

DATA-DRIVEN REDUCED ORDER MODELING FOR APPLICATIONS IN COMPUTATIONAL HYDROLOGY

Sourav Dutta

Coastal & Hydraulics Laboratory

US Army Engineer Research & Development Center

sourav.dutta@erdc.dren.mil

Integrable Systems and Nonlinear Mechanics
UTRGV School of Mathematical & Statistical Sciences
November 16, 2021

In collaboration with:

Matthew W. Farthing (ERDC-CHL)
Peter Rivera-Casillas (ERDC-ITL)
Orie M. Cecil (ERDC-CHL)
Emma Perracchione (Genoa)
Mario Putti (Padova)



US Army Corps
of Engineers.



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Overview

1. Background: ERDC
2. Model Reduction: Motivation & Framework
3. Radial Basis Function-based ROM
4. Deep Learning-based ROM
5. ROM for Ship Simulation
6. Conclusion

WHAT IS ERDC?

The Engineer Research and Development Center ...



Mississippi Delta, 1927



Vicksburg, 1927

1929

Established by
Congress as the
Waterways
Experiment Station
(WES)



Today

Applied R&D Organization for the Corps of
Engineers, DoD

55% Engineers and Scientists

Civil Engineers, Physicists, Computer Scientists,
Mathematicians, Anthropologists ...

Computational Scientist

76% Advanced degrees

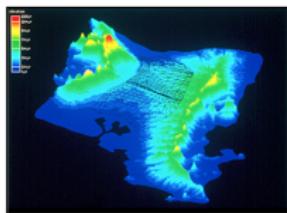
31% PhDs

CIVIL WORKS MISSION

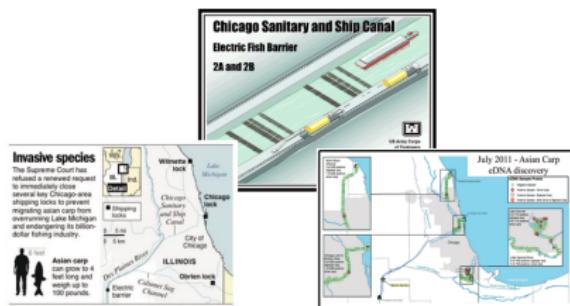


Flood and Storm Damage Reduction

Ice Hydraulics and Engineering



Environmental Contamination



Invasive Species

MILITARY APPLICATIONS

Force Projection

Airborne or vehicle-mounted sensors

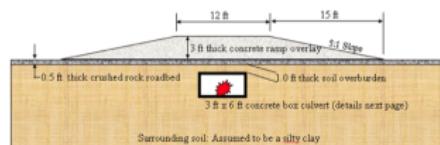
Electro-optical, infrared, and radar are most common

Flightlines are 10s of meters by 1000s of meters – generates a lot of imagery

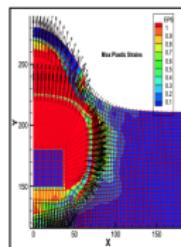
Automated target recognition (ATR) algorithms for image analysis



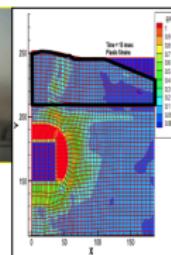
Force Protection



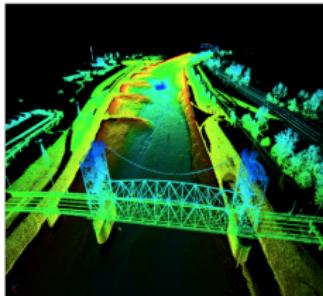
No Overlay



With Overlay

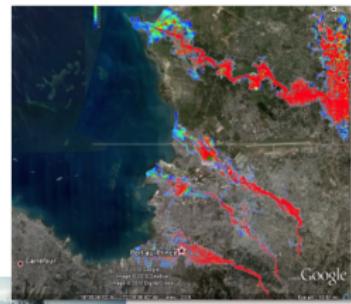


OPERATIONAL SUPPORT



Remote Sensing for
Environmental
Characterization

Flood Inundation Mapping and
Logistics Support for
Humanitarian Assistance

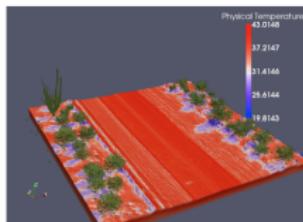


WHAT DO WE HAVE TO OFFER THE COMMUNITY?

Interesting problems

Large (and small) scale experimental facilities

HPC and computational modeling resources



Near surface simulation



Cray XC40/50 6 PFlop



LA Harbor scale model

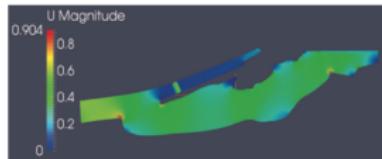


El Morro 1:75 model, 1979
breakwater study

Brandon Road Lock and Dam

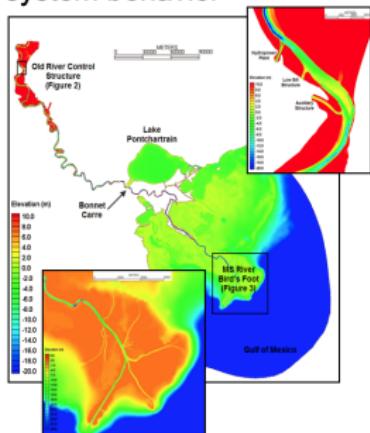


Bubble curtain simulation



MODELING CHALLENGES

Large domains needed to capture system behavior

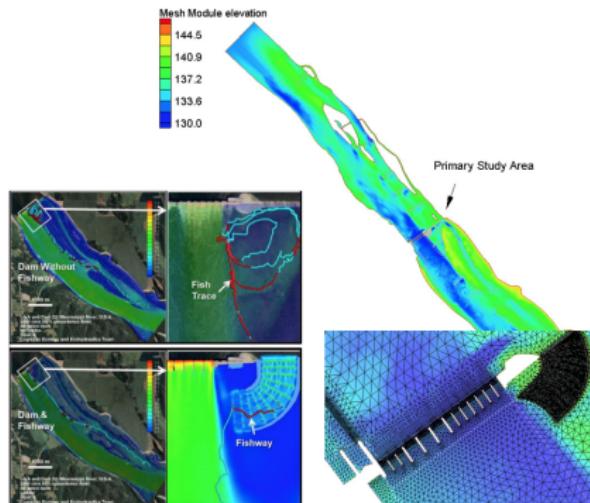


400 river miles from upstream of Old River Control to Gulf of Mexico

9.3+ million acres

100 m resolution in river

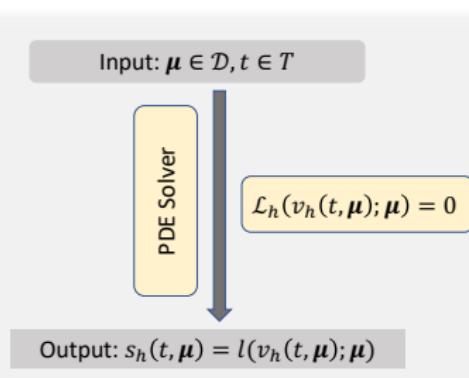
Detailed resolution and physics needed for engineering analysis



Base versus Plan Study for Fish Passage
Andy Goodwin and Dave Smith

Reduced order models

An **accurate** way to evaluate the solution at new parameter values and new time points at **reduced computational cost**.



Applications:

- ✓ Optimization/inversion/control
- ✓ Uncertainty quantification
- ✓ Multi-scale modeling
- ✓ Mobile computation
- ✓ In-situ/deployed modeling

Model order reduction

Consider the generic problem,

$$u_t + \mathcal{L}u + f(u) = g \quad \forall t \in [0, T]$$

which we use to solve for different time configurations.

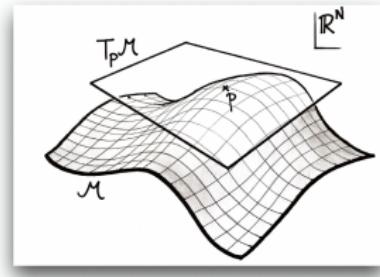
Using our favorite numerical method we obtain,

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{L}\mathbf{u} + \mathbf{f}(\mathbf{u}) = \mathbf{g}, \quad \dim(\mathbf{u}) = N \gg 1$$

This is accurate, **but expensive/slow** when repeated for different time discretizations or for different values of a parameter μ .

Model order reduction

Let's assume there exists \mathbf{V} such that,
 $\mathbf{u} \approx \mathbf{u}_{RB} = \mathbf{V}\mathbf{z}$, where $\mathbf{V}^T\mathbf{V} = \mathbf{I}$,
 $\dim(\mathbf{z}) = k \ll N$, and $\mathbf{V} \in \mathbb{R}^{N \times k}$.



Then using Galerkin projection we can recover solution for new system configurations as,

$$\underbrace{(\mathbf{V}^T \mathbf{M} \mathbf{V})}_{k \times k} \dot{\mathbf{z}} + \underbrace{(\mathbf{V}^T \mathbf{L} \mathbf{V})}_{k \times k} \mathbf{z} + \underbrace{\mathbf{V}^T \mathbf{f}}_k (\underbrace{\mathbf{V}\mathbf{z}}_N) = \underbrace{\mathbf{V}^T \mathbf{g}}_k$$

- For nonlinear problems, acceleration is much harder.
- Requires access to system operators \mathbf{L} , \mathbf{M} etc.

Non-intrusive model order reduction

What is non-intrusive model order reduction?

- ★ Full-order solver is a black box
- ★ ROM without projection

Approach:

- Recover the reduced basis using standard methods
- Compute the expansion coefficients by function approximation

$$\mathbf{u}(x, t) = \sum_i z_i(t) \mathbf{V}_i(x) \quad z_i(t) = \sum_j \alpha_j(t_j) \phi_j(t)$$

Non-intrusive model order reduction

What is non-intrusive model order reduction?

