DOMAIN DECOMPOSITION FOR ENHANCEMENT OF REDUCED-ORDER MODELS

B. Abylkhani,[1] *D. Dwivedi,*[2] *S.B. Yabusaki,*[3] *&*
D. M. Tartakovsky[1,*]

[1]*Department of Energy Science and Engineering, Stanford University, California, USA*

[2]*Lawrence Berkeley National Laboratory, Berkeley, California, USA*

[3]*Pacific Northwest National Laboratory, Richland, Washington, USA*

*Address all correspondence to: D. M. Tartakovsky, Department of Energy Science and Engineering, Stanford University, California, USA, E-mail: tartakovsky@stanford.edu

Original Manuscript Submitted: mm/dd/yyyy; Final Draft Received: mm/dd/yyyy

Decision-support systems for environmental management of coastal areas must account for brine and seawater dynamics. Physics-based models of these phenomena are computationally expensive, which limits their usefulness for decision-making under uncertainty. Data-driven modeling techniques, such as extended dynamic mode decomposition (xDMD), ameliorate these challenges. We demonstrate that xDMD, equipped with a novel domain-decomposition component, effectively represents a validated, real-world, coupled nonlinear seawater inundation model. It serves as an efficient surrogate of process-based simulations, capable of accurate reproduction and reconstruction of missing pressure and salinity data in the interpolation regime. It effectively predicts low-rank pressure distributions (repeated dynamics) but struggles to forecast long-term salinity dynamics (cumulative evolution). The addition of domain decomposition improves the robustness and accuracy of xDMD, with the overlapping domain approach outperforming the non-overlapping one in the projection accuracy. In our experiments, xDMD is 1700 times faster than the process-based model and requires 800 times less storage, while efficiently capturing pressure and salinity dynamics.

KEY WORDS: *surrogate, domain decomposition, ensemble generation, coupled processes*

1. INTRODUCTION

Rising sea levels and more frequent, intense storm surges pose serious challenges to freshwater resources management, infrastructure protection, and resilience of coastal communities (Azevedo de Almeida and Mostafavi, 2021). Process-based models are critical to addressing these issues as they provide robust predictive understanding of a coastal system by capturing nonlinear interactions between hydrological, geological, and climatic processes. These models form the basis for decision-support systems in coastal environmental management. Despite their utility, such models are often constrained by high computational cost and significant data storage requirements. To overcome these limitations, data-driven surrogates have emerged as an effective alternative. Surrogates leverage the insights and outputs of process-based models while offering computational efficiency and reduced resource demands (Lucia et al., 2004).

Such surrogates are built using statistical, data-driven, and machine learning (ML) approaches aimed at reducing the computational burden of solving coupled subsurface flow and transport equations. Advanced ML models, such as support vector machines, random forests (RFs), artificial neural networks (ANNs), and genetic algorithms (GAs), are able to handle high-dimensional and nonlinear relationships in hydrological systems. These models have been applied to map groundwater pollution vulnerability, predict contaminant transport, and design groundwater remediation strategies (Zhou and Tartakovsky, 2021; Zhou et al., 2022). However, ML-based approaches often require substantial computational resources, large training datasets, and parameter tuning. RFs, ANNs, and GAs often regarded as black-box models, offering limited interpretability of the underlying physical processes which poses a significant challenge in decision-making contexts (Aria et al., 2021). These factors increase the complexity of the modeling process and heightens the risk of overfitting and suboptimal performance.

In contrast, data-driven ROMs built, e.g., via dynamic mode decomposition (DMD) leverage process-based models and are known for their ability to balance predictive accuracy and computational efficiency. DMD approximates complex, high-dimensional nonlinear dynamic systems by reducing spatial dimension with an optimal low-dimensional linear operator. DMD is especially useful for identifying patterns and tracking changes over time in complex datasets, making it applicable to various fields (Hess et al., 2023). Standard DMD is modified in various ways to address challenges with predictive accuracy, robustness, and efficiency for different process-based models. One such variant is extended DMD (xDMD); it improves the representation of physical systems governed by both linear and non-linear PDEs with sources and sinks, for which standard DMD fails (Lu and Tartakovsky, 2020). We investigate xDMD's ability to accurately reproduce coupled flow and transport phenomena in coastal aquifers, including spatiotemporal salinity and pressure distributions and total salt mass dynamics. Additionally, its performance in interpolation and extrapolation, computational efficiency, and ability to capture and differentiate dynamic activity—particularly in distinguishing slower and faster dynamic regions within a coastal system—require further investigation.

We explore these questions and evaluates the feasibility of xDMD, building on the previously validated process-based model for the Beaver Creek site in Washington (Yabusaki et al., 2020), as a lightweight, efficient, and robust tool for managing coastal groundwater systems. Additionally, we apply the domain decomposition technique, which enhances the robustness of xDMD in projecting salinity in a system with different dynamics.

2. METHODOLOGY

2.1 Numerical Experiment

2.1.1 Governing Equation

For flood-type (vertical) seawater intrusion scenarios, the coupled PDE system involves solving Richards' equation for flow and advection-dispersion equation for solute transport. General Darcy's flux, describing isothermal, single-phase, variable saturated flow (VSF) in porous media with a saturation-based formulation along with Richards' equation, employed for solving time-dependent water table profiles in the saturation/water content form (Bedient et al., 2013),

$$\begin{aligned} \mathbf{q} &= -\frac{kk_{\mathrm{r}}(s)}{\mu}\nabla(P - \rho g z) \\ &\frac{\partial}{\partial t}(\phi s \eta) + \nabla \cdot (\eta \mathbf{q}) = Q_w, \end{aligned} \tag{1a}$$

This is coupled with the transport equation, where solute mass conservation in a variably saturated medium is governed by the advection-dispersion equation,

$$\frac{\partial c}{\partial t} = \nabla \cdot (\mathbf{D}\nabla c) - \nabla \cdot \left(\frac{\mathbf{q}c}{\phi s}\right) + Q_{\mathrm{r}}. \tag{1b}$$

The system parameters and system states in these equations are defined in Table 1.

TABLE 1: System parameters and system states in the governing equations.

