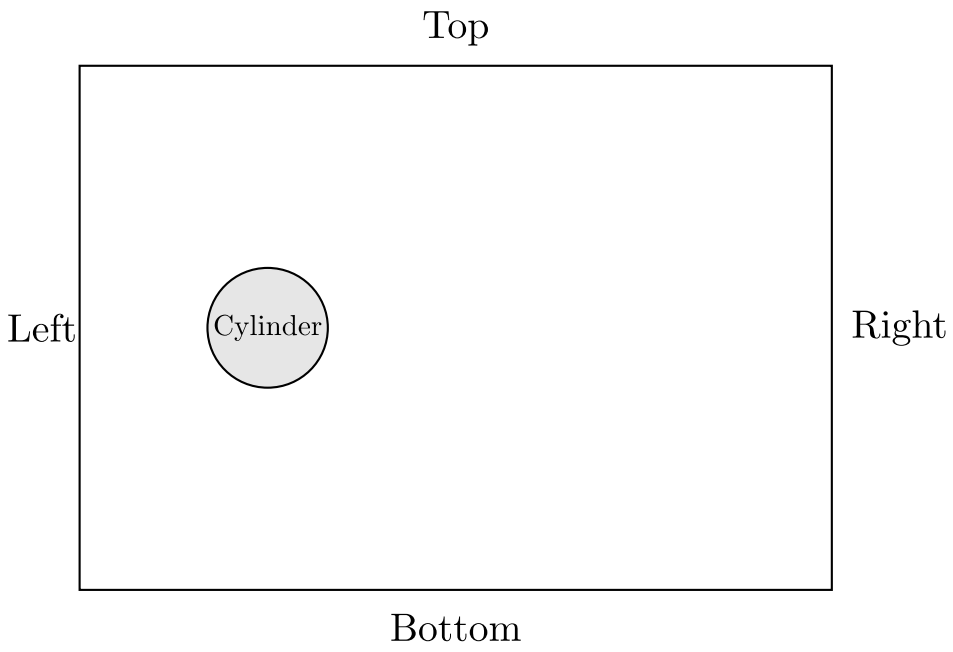
- $\Omega = [0, 20] \times [0, 14]$.
- Reynolds number range: 112 – 199.
- $\Delta t = 0.001,\ T = 10.$
- Initial condition:

$$\mathbf{u}(\mathbf{x}, 0) = \text{specified}.$$

- Boundary condition:

$$\begin{aligned} \mathbf{u}^{\text{left}} &= \text{specified}, \\ \mathbf{u}^{\text{cylinder}} &= 0, \\ v^{\text{top, bottom}} &= 0,\ \mathbf{u} = (u, v), \\ p^{\text{right}} &= 0. \end{aligned}$$

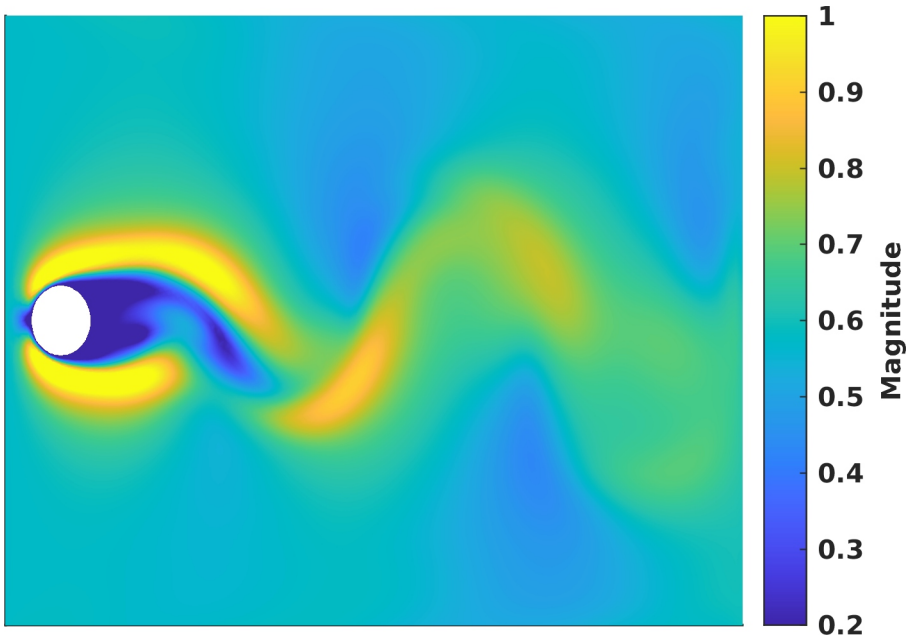
$$G : \mathbf{u}(\mathbf{x}, 0) \longrightarrow \mathbf{u}(\mathbf{x}, 10)$$