- ★ Full-order solver is a black box
- ★ ROM without projection

Approach:

- Recover the reduced basis using standard methods
- Compute the expansion coefficients by function approximation

$$\mathbf{u}(x, t) = \sum_i z_i(t) \mathbf{V}_i(x) \quad z_i(t) = \sum_j \alpha_j(t_j) \phi_j(t)$$

Proper Orthogonal Decomposition (POD)

Let $\mathbf{S} = [\mathbf{v}^1 \dots \mathbf{v}^{N_s}]$ be the snapshot matrix.

- ♦ Compute SVD: $\mathbf{S} = \widehat{\mathbf{U}} \widehat{\Sigma} \widehat{\mathbf{W}}^T$, where $\widehat{\Sigma} = \text{diag}(\sigma^1, \dots, \sigma^{N_s})$.
- ♦ Compute truncated global basis vectors, $\mathbf{U} \in \mathbb{R}^{N \times k}$.
- ♦ L^2 Optimality property: $\sum_{j=1}^{N_s} \| \mathbf{v}^j - \mathbf{U} \mathbf{U}^T \mathbf{v}^j \|_{\mathbb{R}^N}^2 = \sum_{j=k+1}^{N_s} (\sigma^j)^2$

Framework for non-intrusive PODRBF ROM

Idea: Combine POD projection with radial basis function interpolation

HFM: $\mathbf{M}\dot{\mathbf{v}} + \mathbf{L}\mathbf{v} + \mathbf{q}(\mathbf{v}, \dot{\mathbf{v}}) = \mathbf{v}_d(t), \quad \text{where } \mathbf{v} \in \mathbb{R}^{3N}$ (1)

Snapshots: $\mathbf{S}_i = (\mathbf{v}_i^1 - \bar{\mathbf{v}}_i, \dots, \mathbf{v}_i^{N_s} - \bar{\mathbf{v}}_i) \in \mathbb{R}^{N \times N_s}, 1 \leq i \leq 3$

SVD: $\mathbf{S}_i = \widehat{\mathbf{U}}_i \widehat{\Sigma}_i \widehat{\mathbf{W}}_i^T, \quad \text{where } \widehat{\Sigma}_i = \text{diag}(\sigma_i^1, \dots, \sigma_i^{N_s})$

POD-Projection: $\mathbf{z}^n = \mathbf{U}^T (\mathbf{v}^n - \bar{\mathbf{v}}), \quad \text{where } U_{ii} = \widehat{\mathbf{U}}_i|_{N \times m_i}, m_i \ll N$ (2)

NIROM : $\dot{\mathbf{z}} = \mathbf{F}(\mathbf{z}, t) \approx \sum_{i=1}^{N_s} \alpha_i(t) \phi(\|\mathbf{z} - \mathbf{z}_i\|)$ (3)

Reconstruction: $\mathbf{v}^m \approx \bar{\mathbf{v}} + \mathbf{U}\mathbf{z}^m$ (4)

Radial basis function interpolation

- Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \Omega \subset \mathbb{R}^d$,
- $f \in C^0$ such that $f|_X = \{f_1, f_2, \dots, f_N\} \in \mathbb{R}$.
- **RBF interpolant to $f|_X$** :
$$F(\mathbf{x}) = \sum_{i=1}^N \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|), \quad \phi : \mathbb{R} \setminus \mathbb{R}^- \rightarrow \mathbb{R}$$
- **Solve for α_i** : $F(\mathbf{x}_i) = f_i, (i = 1, \dots, N)$.

Radial basis function interpolation

- Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \Omega \subset \mathbb{R}^d$,
- $f \in C^0$ such that $f|_X = \{f_1, f_2, \dots, f_N\} \in \mathbb{R}$.

- RBF interpolant to $f|_X$:

$$F(\mathbf{x}) = \sum_{i=1}^N \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|), \quad \phi : \mathbb{R} \setminus \mathbb{R}^- \rightarrow \mathbb{R}$$

- Solve for α_i : $F(\mathbf{x}_i) = f_i, (i = 1, \dots, N)$.

Evolution of expansion coefficients in latent space:

$$z_i^{n+1} = z_i^n + \Delta t^{n+1} f_i^{n+1}(\mathbf{z}^n), \quad n \in \{0, \dots, N_s - 1\}, i \in \{1, \dots, k\}.$$

RBF NIROM: Approximate $f(\mathbf{z})$ by a RBF Interpolant $F(\mathbf{z})$ (Dutta *et al.*, 2021 [1])

$$\frac{z_j^m - z_j^{m-1}}{\Delta t} = F_j(\mathbf{z}^m) \equiv \sum_{l=1}^{N_m} \alpha_{j,l} \phi(||\mathbf{z}^m - \hat{\mathbf{z}}_l||), \quad j = 1, \dots, k; m = 1, \dots, N_s$$

S.Dutta, M.W.Farthing, E. Perrachhione, G. Savant, M. Putti,
"A greedy non-intrusive reduced order model for shallow water equations", **J. Comp. Phys.**, 439, 110378 (2021)

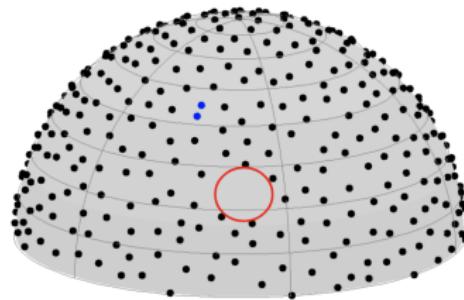
RBF kernel properties

We use the s.p.d. **Matérn C^0 kernel**

$$\phi(r) = e^{-cr}$$

with a constant shape factor

$c \approx h_{X,\Omega}/\sqrt{2}$, where $h_{X,\Omega}$ =**fill distance**,
$$h_{X,\Omega} = \max_{x \in \Omega} \min_{x_j \in X} \|x - x_j\|_2$$



RBF kernel properties

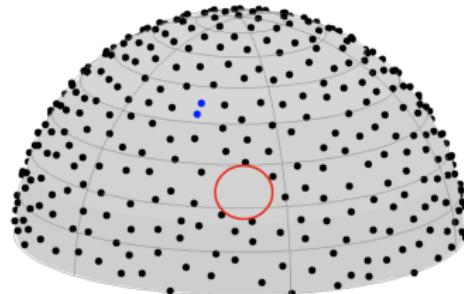
We use the s.p.d. **Matérn C^0 kernel**

$$\phi(r) = e^{-cr}$$

with a constant shape factor

$c \approx h_{X,\Omega}/\sqrt{2}$, where $h_{X,\Omega}$ = **fill distance**,

$$h_{X,\Omega} = \max_{x \in \Omega} \min_{x_j \in X} \|x - x_j\|_2$$



Optimal distribution of data points

Power function: Consider PD kernel Φ , $X = \{x_1, \dots, x_N\} \subset \Omega \subset \mathbb{R}^d$

$$P_X(x) = \sqrt{\Phi(x, x) - (\mathbf{b}(x))^T \mathbf{A}_X^{-1} \mathbf{b}(x)}, \quad x \in \Omega.$$

$$\text{where } \mathbf{b} = [\Phi(x, x_1), \dots, \Phi(x, x_N)]^T, \quad \mathbf{A}_X = [A_{ij}] = [\Phi(x_i, x_j)]$$

Interpolation error estimate:

$$|f(x) - F(x)| \leq P_X(x) \|f\|_{\mathcal{N}_\phi(\Omega)}.$$

Optimal selection of RBF centers

greedy algorithms

Result: \tilde{X} : Optimal set of RBF centers

Input : $X^0 = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{N_0}\}, N_0 < N_{max}$

$Y = \{\mathbf{z}_k | \mathbf{z}_k \in X \setminus X^0\}$

$L = \{m_j | j \leq m\}, |L| \ll m$: List of selected significant modes

Initial RBF system: F_0, A_0

p-greedy (De Marchi *et al.*, 2005[2])

while $\max\{P_{X_k}(\mathbf{z}_i)\} > \tau_p$ & $N_k < N_{max}$ **do**

Compute $P_{X_k}(\mathbf{z}_i) = \sqrt{\phi(0) - \mathbf{b}_i^T A_k \mathbf{b}_i}, \forall \mathbf{z}_i \in Y \setminus X_k$

Let $\mathbf{z}_{k+1} = \arg\max_{\mathbf{z}_i} \{P_{X_k}(\mathbf{z}_i)\}$

$X_{k+1} \leftarrow X_k \cup \{\mathbf{z}_{k+1}\}$

Re-compute RBF system: F_{k+1}, A_{k+1}

end

f-greedy (Schaback & Werner, '06[3], Wirtz & Haasdonk, '13[4])

Initial center: $\mathbf{z}_1 = \operatorname{argmax}_{\mathbf{z}_i \in X} \{|g(\mathbf{z}_i)| \equiv \|\mathbf{f}(\mathbf{z}_i)\|_2\}$

do

Compute $\xi_{X_k}(\mathbf{z}_i) = g(\mathbf{z}_i) - \tilde{F}^k(\mathbf{z}_i), \quad \forall \mathbf{z}_i \in X \setminus X_k$

Set $\mathbf{z}_{k+1} = \operatorname{argmax}_{\mathbf{z}_i \in X \setminus X_k} \{|\xi_{X_k}(\mathbf{z}_i)|\}$

$X_{k+1} \leftarrow X_k \cup \{\mathbf{z}_{k+1}\}$

Re-compute RBF system: $\tilde{F}^{k+1}, \mathbf{A}^{k+1}$

while $\left| \max\{\xi_{X_k}\} \right| > \tau_f$ & $k < N_{max}$

In the above, $\mathbf{b}_i = [\Phi(\mathbf{z}_i, \mathbf{z}_1), \dots, \Phi(\mathbf{z}_i, \mathbf{z}_{N_k})]^T, \forall \mathbf{z}_i \in Y \setminus X_k, X_k = \{\mathbf{z}_1, \dots, \mathbf{z}_{N_k}\}$

Optimal selection of RBF centers

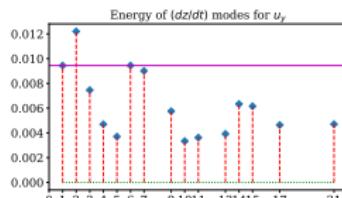
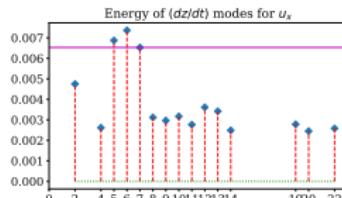
psr-greedy algorithm (Dutta et al., 2021[1])