Symbol	**Quantity**	**Units**
$\mathbf{q}$	Darcy's flux	L/T
k	Intrinsic permeability	L^2
$k_r(s)$	Relative permeability	–
s	Aqueous phase saturation	L^3/L^3
μ	Dynamic viscosity of water	M/(LT)
P	Pore-water pressure	M/LT^2
ρ	Water density	M/L^3
g	Gravity acceleration	L/T^2
z	Elevation head (positive downward)	L
η	Molar water density	$kmol/L^3$
ϕ	Medium porosity	–
Q_w	Source/sink term	$kmol/L^3T$
c	Solute concentration	mol/M
D	Diffusion/dispersion tensor	L^2/T
Q_r	Source/sink term	$kmol/L^3T$
ϕ	Porosity	–
s_r	Residual saturation	–
α	Inverse of air-entry value	L^{-1}
m	Pore-shape parameter	–

In numerical models, the coupled flow and transport equations are discretized and solved simultaneously. This concurrent solution approach, crucial for flooding scenarios like seawater intrusion, enhances efficiency by capturing the interdependence of these processes. We used PFLOTRAN, an open-source, massively parallel subsurface flow and reactive transport simulator designed for high-performance computing environments (PFLOTRAN Development Team, 2024). It employs a fully implicit backward Euler approach for time integration, combined with a Newton-Krylov method for solving the nonlinear equations governing flow and transport. Finite-volume spatial discretization is used to compute pressure and concentration distributions in each cell throughout the simulation period.

2.1.2 Numerical Model

The accuracy of data-driven surrogates like DMD depends on data quality and quantity. To generate a high-quality data set, it is imperative to use a physics-based model that has been validated and calibrated to the available site/field data and existing site conditions. We utilize a previously published, validated, and history-matched model developed for floodplain salinization near Beaver Creek, WA (Yabusaki et al., 2020). The simulation was performed using PFLOTRAN to model 2D cross-sectional flow and salinity transport within the existing floodplain, which has been inundated by tidal river influences for four years. Model goodness-of-fit was evaluated using a variety of common evaluation statistics, group statistics, and graphical evaluation techniques. The floodplain model successfully captured many characteristics of observed floodplain well water levels and salinities across the floodplain (Yabusaki et al., 2020), thereby validating its robustness and accuracy.

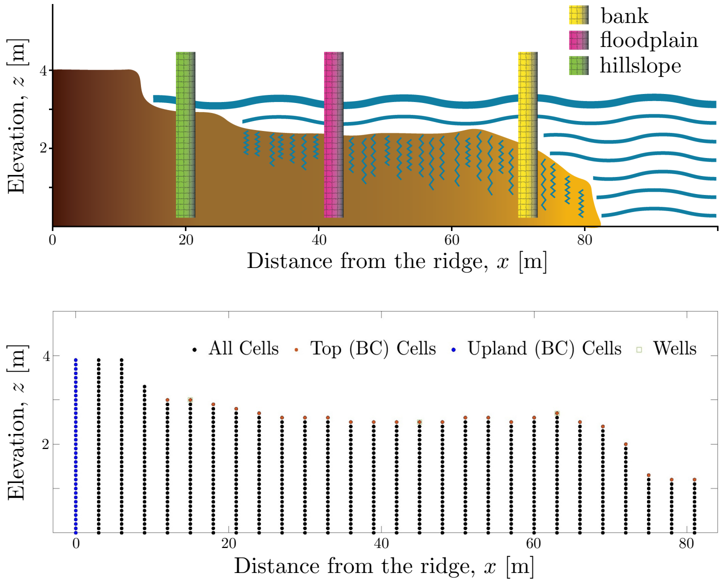

FIG. 1: Two-dimensional (x, z) simulation domain representing a coastal floodplain (Yabusaki et al., 2020). **Top:** The vertical bars indicate locations of the wells collecting measurements of water pressure and salinity. **Bottom:** The simulation domain is discretized with numerical cells, which are subsequently subdivided into classes during a domain decomposition.

The floodplain's (x, z) transect is 84 m by 4 m, and hosts three observational wells (Fig. 1). The stream serves as the sole source of salinity in the floodplain. Salinity enters the domain laterally through the surface water-groundwater interface and vertically via infiltration through the ground surface during inundation events (Fig. 1). The stream, located adjacent to the floodplain, is tidally influenced and provides a time-dependent boundary condition that drives the dynamics of flow and transport in the domain. Observed salinity in the stream was specified as a concentration accompanying water entering the floodplain, either through lateral flow across the floodplain-stream interface or infiltration through the ground surface during inundation events.

Field measurements were obtained from these wells and from the creek (water level and salinity) over the course of one year. The entire model domain was represented by a single, homogeneous silty clay soil material following the methodology of Yabusaki et al. (2020), which adopted a homogeneous domain to simplify the representation of floodplain dynamics while focusing on the interplay of variably saturated flow and salinity transport with time-dependent boundary conditions. The material properties were guided by observed soil texture (Carsel and Parrish, 1988), and the van Genuchten parameters along with the Mualem relative permeability function were used to characterize the soil's hydraulic properties. The dependence of water density and viscosity on saltwater content and temperature was modeled with the methodology of Yabusaki et al. (2020), which found the salinity-density effects on simulations to be negligible. This assumption was validated by testing the equations of state formulations (Batzle and Wang, 1992) against a scenario with constant salinity, and the numerical outputs showed negligible differences. This negligible impact of salinity-density coupling in the modeled system was further confirmed through discussions with the original authors. Material properties such as permeability, porosity, and van Genuchten water retention function parameters are collated in Table 2.

TABLE 2: Parameter values for the floodplain simulation.

Parameter	Value	Units
k	$4.86 \cdot 10^{-13}$	m^2
D	$1.0 \cdot 10^{-9}$	m^2/s
ϕ	0.45	–
s_r	0.15	–
α	$2.0 \cdot 10^{-4}$	1/Pa
m	0.45	–
$N_x \times N_z$	28×40	–
Δt	0.25	hr
T	20	yr

The PFLOTRAN infiltration boundary condition was used on the top surface of the model, allowing pressure-driven infiltration when the ground surface is inundated and exfiltration when the water pressure in surface cells exceeds atmospheric pressure. Along the upland boundary, Dirichlet pressure boundary conditions were imposed to simulate hillslope groundwater contributions. On the stream boundary, tidally driven water level variations were specified, based on observed stage data collected over one year at the site. At the bottom of the model, no-flow boundary conditions were applied, as specified in the original numerical model description (Yabusaki et al., 2020). Initial condition for the flow modeling in the floodplain was established by cycling one year of water level boundary conditions in the hillslope groundwater and tidally-driven stream (30 minute resolution) to a repeating dynamic equilibrium in the floodplain. This hydrologic boundary condition dataset was repeated for two model years to spin up the subsurface flow model to a dynamic steady state of a repeating annual cycle of water levels, porewater saturation, and total water mass in the model domain. After spinning up the model to an initial flow condition, salinity transport was coupled to the flow simulation.