Results – 2D Laminar Flow past a Cylinder

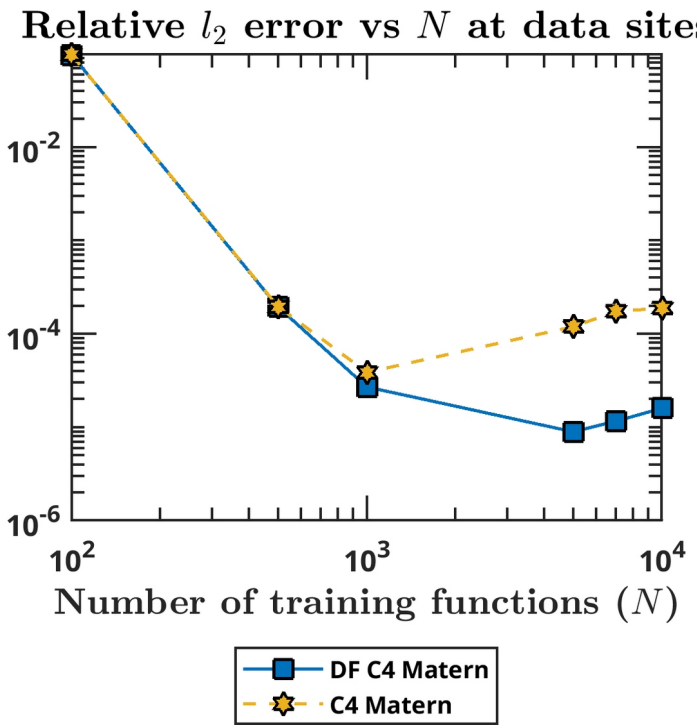
- Triangular mesh with 9520 points.
- Example input and output functions.