Select a subset of modal coefficients, L

$$\hat{e}_j^i = \sum_{k=0}^{M-1} \left(\frac{d z_{k,j}}{dt} \Big|_i \right)^2,$$

$$\left(\sum_{j=0}^{j=m_i^j} \hat{e}_j^i \right) / \left(\sum_{j=0}^{j=m_i} \hat{e}_j^i \right) \leq \tau_{greedy}.$$

$$P_X(\mathbf{x}) |f(\mathbf{x}) - F(\mathbf{x})| \leq P_X^2(\mathbf{x}) \|f\|_\Phi \leq \|f\|_\Phi$$



Result: \tilde{X} : Optimal set of RBF centers

Input : $X = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M\}, M > N_{max}$

$L = \{j \mid j \leq m_i^j\}, |L| \ll m$: List of selected significant modes

Initial center: $\mathbf{z}_1 = \operatorname{argmax}_{\mathbf{z}_i \in X} \{\Phi(\mathbf{z}_i, \mathbf{z}_i) | f_j(\mathbf{z}_i) | : j \text{ is the first mode in } L\}$

for j in L **do**

do

Compute $\xi_{X_k}^j(\mathbf{z}_i) = P_{X_k}(\mathbf{z}_i) |f_j(\mathbf{z}_i) - F_j^k(\mathbf{z}_i)|, \forall \mathbf{z}_i \in X \setminus X_k$

Set $\mathbf{z}_{k+1} = \operatorname{argmax}_{\mathbf{z}_i \in X \setminus X_k} \{\xi_{X_k}^j(\mathbf{z}_i)\}$

$X_{k+1} \leftarrow X_k \cup \{\mathbf{z}_{k+1}\}$

Re-compute RBF system: $\mathbf{F}^{k+1}, \mathbf{A}^{k+1}$

while $|\max\{\xi_{X_k}^j\}| > \tau_{psr} \text{ & } k < N_{max}$

end

S.Dutta, M.W.Farthing, E. Perrachhione, G. Savant, M. Putti,
“A greedy non-intrusive reduced order model for shallow water equations”, J. Comp. Phys., 439, 110378 (2021)

High Fidelity Model: Shallow Water Equations

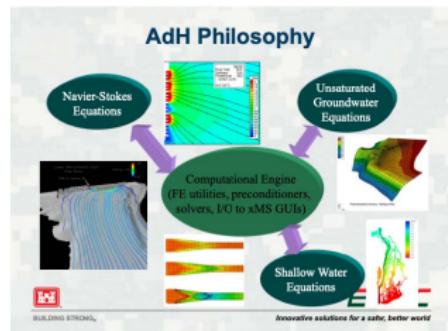
2D Shallow Water Equations

$$R^{SW,2d} \equiv \frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{p} + \mathbf{r} = 0, \quad \mathbf{q} = [h, hu_x, hu_y]^T$$

$$\mathbf{p}_x = \begin{Bmatrix} u_x h \\ u_x^2 h + (1/2)gh^2 - h(\sigma_{xx}/\rho) \\ u_x u_y h - h(\sigma_{yx}/\rho) \end{Bmatrix},$$

$$\mathbf{p}_y = \begin{Bmatrix} u_y h \\ u_x u_y h - h(\sigma_{xy}/\rho) \\ u_y^2 h + (1/2)gh^2 - h(\sigma_{yy}/\rho) \end{Bmatrix},$$

$$\mathbf{r} = \begin{Bmatrix} gh \frac{\partial h_b}{\partial x} + gh \left[\left(n_{mn}^2 u_x \sqrt{u_x^2 + u_y^2} \right) / h^{4/3} \right] - f_c h u_y \\ gh \frac{\partial h_b}{\partial y} + gh \left[\left(n_{mn}^2 u_y \sqrt{u_x^2 + u_y^2} \right) / h^{4/3} \right] + f_c h u_x \end{Bmatrix},$$



Adaptive Hydraulics (AdH)

Fully adaptive, CG Finite Element engine with SUPG stabilization

Trahan *et al.* 2018 [5]

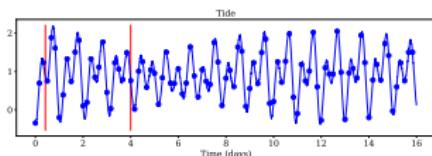
Shallow water examples

San Diego bay

Red River



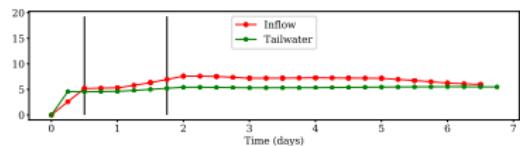
$N = 6311, T = 50$ hrs
 $\Delta t^{train} = 100$ secs, $\Delta t^{test} = 50$ secs



Tidal flow data



$N = 12291, T = 9$ hrs
 $\Delta t^{train} = 90$ secs, $\Delta t^{test} = 30$ secs

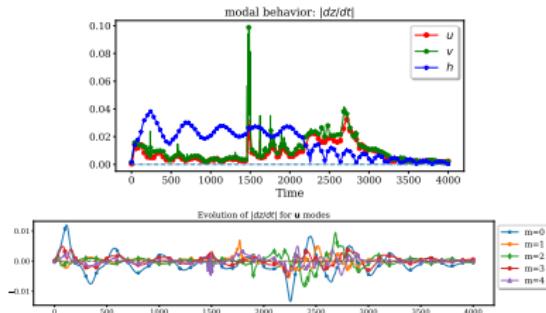


Hydrograph data

Numerical results: Greedy center selection

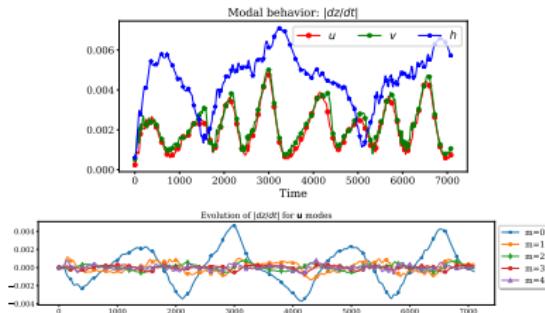
Red River problem

$N = 12291, N_s = 5002,$
 $k_1 = 557, k_2 = 772, k_3 = 763.$



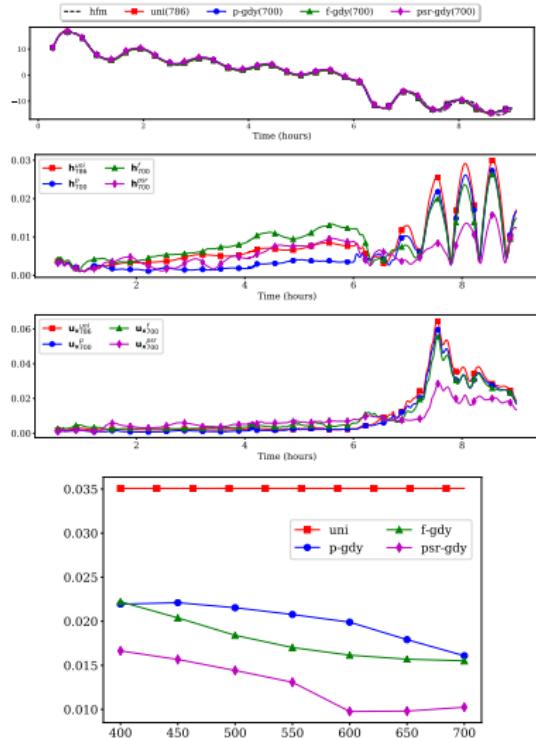
San Diego Bay problem

$N = 6311, N_s = 7200,$
 $k_1 = 473, k_2 = 593, k_3 = 598.$

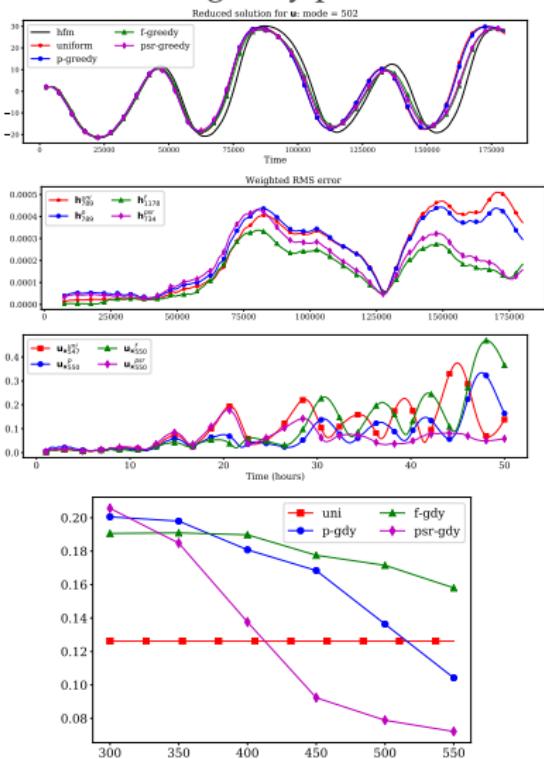


Numerical results: Root mean square errors

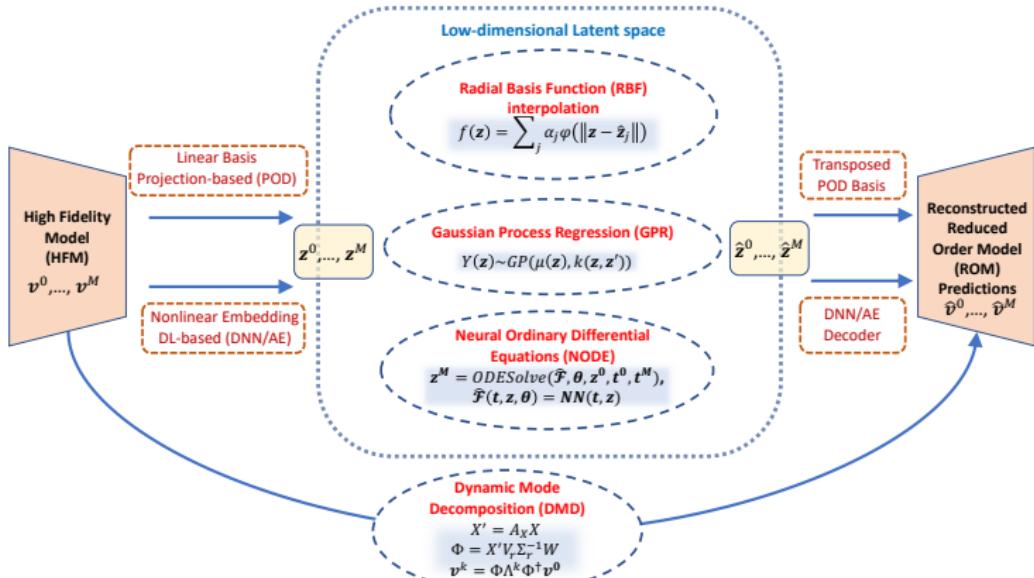
Red River problem



San Diego Bay problem



Data-driven Model Order Reduction



ROM development steps

- Extract low-dimensional latent space
- Model latent space system dynamics
- Reconstruct full-order predictions