For the transport, the salt concentration in the floodplain was set to 0, reflecting a freshwater system prior to exposure with tidal river water. At the upland boundary, a freshwater concentration is prescribed, $c(\mathbf{x}, t) = 0$. A time-dependent Dirichlet boundary condition, $c(\mathbf{x}, t) = c_{\mathrm{t}}(t)$, is applied at the top boundary, where $c_{\mathrm{t}}(t)$ represents a prescribed periodic function that varies with time, reflecting cyclic behavior over the simulation period to reflect the salinity in the inundated stream. Boundary conditions of this system are illustrated in Figure 1. The Richards equation governing variably saturated flow was solved using a Newton's nonlinear solver with specific parameters, including an infinity norm tolerance of 1 Pa, a maximum of 20 iterations, and a time step of 15 minutes. Subsurface saline transport was simulated using a global implicit scheme. Simulated floodplain inflows and outflows were continuously tracked to examine the water mass balance in the floodplain. The simulation was carried out on Stanford University's Sherlock High Performance Cluster, allocating 16 CPU cores for a runtime of 14 hours for 50 years of simulation.

2.1.3 Dynamic Mode Decomposition (DMD)

The initial step of DMD organizes data (pressure and salinity) extracted from the numerical model. For the 2D computational domain discretized with an N_x by N_z mesh, and for M time steps, these data are arranged into long column 1D vectors,

$$\mathbf{c}_k = \begin{bmatrix} c(x_1, z_1, t_k) \\ c(x_1, z_2, t_k) \\ \vdots \\ c(x_1, z_{N_z}, t_k) \\ c(x_2, z_1, t_k) \\ \vdots \\ c(x_{N_x}, z_{N_z}, t_k) \end{bmatrix}, \qquad k = 1, \ldots, M. \tag{2}$$

The vectors $\mathbf{c}_k \equiv \mathbf{c}(t_k) \in \mathbb{R}^N$, where $N = N_x \cdot N_z$ is the total number of elements in the numerical mesh, are stacked together to form two sets of data matrices in $\mathbb{R}^{N\times(M-1)}$: the original snapshots $\mathbf{C}$ and the shifted-in-time snapshots $\mathbf{C}'$,

$$\mathbf{C} = \begin{bmatrix} | & | & & | \\ \mathbf{c}_1 & \mathbf{c}_2 & \ldots & \mathbf{c}_{M-1} \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \mathbf{C}' = \begin{bmatrix} | & | & & | \\ \mathbf{c}_2 & \mathbf{c}_3 & \ldots & \mathbf{c}_M \\ | & | & & | \end{bmatrix}. \tag{3}$$

The DMD algorithm finds a linear operator matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ that minimizes the Frobenius norm (Kutz et al., 2016),

$$\|\mathbf{C}' - \mathbf{AC}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M} |\mathbf{C}'_{ij} - (\mathbf{AC})_{ij}|^2},$$

which quantifies the overall discrepancy between the actual system dynamics (represented by $\mathbf{C}'$) and the dynamics predicted by the operator $\mathbf{A}$. The algorithm advances the system by one time step as

$$\mathbf{c}_{k+1} = \mathbf{A}\mathbf{c}_k \quad \text{or} \quad \mathbf{C}' = \mathbf{AC}, \qquad k = 1, \ldots, M-1. \tag{4}$$

For DMD, $\mathbf{A}$ is computed as

$$\mathbf{A} = \mathbf{C}'\mathbf{C}^{\dagger} \qquad \Rightarrow \qquad \mathbf{A} = \mathbf{C}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^{*}, \tag{5}$$

where † represents the Moore-Penrose pseudo-inverse. The pseudo-inverse $\mathbf{C}^\dagger$ is computed as

$$\mathbf{C}^\dagger = \tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^*.$$

The matrices $\tilde{\mathbf{U}} \in \mathbb{R}^{N\times r}$ and $\tilde{\mathbf{V}}^* \in \mathbb{R}^{r\times(M-1)}$ are orthonormal; and $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{r\times r}$ is a diagonal matrix containing the singular values σ_i, derived from the singular value decomposition (SVD) of $\mathbf{C}$ for a chosen rank truncation r (see Algorithm 1). The singular values in $\tilde{\mathbf{\Sigma}}$ quantify the energy or importance of each mode and are essential for reducing the rank of the data. Only the top r singular values, corresponding to the most dominant modes, are retained during truncation to balance computational efficiency with model accuracy. The reduced rank approximations $\tilde{\mathbf{U}}$, $\tilde{\mathbf{\Sigma}}$, and $\tilde{\mathbf{V}}$ correspond to these top r singular values. The pseudo-inverse is estimated using SVD of a given (rectangular) matrix and approximated using a reduced rank approximation, where rank truncation, r, is the rank of the matrix (i.e., matrix $\mathbf{C}$) unless otherwise specified. Once the mapping matrix $\mathbf{A}$ is constructed, future state solutions are constructed via Eq. (4) (Step 5 of Algorithm 1).

```
Algorithm 1: Dynamic Mode Decomposition (DMD)
    Step 1: Compute SVD of $\mathbf{C}$
        $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}^* \leftarrow \mathrm{svd}(\mathbf{C})$
    Step 2: Truncate to rank $r$ or tolerance $\xi$
        $\tilde{\mathbf{U}} \leftarrow \mathbf{U}(:, 1:r)$, $\tilde{\mathbf{\Sigma}} \leftarrow \mathbf{\Sigma}(1:r, 1:r)$, $\tilde{\mathbf{V}}^* \leftarrow \mathbf{V}^*(1:r, :)$
    Step 3: Estimate the pseudo-inverse $\mathbf{C}^\dagger$
        $\mathbf{C}^\dagger \leftarrow \tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^*$
    Step 4: Compute the mapping matrix $\mathbf{A}$
        $\mathbf{A} \leftarrow \mathbf{C}'\mathbf{C}^\dagger$
    Step 5: Reconstruct snapshots
        $\mathbf{c}_{k+1} \leftarrow \mathbf{A}\mathbf{c}_k$
```

2.1.4 Extended Dynamic Mode Decomposition (XDMD)

Extended DMD (xDMD) combines approaches from generalized and residual DMDs. The generalized DMD (gDMD) introduces enhancements to the time advancement process and incorporates a learning bias term, represented by the vector **b**, to address inhomogeneity in both PDEs and boundary conditions. The gDMD formulation is found as

$$\mathbf{c}_{k+1} = \mathbf{A}_g\mathbf{c}_k + \mathbf{b}, \tag{6}$$

where $\mathbf{A}_g \in \mathbb{R}^{N\times N}$ and $\mathbf{b} \in \mathbb{R}^N$ are estimated after extending matrix $\mathbf{C}$ to a matrix $\widetilde{\mathbf{C}} \in \mathbb{R}^{(N+1)\times(M-1)}$ by adding an extra row vector $1 \in \mathbb{R}^{M-1}$ at the end of the matrix, such that

$$\widetilde{\mathbf{C}} = \begin{bmatrix}\mathbf{C}\\ \mathbf{1}\end{bmatrix}, \qquad [\mathbf{A}_g, \mathbf{b}] = \mathbf{C}'\widetilde{\mathbf{C}}^\dagger. \tag{7}$$

By subtracting the identity matrix from the original matrix $\mathbf{A}$, an approximation to an effective increment, commonly referred to as the residual increment in the machine learning community, is obtained.