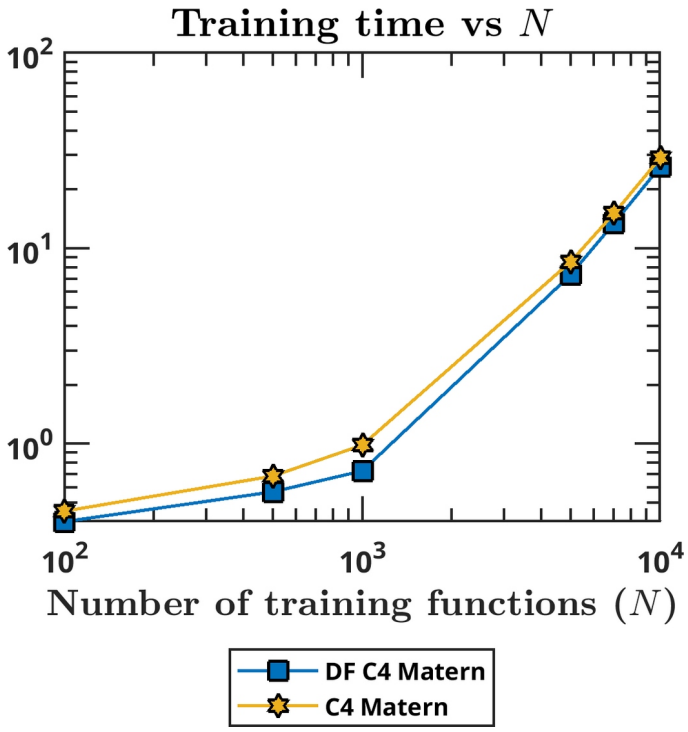
Input function

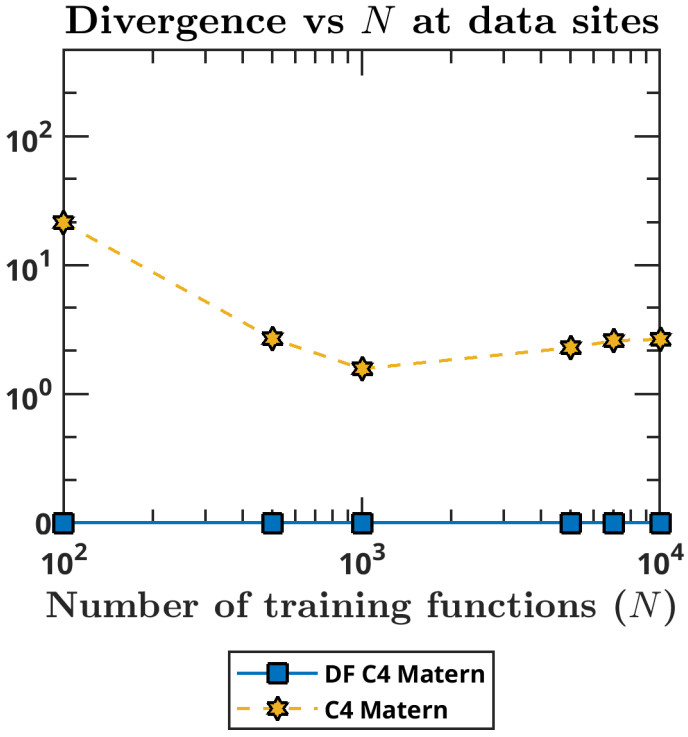


Output function

Results – 2D Laminar Flow past a Cylinder







Results – 2D Backward-facing step

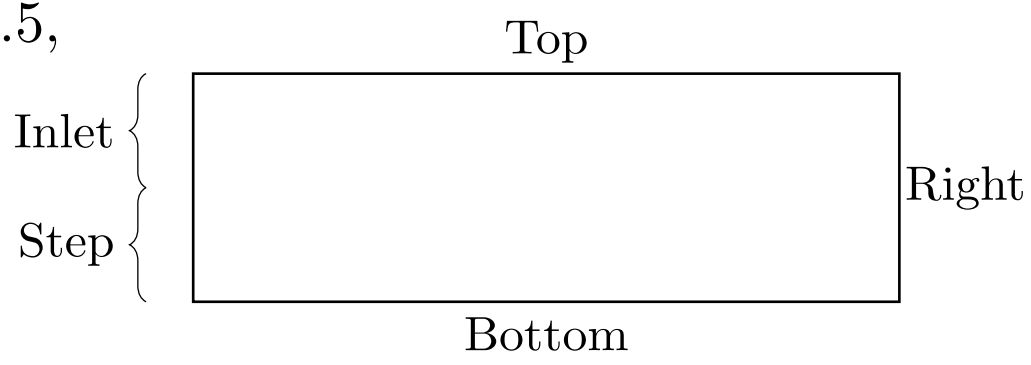
- $\Omega = [0, 15] \times [-0.5, 0.5]$
- Reynolds number range: 28 – 900.
- $\Delta t = 0.001,\ T = 5.$
- Initial condition:

$$\mathbf{u}^{\text{inlet}} = \nu\ y(0.5 - y), \quad 0 \le y \le 0.5,$$
$$\mathbf{u} = 0, \quad \text{everywhere else}$$

- Boundary condition:

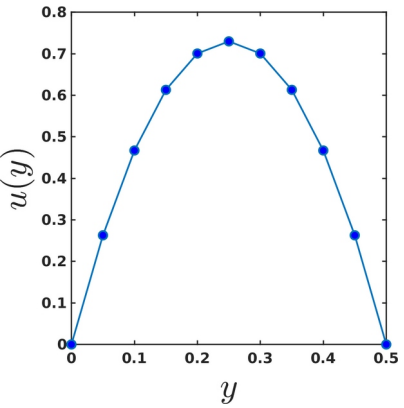
$$\mathbf{u}^{\text{inlet}} = \nu y(0.5 - y), \quad 0 \le y \le 0.5,$$
$$\mathbf{u}^{\text{step}} = 0,$$
$$\mathbf{u}^{\text{top, bottom}} = 0,$$
$$p^{\text{right}} = 0.$$

$$G : \mathbf{u}(\mathbf{x}, 0) \longrightarrow \mathbf{u}(\mathbf{x}, 5)$$



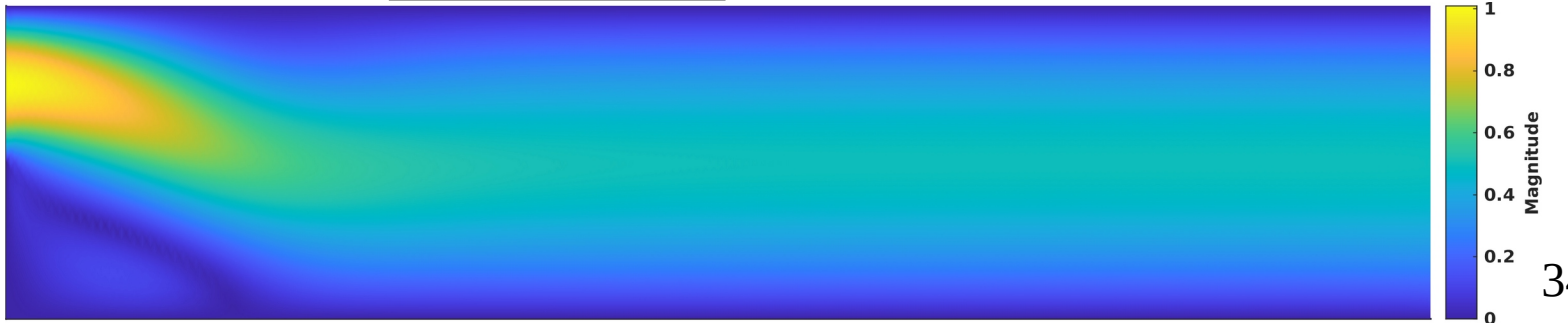
Results – 2D Backward-facing step

- Triangular mesh with 7359 points.
- Example input and output functions.



Input function

Output function



Results – 2D Backward-facing step