RBF (Xiao et al. 2015 [6], Xiao et al. 2017, [7], Dutta et al. 2021 [1])

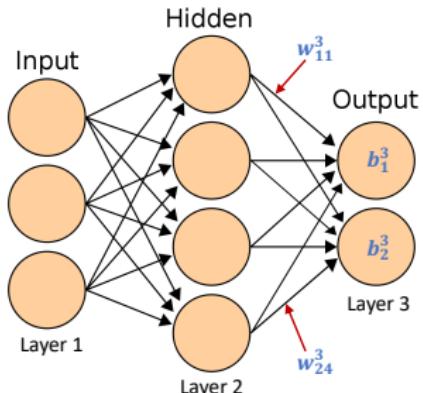
DMD (Kutz et al. 2016[8], Bistrian & Navon 2017 [9])

GPR (Guo & Hesthaven 2019 [10])

NODE (Chen et al. 2018 [11], Dutta et al. 2021 [12], Dutta et al. 2021 [13])

Artificial Neural Networks

Universal Function Approximation (Cybenko, 1989): Finite sums of continuous sigmoidal functions are dense in $C(I_d)$.



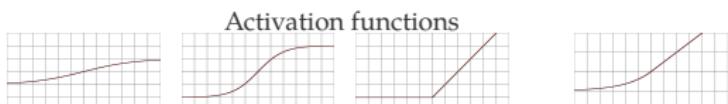
Activation of j^{th} neuron in l^{th} layer:

$$a_j^l = \mathcal{F} \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right)$$

a_j^l - neuron activations

w_{jk}^l, b_j^l - weights and biases

\mathcal{F} - activation function.



Sigmoid

$$\frac{1}{1+e^{-x}}$$

Tanh

$$\frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Relu

$$\max(0, x)$$

Elu

$$\begin{cases} \alpha(e^x - 1), & x \leq 0 \\ x, & x > 0 \end{cases}$$

Cost function:

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2$$

x - given input vector

y - given output vector

$a^L(x)$ - NN output vector

Autoencoders: Nonlinear Dimension Reduction

Autoencoder: Feedforward neural network to learn the identity mapping.
 $\mathbf{h} : \mathbf{v} \mapsto \tilde{\mathbf{v}}$ such that $\tilde{\mathbf{v}} \approx \mathbf{v}$.

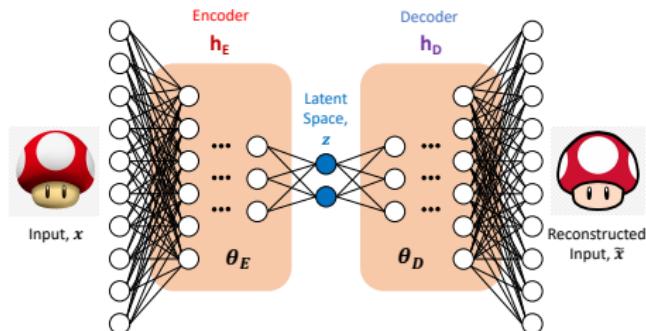
Consists of two parts -

Encoder: $\mathbf{z} = \mathbf{h}_E(\mathbf{v}; \theta_E)$,

Decoder: $\tilde{\mathbf{v}} = \mathbf{h}_D(\mathbf{z}; \theta_D)$.

$$\mathbf{h} : \mathbf{v} \mapsto \mathbf{h}_D \circ \mathbf{h}_E(\mathbf{v}). \quad (5)$$

This autoencoder is trained by computing the optimal values of the parameters (θ_E^*, θ_D^*) that minimize the reconstruction error over all the training data



$$\theta_E^*, \theta_D^* = \underset{\theta_E, \theta_D}{\operatorname{argmin}} \mathcal{L}(\mathbf{v}, \tilde{\mathbf{v}}). \quad (6)$$

$\mathcal{L}(\mathbf{v}, \tilde{\mathbf{v}})$ is a measure of discrepancy between \mathbf{v} and $\tilde{\mathbf{v}}$.

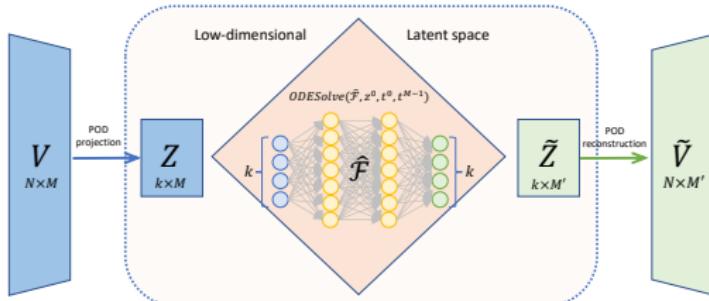
Undercomplete Autoencoder: $\dim(\mathbf{z}) = m \ll N = \dim(\mathbf{v})$

Lee & Carlberg, J. Comp.Phys., 2020; Kim, Choi, et al, arXiv, 2020

Champion, Lusch, et al, PNAS, 2019; Plaut, arXiv, 2018

Gonzalez et al, arXiv, 2018; Otto & Rowley, SIAM J.Appl.Dyn.Sys., 2019; Mauli, Lusch, et al, Phys.Fluids, 2021.

Neural Ordinary Differential Equations (NODE)



Latent space dynamics: $\frac{dz}{dt} = \mathcal{F}(t, z(t))$, with $z(0) = z^0, z \in \mathbb{R}^d$.

Goal: Obtain $\widehat{\mathcal{F}}(t, z, \omega) = NN(t, z)$ such that $\frac{dz}{dt} \approx \widehat{\mathcal{F}}(t, z, \omega)$

1. Compute time trajectory: $\bar{z}^{M-1} = ODESolve(\widehat{\mathcal{F}}, \omega, z^0, t^0, t^{M-1})$
2. Evaluate loss: $\mathcal{L}(\widehat{\mathcal{F}}(t, \bar{z}, \omega), \bar{z}^{M-1})$
3. Compute gradients $\frac{\partial \mathcal{L}}{\partial \omega}(t=0)$ using the adjoint method to update network parameters ω .

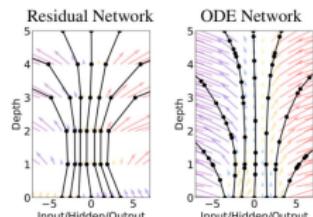


Figure from Chen et al. 2018 [11]

S. Dutta, P. Rivera-Casillas, M.W. Farthing, " Neural ordinary differential equations for data-driven reduced order modeling of environmental hydrodynamics", *Proceedings of AAAI MLPS 2021*, arXiv:2104.13962 (2021)

S. Dutta, P. Rivera-Casillas, O.M. Cecil, M.W. Farthing, E. Perracchione, M. Putti" Data-driven reduced order modeling of environmental hydrodynamics using deep autoencoders and neural ODEs", *Proceedings of IXth International Conference on Coupled Problems 2021*, arXiv:2107.02784 (2021)

Classical Data-driven ROMs

Dynamic Mode Decomposition (DMD)

1. Split the time series:
 $\mathbf{X} = [\mathbf{v}^1, \mathbf{v}^2, \dots, \mathbf{v}^{N_S-1}]$, and $\mathbf{X}' = [\mathbf{v}^2, \mathbf{v}^3, \dots, \mathbf{v}^{N_S}]$
DMD computes the leading eigendecomposition of the best fit linear operator \mathbf{A} s.t. $\mathbf{X}' = \mathbf{AX}$.
2. Compute r -rank SVD: $\mathbf{X} \approx \mathbf{U}\Sigma\mathbf{V}^*$
3. Compute $\tilde{\mathbf{A}} = \mathbf{U}^*\mathbf{A}\mathbf{U} = \mathbf{U}^*\mathbf{X}'\mathbf{V}\Sigma^{-1}$, and its eigenpairs:
 $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\Lambda$.
4. DMD modes: $\Psi = \mathbf{X}'\mathbf{V}\Sigma^{-1}\mathbf{W}$.
5. Predict: $\hat{\mathbf{v}}(t^*) \approx \sum_{k=1}^r \psi_k \exp(\omega_k t^*) b_k = \Psi \exp(\Omega t^*) \mathbf{b}$,
where $\mathbf{b} = \Psi^\dagger \hat{\mathbf{v}}_1$, and
 $\Omega = \text{diag}(\omega)$ where $\omega_k = \ln(\lambda_k)/\Delta t$.