Algorithm 2: Extended Dynamic Mode Decomposition (xDMD)

```
Step 1: Extend $\mathbf{C}$ to $\widetilde{\mathbf{C}}$
    $\widetilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} \\ 1 \end{bmatrix}$
Step 2: Compute SVD of $\widetilde{\mathbf{C}}$
    $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}^* \leftarrow \mathrm{svd}(\widetilde{\mathbf{C}})$
Step 3: Truncate to rank $r$ or tolerance $\xi$
    $\tilde{\mathbf{U}} \leftarrow \mathbf{U}(:, 1:r), \tilde{\mathbf{\Sigma}} \leftarrow \mathbf{\Sigma}(1:r, 1:r), \tilde{\mathbf{V}}^* \leftarrow \mathbf{V}^*(1:r, :)$
Step 4: Estimate the pseudo-inverse $\widetilde{\mathbf{C}}^\dagger$
    $\widetilde{\mathbf{C}}^\dagger \leftarrow \tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^*$
Step 5: Compute the mapping matrix $\mathbf{A}_x$ and the bias vector $\mathbf{b}$
    $[\mathbf{A}_x, \mathbf{b}] \leftarrow (\mathbf{C}' - \mathbf{C})\widetilde{\mathbf{C}}^\dagger$
Step 6: Reconstruct snapshots
    $\mathbf{c}_{k+1} \leftarrow \mathbf{c}_k + \mathbf{A}_x \mathbf{c}_k + \mathbf{b}$
```

The residual DMD (rDMD) involves the computation of the remainder matrix $\mathbf{A}_r \in \mathbb{R}^{N \times N}$ from the decomposition $\mathbf{A} = \mathbf{A}_r + \mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the identity matrix. The derivation of $\mathbf{A}_r$ is obtained from

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \mathbf{A}_r \mathbf{c}_k, \tag{8}$$

so that

$$\mathbf{A}_r = (\mathbf{C}' - \mathbf{C})\mathbf{C}^\dagger. \tag{9}$$

Algorithm 2 implements xDMD by combining gDMD and rDMD such that the time advance is

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \mathbf{A}_x \mathbf{c}_k + \mathbf{b}, \tag{10}$$

and $\mathbf{A}_x$ and $\mathbf{b}$ are computed as

$$[\mathbf{A}_x, \mathbf{b}] = (\mathbf{C}' - \mathbf{C})\widetilde{\mathbf{C}}^\dagger. \tag{11}$$

The four DMD variants are summarized in Table 3. We utilize xDMD due to its demonstrated superior performance in previous studies (Libero et al., 2024; Lu and Tartakovsky, 2021).

TABLE 3: Summary of DMD models

Models	Modification	Mapping Matrix	Reconstruction
DMD	-	$\mathbf{A} = \mathbf{C}'\mathbf{C}^\dagger$	$\mathbf{c}_{k+1} = \mathbf{A}\mathbf{c}_k$
rDMD	$\mathbf{A}_r = \mathbf{A} - \mathbf{I}$	$\mathbf{A}_r = (\mathbf{C}' - \mathbf{C})\mathbf{C}^\dagger$	$\mathbf{c}_{k+1} = \mathbf{A}_r\mathbf{c}_k + \mathbf{c}_k$
gDMD	$\widetilde{\mathbf{C}} = [\mathbf{C}, 1]^\top$	$[\mathbf{A}_g, \mathbf{b}] = \mathbf{C}'\widetilde{\mathbf{C}}^\dagger$	$\mathbf{c}_{k+1} = \mathbf{A}_g\mathbf{c}_k + \mathbf{b}$
xDMD	rDMD & gDMD	$[\mathbf{A}_x, \mathbf{b}] = (\mathbf{C}' - \mathbf{C})\widetilde{\mathbf{C}}^\dagger$	$\mathbf{c}_{k+1} = \mathbf{A}_x\mathbf{c}_k + \mathbf{c}_k + \mathbf{b}$

The xDMD surrogate represents system dynamics using a linear state-space equation with constant matrix $\mathbf{A}$ and bias vector $\mathbf{b}$. This formulation assumes that the dominant spatiotemporal modes of the system evolve approximately linearly over the chosen training window. For highly nonlinear systems with rapid transitions or strong spatial heterogeneities, the constant nature of $\mathbf{A}$ and $\mathbf{b}$ can introduce limitations. Furthermore, unlike Kalman filtering, which iteratively updates state estimates using time-variable gains based on observed data, xDMD is an “offline” method that relies on a fixed framework derived from past states. This approach provides computational efficiency and interpretability but lacks the flexibility of methods like the Kalman filter that dynamically condition on real-time data. This limits the ability of xDMD approaches to reflect possible changes in flow or solute boundary conditions that include hydrometeorological conditions at the soil-air interface and flooding scenarios. Nevertheless, there are adaptations of DMD that update DMD “on the fly” as new data is generated like online DMD (Zhang et al., 2019) and streaming DMD (Hemati et al., 2014) approaches. However, such methods are outside the scope of this study.

The DMD and xDMD frameworks yield parameter-free surrogate models designed for fast forward predictions under predefined conditions. While effective for stable dynamics, they do not explicitly account for evolving boundary conditions, such as variable hydrometeorological forcing or changes in flooding magnitude and frequency. Addressing such dynamic scenarios is outside the scope of this work. Future research will explore integrating DMD with parametric model order reduction techniques, such as DRIPS (Lu and Tartakovsky, 2023), to handle varying boundary conditions and external forcing. Such hybrid approaches could enable the surrogate model to interpolate across parameter spaces and dynamically adjust to new conditions, broadening its applicability for planning and management scenarios.

2.1.5 Accuracy Metric

We use the relative error,

$$\varepsilon_{\text{xDMD}} = \log_{10}\left(\frac{\|\mathbf{c}_k - \mathbf{c}_{k,\text{xDMD}}\|_2}{\|\mathbf{c}_k\|_2}\right), \tag{12}$$

to measure the accuracy of our xDMD surrogate in capturing pressure and salinity distributions. Here, $\mathbf{c}_{k,\text{xDMD}}$ represents the estimated states from the xDMD surrogate, and $\mathbf{c}_k$ corresponds to the true state obtained from PFLOTRAN data. This quantification of relative error is crucial for assessing the accuracy of xDMD compared to the ground truth data from PFLOTRAN.

Three tests are conducted: representation, interpolation, and extrapolation. Representation shows the the xDMD-surrogate’s ability to capture the dynamics and represent numerical output for salinity distribution across the entire 50-year simulation period. In the interpolation test, the xDMD surrogate is trained on weekly salinity snapshot data and tasked with reconstructing daily snapshots within the training domain by shifting the initial snapshot. In the extrapolation test, xDMD-surrogate is trained on eight-year salinity output and extrapolated for forty-two-year period for both pressure and salinity distributions.

2.2 Domain Decomposition

Domain decomposition is a family of methods used in numerical simulations and parallel computing to enhance the efficiency and convergence of numerical solvers of discretized partial differential equations (PDEs). It has also been used in supervised, unsupervised, and scientific machine learning algorithms to tackle computationally expensive problems, improve conditioning, enhance generalization, and accelerate performance (Klawonn et al., 2024). In the context of

DMD-based surrogates, domain decomposition improves robustness and efficiency. By partitioning the simulation domain into regions with distinct dynamics, domain decomposition reduces the complexity of surrogate modeling tasks, enabling DMD surrogates to focus on capturing dominant modes and coherent structures within each region. Drawing on advancements in multiscale modeling and scalable neural network frameworks (Moseley et al., 2023), this approach enhances predictive accuracy and computational efficiency. It also strengthens the robustness of DMD surrogates for long-term extrapolations, where variations in system behavior are more pronounced, offering a promising pathway for addressing complex, time-dependent systems.

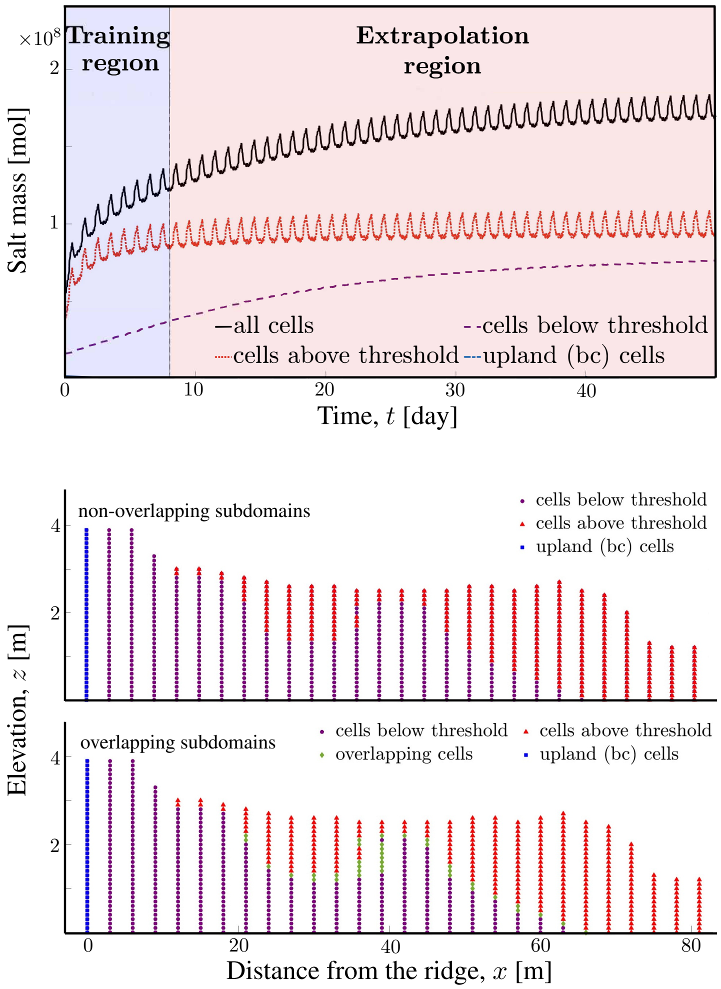

FIG. 2: Salinity dynamics predicted by the process-based simulations in PFLOTRAN. It exhibits two distinct behaviors when computed for the cells with salinity below and above threshold of 1 ppt (**top**). These cells form overlapping and non-overlapping subdomains (**bottom**) in alternative implementations of domain decomposition.

Figure 1 represents the simulation domain with all active cells from the PFLOTRAN simulations, hereafter referred to as “All Cells”. The Upland Boundary cells are enforced as non-contaminated freshwater values (zero-salinity). The cells close to the top boundary exhibit dynamics distinct from other cells due to differential salt accumulation rates (Fig. 2). This variation motivates the use of domain decomposition as a strategy to segment the domain into regions with distinct dynamics to improve surrogate accuracy and reduce errors caused by extrapolation across vastly different behaviors. Active cells are split into three non-overlapping regions: “Below Threshold”, “Above Threshold”, and “Inactive Cells”. These regions are defined based on the results of the initial simulation phase to identify areas of differing salinity dynamics. This partitioning is utilized solely for computational efficiency in the simulation setup and does not limit the applicability of the DMD procedure, which analyzes the global system behavior across the entire domain, independent of this decomposition. The DMD methodology remains applicable to both simulated and field data as it synthesizes a holistic representation of the system dynamics. The threshold is defined as 3.5 PSU (Practical Salinity Units), a measure of seawater salinity based on conductivity that is approximately equivalent to 1 part per thousand (ppt) (UNESCO, 1981). This value was determined based on the floodplain geometry and the salinity accumulation/dynamics observed within the first eight years of the simulation (Fig. 2). The threshold is determined empirically to ensure clear separation of dynamics, particularly in areas where salt accumulation dominates. This segmentation ensures that the surrogate can focus on coherent structures in each region, improving accuracy in the extrapolation region. The initial eight years of simulation took 0.5 hours of computational time, while the simulation of the remaining 42 years took 13.5 hours.

To demonstrate the robustness of the domain decomposition method and its insensitivity to the threshold choice, overlapping domain experiment was conducted. In the overlapping domain experiment, the threshold for cell separation is lowered for the Cells Above Threshold so that 10% of these cells overlap with the Cells Below Threshold in Fig. 2). The resulting concentrations in these overlapping cells are averaged to construct the salinity distributions. Our results reported below demonstrate that this strategy removes nonphysical negative salinity values generated by the xDMD surrogate in the extrapolation regime.

Figure 3 illustrates the proposed domain decomposition scheme for the xDMD surrogate. The numerical model output is partitioned into three distinct categories, as previously outlined. Notably, xDMD surrogates are separately trained for cells above and below the threshold. Following training, these surrogates undergo both reconstruction and extrapolation processes. Given that upland boundary cells maintain constant freshwater concentrations throughout the simulation, their values remain unaltered during extrapolation. Subsequently, the data from these surrogates is amalgamated into a unified output, which is then juxtaposed with the numerical model output for comparative analysis.