Kutz *et al.* 2016 [8], Bistrian & Navon 2017 [14]

Classical Data-driven ROMs

Dynamic Mode Decomposition (DMD)

1. Split the time series:
 $\mathbf{X} = [\mathbf{v}^1, \mathbf{v}^2, \dots, \mathbf{v}^{N_S-1}]$, and $\mathbf{X}' = [\mathbf{v}^2, \mathbf{v}^3, \dots, \mathbf{v}^{N_S}]$
DMD computes the leading eigendecomposition of the best fit linear operator \mathbf{A} s.t. $\mathbf{X}' = \mathbf{AX}$.
2. Compute r -rank SVD: $\mathbf{X} \approx \mathbf{U}\Sigma\mathbf{V}^*$
3. Compute $\tilde{\mathbf{A}} = \mathbf{U}^*\mathbf{A}\mathbf{U} = \mathbf{U}^*\mathbf{X}'\mathbf{V}\Sigma^{-1}$, and its eigenpairs:
 $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\Lambda$.
4. DMD modes: $\Psi = \mathbf{X}'\mathbf{V}\Sigma^{-1}\mathbf{W}$.
5. Predict: $\hat{\mathbf{v}}(t^*) \approx \sum_{k=1}^r \psi_k \exp(\omega_k t^*) b_k = \Psi \exp(\Omega t^*) \mathbf{b}$,
where $\mathbf{b} = \Psi^\dagger \hat{\mathbf{v}}_1$, and
 $\Omega = \text{diag}(\omega)$ where $\omega_k = \ln(\lambda_k)/\Delta t$.

Kutz *et al.* 2016 [8], Bistrian & Navon 2017 [14]

Gaussian Process Regression (GPR)

Observed data: $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$

1. Noisy regression function: $y_i = f(\mathbf{x}_i) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$
2. GP prior: $f(\mathbf{x}) \sim GP(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$
3. Prior joint Gaussian: $\mathbf{y}|\mathbf{X} \sim \mathcal{N}(m(\mathbf{X}), \mathbf{K}_y)$, where
 $\mathbf{K}_y = \text{cov}[\mathbf{y}|\mathbf{X}] = \kappa(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I}_M$
4. GP posterior: $f^*(\mathbf{s})|\mathbf{s}, \mathbf{X}, \mathbf{y} \sim GP(m^*(\mathbf{s}), \mathbf{K}^*(\mathbf{s}, \mathbf{s}'))$
Posterior mean: $m^*(\mathbf{s}) = m(\mathbf{s}) + \kappa(\mathbf{s}, \mathbf{X})\mathbf{K}_y^{-1}(\mathbf{y} - m(\mathbf{X}))$,
Covariance: $\mathbf{K}^*(\mathbf{s}, \mathbf{s}') = \kappa(\mathbf{s}, \mathbf{s}') - \kappa(\mathbf{s}, \mathbf{X})\mathbf{K}_y^{-1}\kappa(\mathbf{X}, \mathbf{s}')$

Kernels: $\kappa(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{m=1}^d \frac{(x_m - x'_m)^2}{l_m^2}\right)$

Hyperparameters: $\theta = \{\sigma_f, \sigma_y, l_1, \dots, l_d\}$

$\theta_{opt} = \arg \max_{\theta} \log p(\mathbf{y}|\mathbf{X}, \theta)$, where

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{X}, \theta)p(\mathbf{f}|\mathbf{X}, \theta) d\mathbf{f}$$

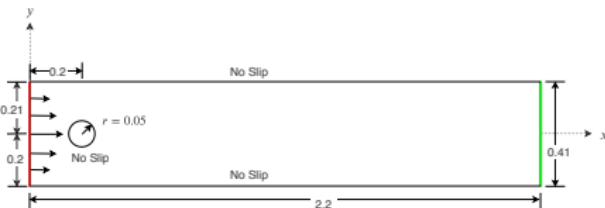
Guo & Hesthaven 2019 [10]

High Fidelity Models: Navier Stokes

2D Flow past a cylinder

$$R^{m,2d} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p,$$

$$R^{c,2d} = \nabla \cdot \mathbf{u}$$



$$\mathbf{u}|_{\Gamma_i} = 0, \quad i \in \{\text{top, bottom, cylinder}\}$$

$$\mathbf{u}(0, y) = \left(4U \frac{(0.21 - y)(y - 0.2)}{0.41^2}, 0 \right)$$

$$\nu \nabla \mathbf{u} \cdot \mathbf{n} = 0, \quad p \mathbf{n} = 0, \quad \text{outflow}$$



$$N = 14605, T = 6 \text{ s}, DT = 10^{-4} \text{ s}$$
$$Re = 100, U = 1.5 \text{ m/s}, \nu = 0.001$$

Simulation conditions:

Training with $t \in \{2.500, 2.508, 2.516, \dots, 4.996\}$ s, $\Delta t = 0.008$ s

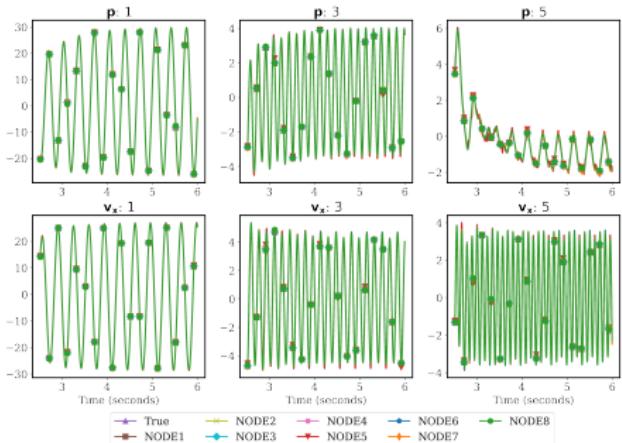
Compute solution for $t \in \{2.500, 2.502, 2.504, \dots, 5.998\}$ s, $\Delta t = 0.002$ s

Numerical Results: NODE architecture search

≈ 400 architectures

- **Solvers:** Euler, midpoint, huen, RK4, Dopri5, Adams ...
- **Scaling:** MinMax, Standard, ...
- **Optimizers:** RMSProp, Adam, SGD, Adamax, Adagrad, ...
- **LR Scheduler:** Stepwise, Exponential

S. Dutta, P. Rivera-Casillas, M.W. Farthing, "Neural ordinary differential equations for data-driven reduced order modeling of environmental hydrodynamics", **Proceedings of AAAI MLPS 2021**, arXiv:2104.13962 (2021)



Comparison of projection coefficients

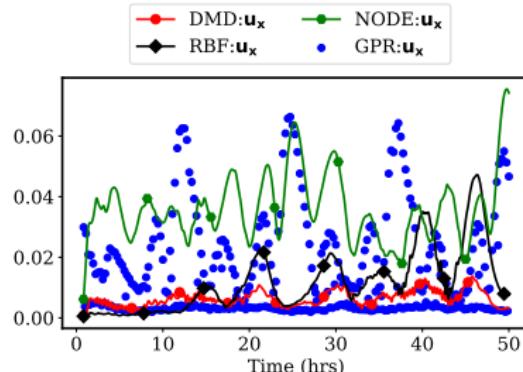
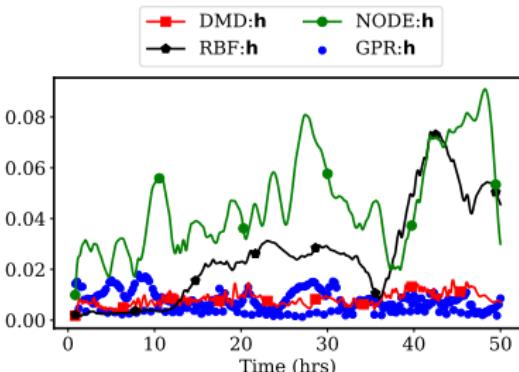
Id	Layers	Units	Act.	LR decay steps, rate	Scaling	Augmented	MSE	Training
NODE1	1	256	elu	10000, 0.3	No	No	8.87×10^{-4}	24.45 hrs
NODE2	1	256	tanh	5000, 0.7	Yes	No	9.01×10^{-4}	24.56 hrs
NODE3	1	512	elu	5000, 0.5	No	No	8.86×10^{-4}	24.39 hrs
NODE4	1	256	tanh	10000, 0.25	Yes	Yes	9.02×10^{-4}	22.97 hrs
NODE5	4	64	tanh	5000, 0.5	Yes	No	9.19×10^{-4}	27.98 hrs
NODE6	1	256	elu	10000, 0.1	No	No	8.87×10^{-4}	24.13 hrs
NODE7	2	128	elu	5000, 0.5	No	No	8.86×10^{-4}	25.80 hrs
NODE8	1	512	tanh	5000, 0.5	Yes	Yes	9.00×10^{-4}	24.77 hrs

Numerical Results: SW Fast replay

San Diego results

$t \in [0.69, 50.69]$ hrs, Prediction $\Delta t = 50$ secs

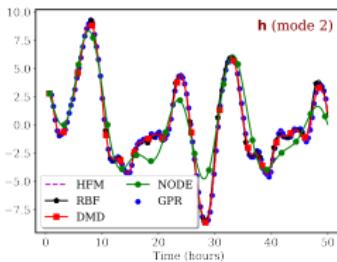
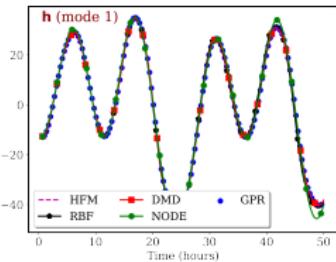
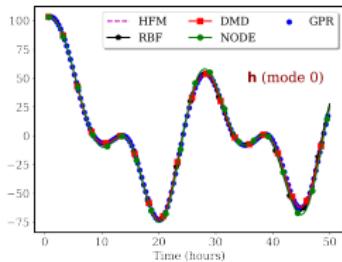
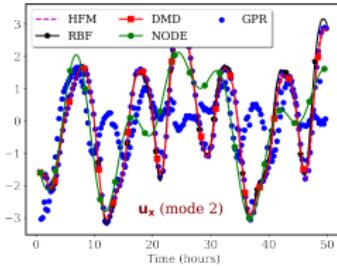
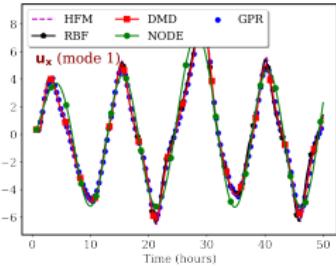
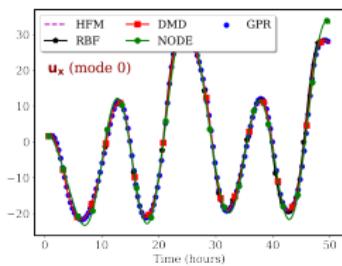
	HFM	RBF	DMD	GPR	NODE
DOF Reduction	18933 0.86%	163(19, 73, 71) 1.82%	345(115, 115, 115) 1.37%	260(33, 115, 112) 1.37%	264(36, 115, 113) 1.39%
Training Δt (secs)	25	100	100	200	100