2.3 Data Sampling and Pre-Processing

From the PFLOTRAN simulation pressure and salinity distribution outputs were sampled on a daily basis ($\Delta t = 1$ day) over a fifty-years of simulation period ($T = 50$ years). For the salinity in the numerical model output below 10^{-2} Practical Salinity Unit (PSU), were adjusted by 10^{-2} PSU, since values below 10^{-2} PSU are indicative of freshwater (Yabusaki et al., 2020). This adjustment is necessary because freshwater in the PFLOTRAN simulations is defined and handled as having salinity near 10^{-10} PSU. For DMD purposes, values below 1×10^{-10} PSU are effectively set to 0.01 PSU to avoid numerical instabilities and ensure consistency in the

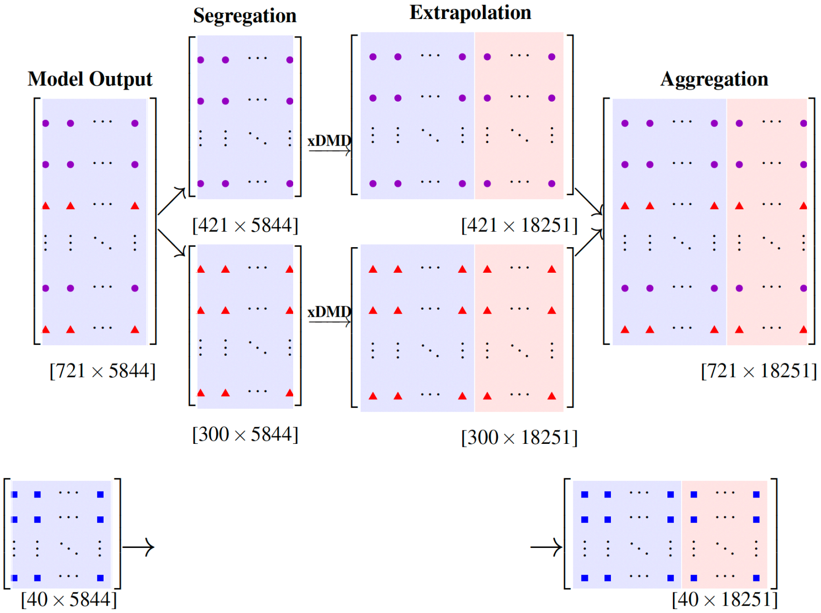

FIG. 3: Domain decomposition for xDMD. Numerical model output is segregated into three categories: upland boundary cells (circles), cells with below-threshold salinity (squares), and cells with above-threshold salinity (triangles).

reduced-order modeling process. 0.01 PSU is equivalent to 0.01 ppt or 10 ppm, classifying it as freshwater. This threshold simplifies the handling of very small salinity values while maintaining meaningful distinctions for the analysis. Furthermore, the pressure values (in kPa) obtained from the PFLOTRAN simulation were normalized to the interval [0, 1]. In the machine learning community, this process is commonly referred to as normalization, and it ensures that all features contribute equally to the model. In addition, these steps were taken to enhance numerical stability and mitigate the propagation of floating-point errors associated with small values of salinity and large values of pressure during the training of the xDMD surrogate.

3. RESULTS AND DISCUSSION

3.1 Representation

Representation refers to the xDMD-surrogate's ability to accurately capture and reproduce the dynamics of the system, matching the numerical output for salinity distribution across the entire 50-year simulation period. The figures below provide a comprehensive overview of the representation experiment (Fig. 4). The xDMD surrogate captures both salinity distribution and total salt mass dynamics from the physics-based numerical model with remarkable accuracy. Employing a singular value tolerance of 10^{-3} resulted in a truncation rank of 333, effectively discarding singular values smaller than 10^{-3} to simplify the model while retaining crucial features. Remarkably, only the mapping matrix $\mathbf{A}_x$, residual vector $\mathbf{b}$, and the initial snapshot vector $\mathbf{c}_0$, which serves as the starting point for reconstructing the system dynamics, need to be stored for the xDMD surrogate model. This streamlined approach necessitates a mere 25 times less storage space compared to the full physics-based PFLOTRAN model. Consequently, the xDMD surrogate emerges as a pivotal tool for significantly reducing storage costs associated with physics-based models, particularly during uncertainty quantification and parameter estimation in both forward and inverse modeling representations.

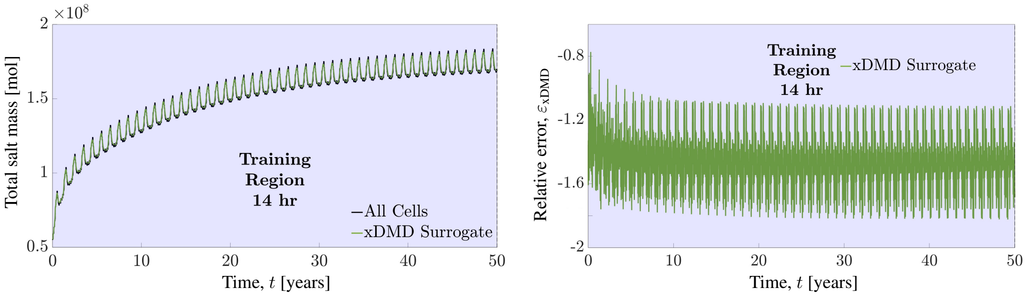

FIG. 4: DMD predictions of total salt mass (**left**) and relative error in prediction of salinity distribution, $\varepsilon_{\mathrm{xDMD}}$ in (12), (**right**) in the representation regime.

3.2 Interpolation

The xDMD surrogate was trained on a set of weekly snapshot data (e.g., $1, 8, 15, \ldots, 18249$). By shifting the initial snapshot by one or several days, sequences such as $3, 10, 17, \ldots, 18245$ were generated, enabling the reconstruction of all missing snapshots within the training region. Figure 5 exhibits the xDMD surrogate's performance on shifting the initial snapshot from the first to the third or fourth day, representing time points furthest from the training regime. Employing a singular value tolerance of 10^{-3} yielded a truncation rank of 298. The xDMD surrogate achieved comprehensive and highly accurate reconstruction of all missing snapshots within the designated training region. Furthermore, interpolation experiments demonstrate that the xDMD surrogate can significantly reduce storage requirements for numerical model output by over 2000 times. These results underscore the xDMD surrogate's robustness in navigating temporal gaps, thus indicating its potential for precise data interpolation across varying timeframes.