RMSE of temporal prediction

Numerical Results: SW Fast replay

San Diego results

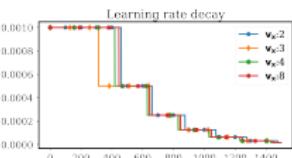
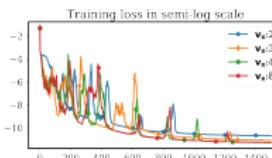


Comparison of expansion coefficients

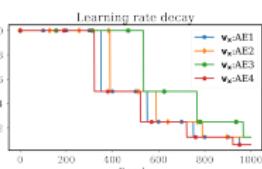
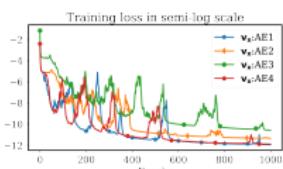
Numerical Results: AE architecture search

Autoencoder architectures

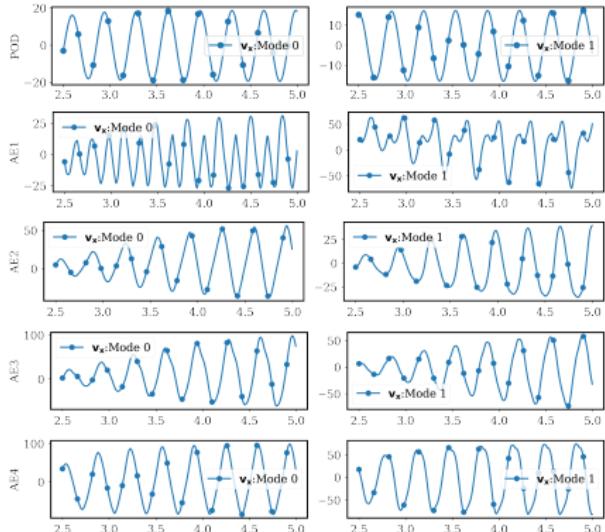
Id	Units	Encoder	Decoder	Scaling	MSE	Training
	(p, v_x, v_y)	linear, elu, elu, ...	elu, tanh, ...			
AE1	(5, 8, 7)	linear	sigmoid	[0, 1]	2.398e-6	9.38 min
AE2	(2, 2, 2)	linear	sigmoid	[0, 1]	4.196e-6	9.56 min
AE3	(2, 2, 2)	linear	tanh	[-1, 1]	2.499e-6	9.29 min
AE4	(2, 3, 3)	linear	sigmoid	[0, 1]	3.107e-6	9.07 min



Variable latent space dimension (using AE3)



Different AE architectures



Expansion coefficients of the first two latent space modes for x-velocity

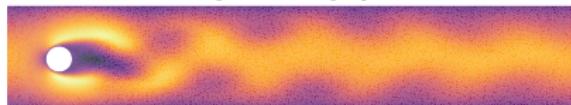
S. Dutta, P. Rivera-Casillas, O.M. Cecil, M.W. Farthing, E. Perrachione, M. Putti "Data-driven reduced order modeling of environmental hydrodynamics using deep autoencoders and neural ODEs", *Proceedings of IXth International Conference on Coupled Problems 2021*, arXiv:2107.02784 (2021)

Numerical Results: NS Fast replay

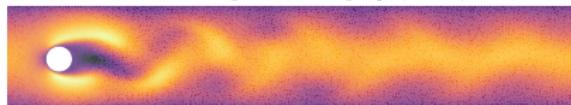
NODE configurations

Id	Lyr	Units	Act.	LR steps, rate	Scaling	Aug.	MSE	Training
	1-4	32-512	tanh, elu,...	5k-25k, 0.1- 0.9				
AE1-NODE1	1	256	elu	10k, 0.3	No	No	2.30e-5	28.80 hrs
AE1-NODE2	1	256	tanh	5k, 0.7	Yes	No	1.34e-4	28.69 hrs
AE1-NODE3	1	512	elu	5k, 0.5	No	No	1.97e-5	29.17 hrs
AE1-NODE4	1	256	tanh	10k, 0.25	Yes	Yes	1.49e-4	28.27 hrs
AE1-NODE5	4	64	tanh	5k, 0.5	Yes	No	1.33e-4	33.08 hrs

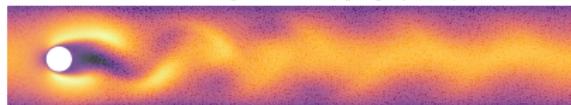
DMD solution: v_x at $t=4.90$ seconds, v_x range = [-0.399, 2.1]



POD-NODE solution: v_x at $t=4.90$ seconds, v_x range = [-0.547, 2.1]



AE-NODE solution: v_x at $t=4.90$ seconds, v_x range = [-0.54, 2.1]



-0.5 0.0 0.5 1.0 1.5 2.0

DMD error: v_x at $t=4.90$ seconds, error range = [-0.295, 0.2]



POD-NODE error: v_x at $t=4.90$ seconds, error range = [-0.0941, 0.092]



AE-NODE error: v_x at $t=4.90$ seconds, error range = [-0.0156, 0.018]



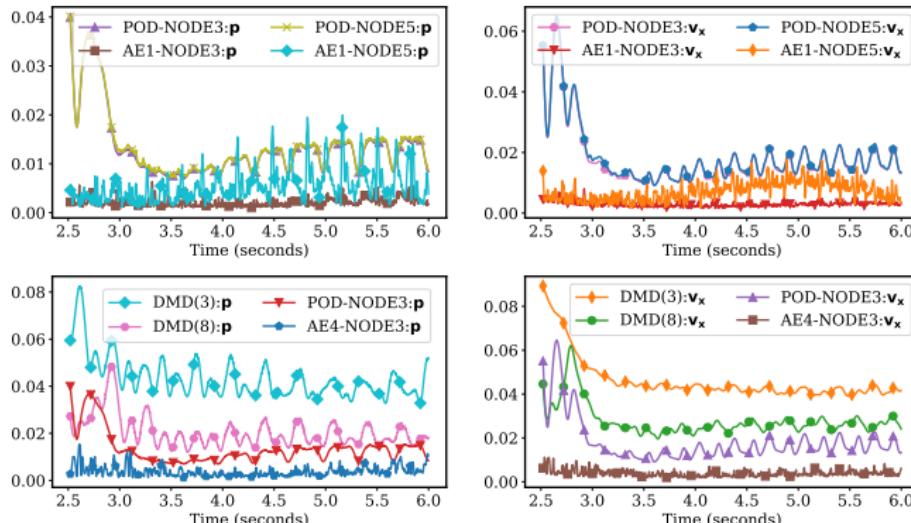
-0.2 -0.1 0.0 0.1 0.2

Comparison of NIROM solutions and errors at $t = 4.90$ s

Numerical Results: NS Fast replay

Training: $t \in [2.5, 4.996]$ s, $\Delta t = 0.008$ s, Prediction: $t \in [2.5, 5.998]$, $\Delta t = 0.002$ s

	HFM	DMD(8)	DMD(3)	POD	AE1	AE4
DOF	43815	24(8, 8, 8)	9(3, 3, 3)	20(5, 8, 7)	20(5, 8, 7)	8(2, 3, 3)
Reduction	0.05%	0.14%	0.02%	0.04%	0.02%	



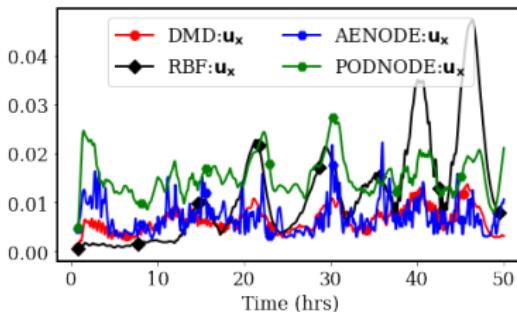
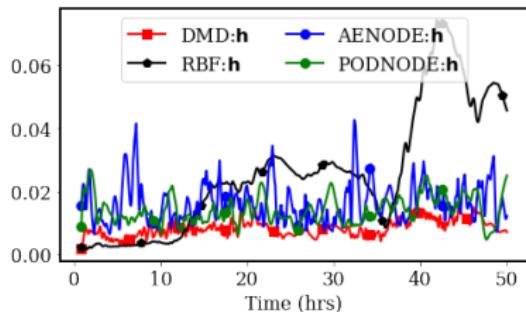
RMSE of temporal prediction

Numerical Results: SW Fast replay

San Diego results

$t \in [0.69, 50.69]$ hrs, Training $\Delta t = 50$ secs, Prediction $\Delta t = 50$ secs

	HFM	RBF	DMD	PODNODE	AENODE
DOF	18933	163(19, 73, 71)	345(115, 115, 115)	265(36, 115, 113)	90(30, 30, 30)
Reduction		0.86%	1.82%	1.40%	0.48%

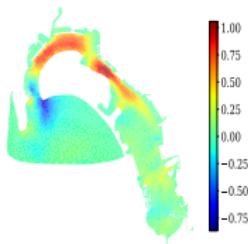


RMSE of temporal prediction

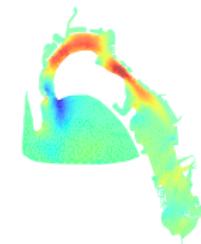
Numerical Results: SW Fast replay

San Diego results

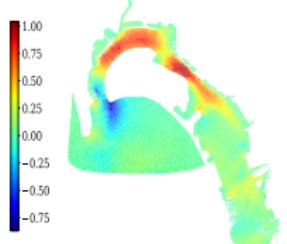
RBF solution at t=17.36 hrs
 $-0.90163 < u_x < 1.09034$