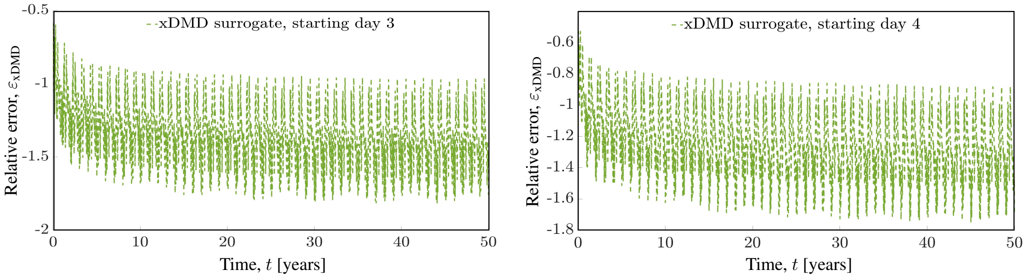

FIG. 5: Interpolation error of xDMD predictions of salinity distribution, ϵ_{xDMD} in (12), with the initial snapshot available at day 3 (**left**) or day 4 (**right**).

3.3 Extrapolation

3.3.1 Pressure

The matrix of fluid pressure has a rank of 239, whereas the matrix of salinity exhibits a considerably higher rank of 664. This discrepancy arises because salinity accumulates over time, whereas pressure remains non-cumulative and displays periodic behavior. The rapid decay of singular values (Fig. 6) underscores the low-rank nature of the pressure matrix. By applying a singular value tolerance of 10^{-3}, we achieve a truncation rank of 97. This reduced-rank approximation via the xDMD surrogate demonstrates its ability to capture the pressure dynamics, enabling accurate reconstruction of the pressure solution in previously unseen extrapolation regions with minimal error (Fig. 6). Such precise extrapolation is crucial for reliable modeling and prediction in complex systems such as seawater intrusion where pressure dynamics plays a pivotal role.

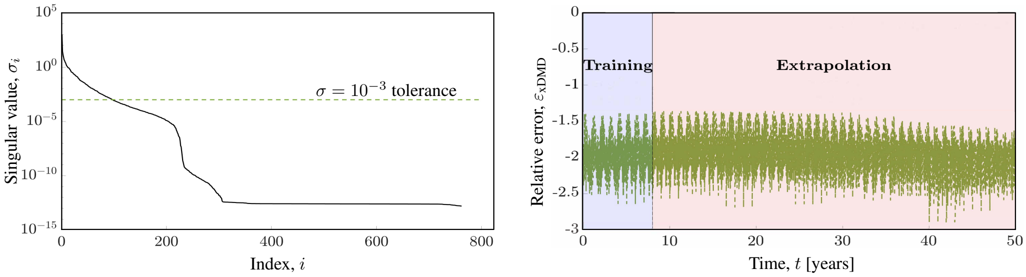

FIG. 6: xDMD prediction of pressure distribution in the extrapolation regime. Rapid decay of singular values σ_i (**left**) enables one to approximate the fluid-pressure matrix of rank 239 with its counterpart of rank 97, if the singular value tolerance is set to 10^{-3}. Relative error, $\varepsilon_{\mathrm{xDMD}}$ in (12), remains stable in the extrapolation regime over a long time horizon (**right**).

3.3.2 Salinity (without Domain Decomposition)

The extrapolation tests for salinity distributions across all cells, without employing domain decomposition, reveal that the salinity distribution snapshots exhibit relatively small errors in the extrapolated region, albeit the error increases with time (Fig. 7). However, a concerning increase in the number of cells with negative salinity is observed in the extrapolation regime (Fig. 7). This issue highlights a fundamental challenge: long-term extrapolation is inherently difficult to capture with DMD surrogates. The observed surge in negative salinity cells leads to a significant disruption in the salt mass dynamics within the system. Several factors contribute to this phenomenon. First, the presence of zero-salinity cells, enforced by the Upland Boundary Condition, distorts the overall distribution. Second, the dynamics differ markedly between cells that are above and below certain thresholds, adding complexity to the modeling process. Third, the time-dependent Top Boundary Condition introduces seasonal variations that further complicate the salt dynamics. Collectively, these factors present substantial challenges for the DMD-surrogate in accurately capturing and predicting the underlying dynamics of salinity transport.

These observations demonstrate that in certain extrapolation regimes, DMD surrogates may produce nonphysical results, such as negative salinity. Importantly, rather than a discouragement, this finding represents a critical insight into the limitations of DMD surrogates and the need for further refinement. This study is among the first to showcase these limitations and to propose domain decomposition as an innovative solution.

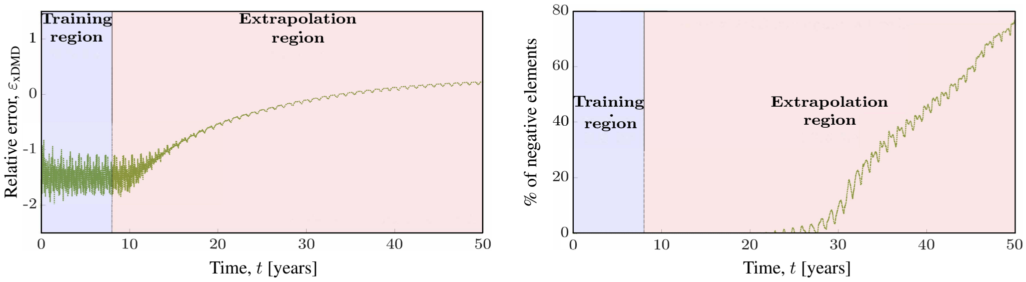

FIG. 7: xDMD predictions of salinity distribution in the extrapolation regime. Relative error, $\varepsilon_{\text{xDMD}}$ in (12), increases rapidly with time (**left**), as does the fraction of the elements at which xDMD predicts unphysical negative salinity (**right**).