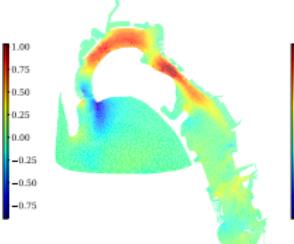
DMD solution at t=17.36 hrs
 $-0.90537 < u_x < 1.07212$



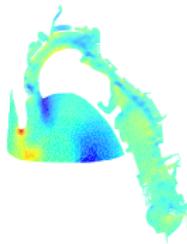
POD-NODE solution at t=17.36 hrs
 $-0.92640 < u_x < 1.10591$



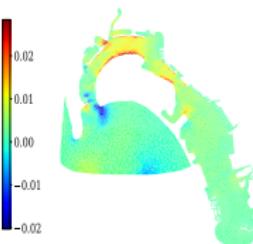
AE-NODE solution at t=17.36 hrs
 $-0.88352 < u_x < 1.07144$



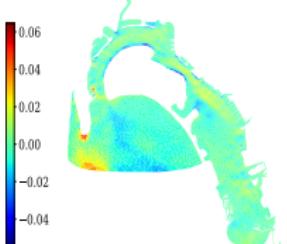
$-0.020851 < \text{RBF } u_x \text{ Error} < 0.030679$
Rel. Error 2-norm : 0.013062



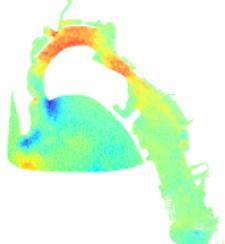
$-0.064100 < \text{DMD } u_x \text{ Error} < 0.082534$
Rel. Error 2-norm : 0.082029



$-0.130506 < \text{POD-NODE } u_x \text{ Error} < 0.135954$
Rel. Error 2-norm : 0.049377



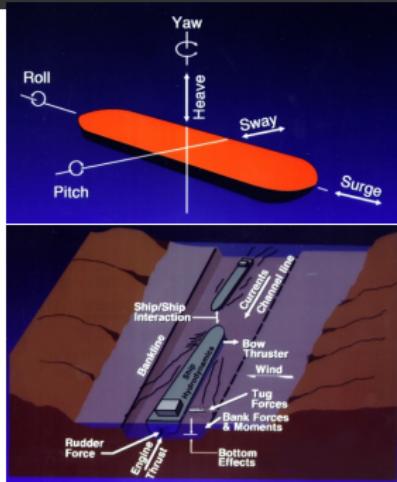
$-0.037588 < \text{AE-NODE } u_x \text{ Error} < 0.042090$
Rel. Error 2-norm : 0.027899



Simulation of Ship Dynamics

What is ship simulation?

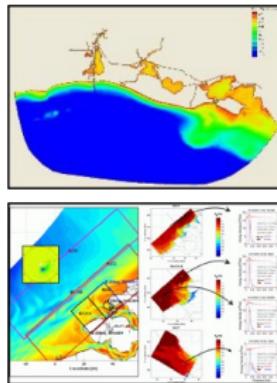
1. Reliable, flexible, and cost-effective design and evaluation tool for simulating vessel navigation.
2. Used for port and channel design, ship operations, and pilot training in no-risk (virtual) environment.
3. High definition visual scene and representative bridge facilities provide a realistic environment for pilots and tug-masters.
4. Incorporates data and models to accurately model an area of interest.



Numerics of Ship Simulation

Environmental conditions

- Channel definition – COE District Office, Bathymetric surveys
- Currents – From a validated numerical model simulation.
- Waves – Usually hindcasting study determines appropriate height/period/direction to be used during simulations
- Wind – Wind roses and pilot experience



Houston Ship Channel (30 minutes)

Challenges:

- Accurate vessel response to environmental forces.
- Impact of vessel dynamics on environment.
- Impact on water control structures, shoreline, wetlands.
- Impact on navigation conditions due to bathymetry changes.
- Nonlinear interactions between waves generated by multiple vessels.

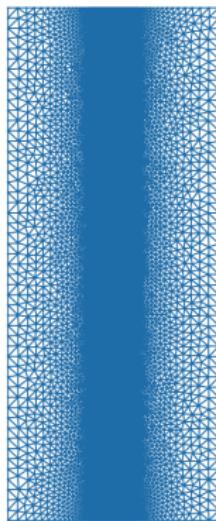


Model Problem

$$0 \leq x \leq 200, 0 \leq y \leq 500$$

$$1.0 \leq \Delta x, \Delta y \leq 10.0$$

unstructured mesh



$$\text{Velocity} = (0, 1.0) \text{ m/s}, \Delta t = 1.0 \text{ s}$$

Boat length = 10 m, width = 4 m, draft = 4 m



ROM challenges: Transport dominated phenomena, Adaptive and unstructured mesh

Autoencoder & Shifted Supervised Encoder

Autoencoder

- ★ Maps true snapshot to itself

Encoder: 4 hidden layers with **relu** activation

Final layer with **linear** activation

Decoder: 4 hidden layers with **relu** activation

Final layer with **sigmoid** activation

Alternate data points for training and validation

Loss function: $\lambda \text{ RMSE} + (1 - \lambda) \text{ Huber}$ ($\lambda = 0.9$)

LR Scheduler based on **val_loss**

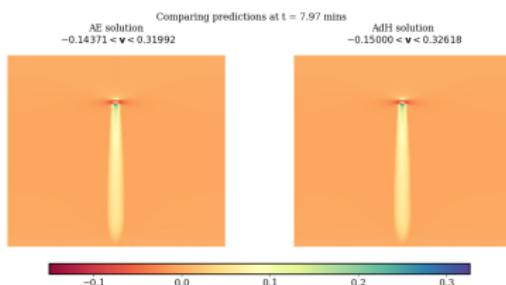
Latent dimensions: $(h, v_x, v_y) \equiv (5, 20, 20)$

Batch size = 32

$$h \text{ MSE} = 1.24 \times 10^{-6}$$

$$v_x \text{ MSE} = 2.72 \times 10^{-8}$$

$$v_y \text{ MSE} = 5.15 \times 10^{-8}$$



Shifted SE

- ★ Maps true snapshot to shifted snapshot

Encoder: 4 hidden layers with **relu** activation

Final layer with **linear** activation

Decoder: 4 hidden layers with **relu** activation

Final layer with **linear** activation

Alternate data points for training and validation

Loss function: $\lambda \text{ RMSE} + (1 - \lambda) \text{ Huber}$ ($\lambda = 0.9$)

LR Scheduler based on **val_loss**

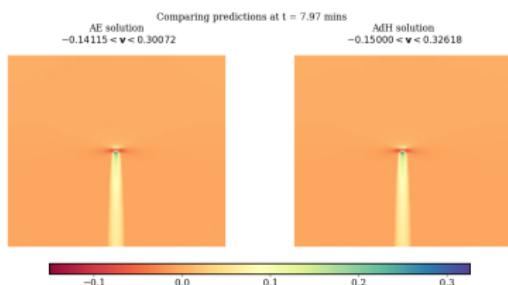
Latent dimensions: $(h, v_x, v_y) \equiv (10, 30, 30)$

Batch size = 16

$$h \text{ MSE} = 8.56 \times 10^{-5}$$

$$v_x \text{ MSE} = 1.39 \times 10^{-7}$$

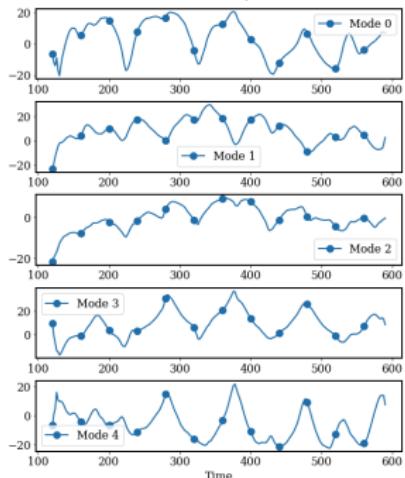
$$v_y \text{ MSE} = 3.38 \times 10^{-6}$$



Autoencoder & Shifted Supervised Encoder

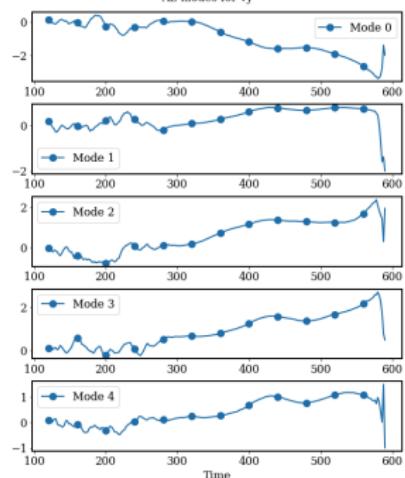
Autoencoder

AE modes for vy

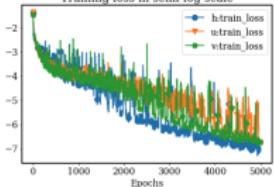


Shifted SE

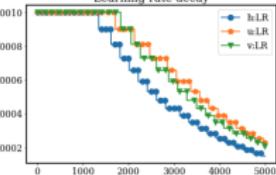
AE modes for vy



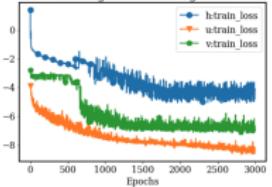
Training loss in semi-log scale



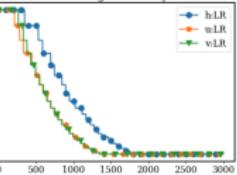
Learning rate decay



Training loss in semi-log scale



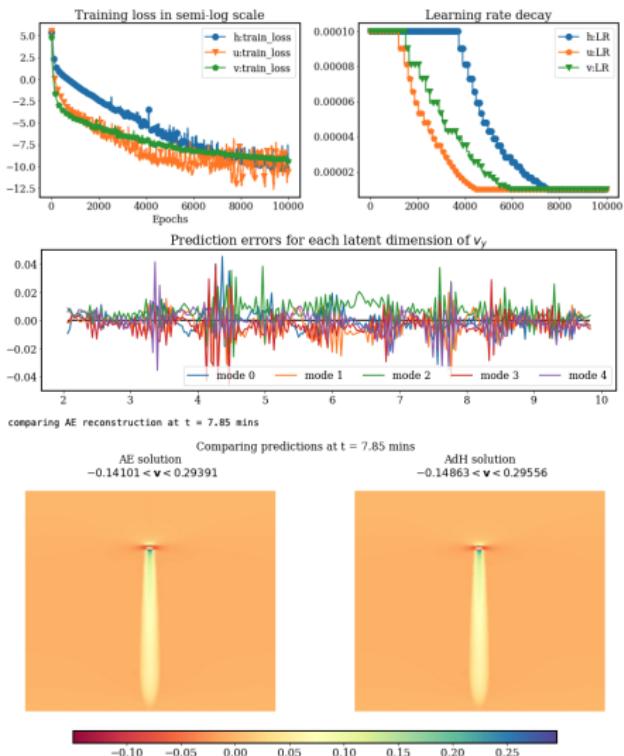
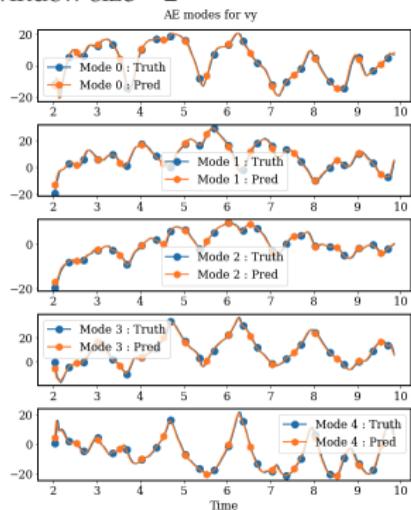
Learning rate decay



AE+LSTM model

2 LSTM layers with **relu** activation
Final Dense layer with **linear** activation
Loss function: **MSE**
Optimizer: **Adam(0.0001)**
LR Scheduler based on **loss**
LSTM units: $(h, v_x, v_y) \equiv (80, 200, 200)$
Batch size = 64

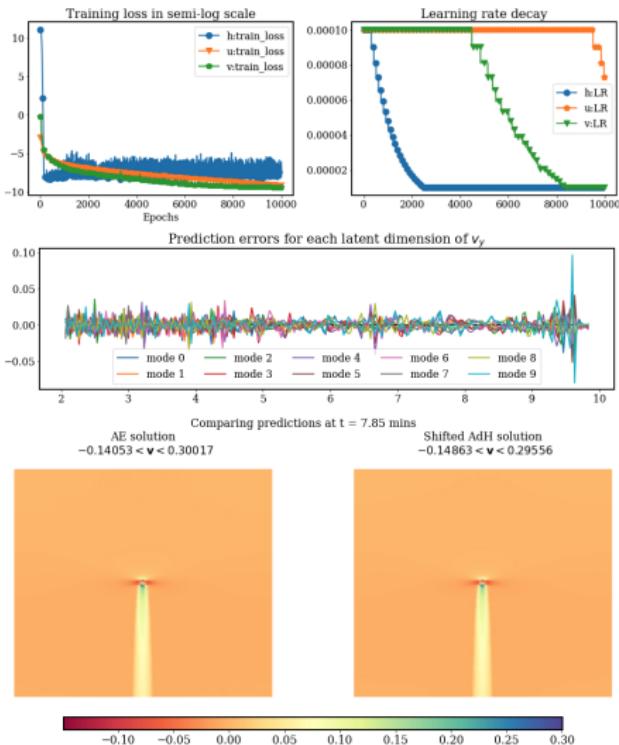
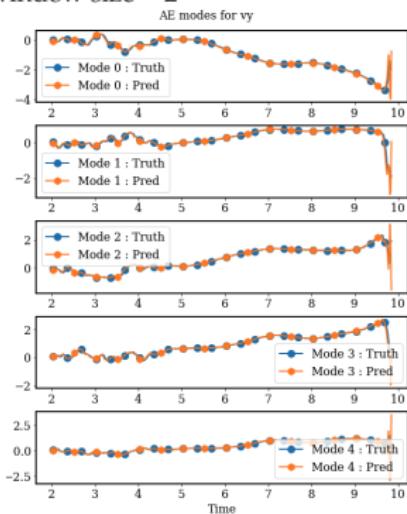
Window size = 2



Shifted SE+LSTM model

2 LSTM layers with **relu** activation
Final Dense layer with **linear** activation
Loss function: **MSE**
Optimizer: **Adam(0.0001)**
LR Scheduler based on **loss**
LSTM units: $(h, v_x, v_y) \equiv (80, 200, 200)$
Batch size = 64

Window size = 2



Conclusions and Future Work

Radial Basis Function:

- *psr-greedy* algorithm generates an efficient RBF approximation.
- Higher-order RBF NIROM generates more robust long-time predictions.
- Natural extension to high-dimensional, parametric ROM development.

Neural ODEs:

- High degree of flexibility of model design for learning latent space dynamics.
- Training and hyperparameter search are expensive.
- Moderately extrapolative temporal predictions. Some normalization may be required.
- Better suited for combining with physics-informed NN, nonlinear AE reduced basis etc.
- NODE-based methodology for parametric MOR being explored.

Gaussian Process Regression

- Trains well on coarser training data, generates error estimate.
- Training is sensitive to regularization parameter, data normalization and choice of kernel.
- A tensor-decomposition-based approach admits extension to parametric ROMs. Guo & Hesthaven 2019

Dynamic Mode Decomposition:

- Works well with periodic or quasi-periodic time-dependent problems.
- Efficient embedding of temporal dynamics. No natural extension to parametric ROM.

Autoencoders

- Impressive flexibility for learning latent spaces for transport dominated phenomena.
- Training and hyperparameter searches are relatively cheap.
- Can be easily combined with time series learning models and trained end-to-end.
- Exploring Conv AE and Graph NN for better compression and robustness even in the presence of unstructured meshes.

Questions ?

Contact

Sourav Dutta - Sourav.Dutta@erdc.dren.mil

Papers

- Dutta, Farthing, et al. (2021), *J. Comp. Phys.*, vol. 439, pp. 110378.
- Dutta, Rivera, Farthing (2021), AAAI-MLPS 2021, *arXiv:2104.13962[cs.LG]*
- Dutta, Rivera, et al. (2021), Coupled 2021, *doi:10.23967/coupled.2021.017*.
- Dutta, Rivera, Cecil, Farthing (2021), *Software Impacts*, vol. 10, pp. 100129



US Army Corps
of Engineers.



INFORMATION
TECHNOLOGY
LABORATORY



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Code

- ⌚ https://github.com/erdc/podrbf_nirom
- ⌚ https://github.com/erdc/node_nirom
- ⌚ <https://github.com/erdc/pynirom/tree/v0.5.0>

References I

-  S. Dutta, M. W. Farthing, E. Perracchione, G. Savant, and M. Putti, "A greedy non-intrusive reduced order model for shallow water equations," *J. Comp. Phys.*, vol. 439, p. 110378, 2021.
-  S. De Marchi, R. Schaback, and H. Wendland, "Near-optimal data-independent point locations for radial basis function interpolation," *Advances in Computational Mathematics*, vol. 23, no. 3, pp. 317–330, 2005.
-  R. Schaback and J. Werner, "Linearly constrained reconstruction of functions by kernels with applications to machine learning," *Advances in Computational Mathematics*, vol. 25, p. 237, Jul 2006.
-  D. Wirtz and B. Haasdonk, "A Vectorial Kernel Orthogonal Greedy Algorithm," *Dolomites Research Notes on Approximation*, vol. 6, pp. 83–100, 2013.
-  C. Trahan, G. Savant, R. Berger, M. Farthing, T. McAlpin, L. Pettey, G. Choudhary, and C. Dawson, "Formulation and application of the adaptive hydraulics three-dimensional shallow water and transport models," *J. Comput. Phys.*, vol. 374, pp. 47–90, 2018.
-  D. Xiao, F. Fang, A. G. Buchan, C. C. Pain, I. M. Navon, and A. Muggeridge, "Non-intrusive reduced order modelling of the Navier-Stokes equations," *Computational Methods in Applied Mechanics and Engineering*, vol. 293, pp. 522–541, 2015.
-  D. Xiao, F. Fang, C. C. Pain, and I. M. Navon, "A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications," *Comput. Methods Appl. Mech. Eng.*, vol. 317, pp. 868–889, 2017.
-  J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Philadelphia, PA: Society for Industrial and Applied Mathematics, third ed., 2016.
-  D. Bistrian and I. Navon, "Randomized Dynamic Mode Decomposition for non-intrusive reduced order modelling," *arXiv:1611.04884v1*, 2016.
-  M. Guo and J. S. Hesthaven, "Data-driven reduced order modeling for time-dependent problems," *Comput. Methods Appl. Mech. Eng.*, vol. 345, pp. 75–99, 2019.

References II

-  R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural Ordinary Differential Equations," *ArXiv*, vol. 109, no. NeurIPS, pp. 31–60, 2018.
-  S. Dutta, P. Rivera-Casillas, and M. W. Farthing, "Neural Ordinary Differential Equations for Data-Driven Reduced Order Modeling of Environmental Hydrodynamics," in *Proc. AAAI 2021 Spring Symp. Comb. Artif. Intell. Mach. Learn. with Phys. Sci.*, (Stanford, CA, USA), CEUR-WS, 2021.
-  S. Dutta, P. Rivera-Casillas, O. M. Cecil, M. W. Farthing, E. Perracchione, and M. Putti, "Data-driven reduced order modeling of environmental hydrodynamics using deep autoencoders and neural ODEs," in *Proc. IXth Int. Conf. Comput. Methods Coupled Probl. Sci. Eng. (COUPLED Probl. 2021)*, (Barcelona, Spain), pp. 1–16, International Center for Numerical Methods in Engineering (CIMNE), 2021.
-  D. Bistrian and I. Navon, "The method of dynamic mode decomposition in shallow water and a swirling flow problem," *International Journal for Numerical Methods in Fluids*, p. DOI:10.1002/fld.4257, 2017.