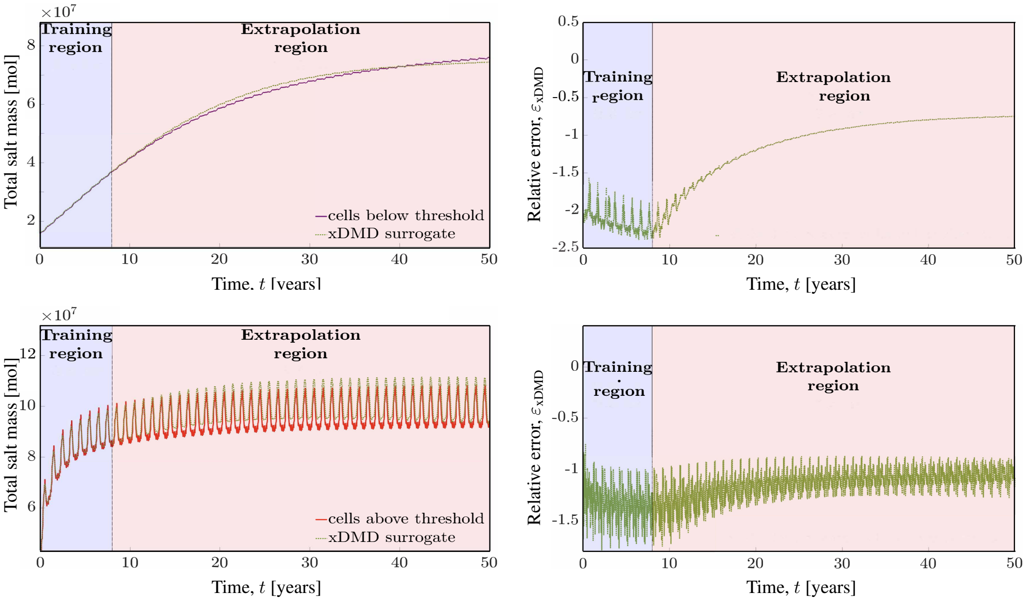

FIG. 8: Non-overlapping domain decomposition strategy for xDMD predictions of salinity distribution in the extrapolation regime. **Left column:** Salt accumulation, predicted with the process-based model and its xDMD surrogate, in the subdomains composed of the “below threshold” (**top**) and “above threshold” (**bottom**) cells. **Right column:** In both subdomains, relative error, $\varepsilon_{\text{xDMD}}$ in (12), is an order of magnitude smaller than that of the xDMD surrogate without domain decomposition (Fig. 7).

3.3.3 Salinity (Non-Overlapping Domain Decomposition)

By introducing domain decomposition, we mitigate the occurrence of unphysical salinity values in the extrapolation region. In the non-overlapping domains, negative salinity is no longer observed in both below and above threshold cells. Additionally, the snapshot-to-snapshot salinity distribution and the overall salt dynamics within the domain are accurately captured (Fig. 8). The extrapolation error for xDMD in the above-threshold cells is more stable than in the below-threshold cells, demonstrating that xDMD performs better in regions with more dynamics. This improvement underscores the effectiveness of domain decomposition in enhancing the accuracy and reliability of salinity extrapolation, providing a robust solution to the challenges previously encountered in the single domain approach.

3.3.4 Salinity (Overlapping Domain Decomposition)

In the overlapping-domains approach, we lowered the threshold for the above-threshold cells, while keeping the cells below threshold unchanged. Figure 9 shows that this adjustment—comparing 300 cells in the non-overlapping domain to 336 cells in the overlapping domain—has only a minor effect on the system. Both the salinity distribution and salt accumulation are accurately maintained despite the threshold change. This indicates that the overlapping domains method is robust, preserving the integrity of salinity dynamics and ensuring reliable extrapolation results even with slight variations in threshold parameters.

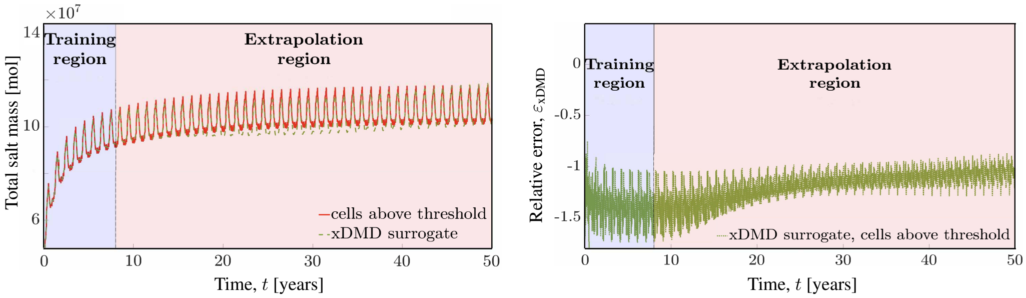

FIG. 9: Overlapping domain decomposition strategy for xDMD predictions of salinity distribution in the extrapolation regime, for cells above threshold. **Left:** Salt accumulation in the above-threshold subdomain, predicted with the process-based model and its xDMD surrogate. **Right:** Relative error, $\varepsilon_{\text{xDMD}}$ in (12), is an order of magnitude smaller than that of the xDMD surrogate without domain decomposition (Fig. 7), and is similar to that of the xDMD surrogate with non-overlapping domain decomposition (Fig. 8).

The computational benefits of xDMD surrogates for both overlapping and non-overlapping domains are highlighted in Table 4. The xDMD surrogate offers a significant computational speedup, being 1780 times faster in simulation run time compared to the physics-based (PFLOTRAN) model. Additionally, it provides an element storage efficiency advantage of 891 times. The Frobenius norm, which measures the approximation error, is slightly lower for the overlapping domain compared to the non-overlapping domain, indicating better accuracy in the overlapping configuration which complements findings in the cumulative sum plot for domain decomposition.

3.4 xDMD Surrogate Eigenvalue Analysis

Both DMD and xDMD provide interpretability by associating eigenvalues with the dynamics of the system, offering insights into the temporal evolution of dominant modes. Eigenvalue analysis assesses the stability and dynamic behavior of an xDMD surrogate, particularly in differentiating between regions with varying levels of activity.

TABLE 4: Comparison of two domain-decomposition strategies for the construction of xDMD surrogates of the physics-based numerical model (NM).

Model	Run time [hr]	Storage cost [elm]	$\|\mathbf{C}_{\text{NM}} - \mathbf{C}_{\text{xDMD}}\|_{\text{F}}$
Physics-based model	13.5	$2.24 \cdot 10^9$	–
xDMD non-overlapping	$7.58 \cdot 10^{-3}$	$2.49 \cdot 10^6$	$1.76 \cdot 10^3$
xDMD overlapping	$8.47 \cdot 10^{-3}$	$2.51 \cdot 10^6$	$1.65 \cdot 10^3$

In comparison to the standard DMD surrogate, unit circle and eigenvalues of xDMD surrogate are shifted left (Fig. 10). This is due the fact that extended DMD matrix, $\mathbf{A}_x$, is shifted by the decomposition $\mathbf{A}_x = \mathbf{A} - \mathbf{I}$: $\det(\mathbf{A_x} - \boldsymbol{\lambda} \times \mathbf{I}) = \det(\mathbf{A} - (\boldsymbol{\lambda} + 1) \times \mathbf{I}) = 0$ (Section 2.1.4). Eigenvalues clustered near the origin represent slow dynamics with negligible activity, while eigenvalues spread farther from the origin indicate faster, more dynamic behavior. In the xDMD surrogate, eigenvalues of cells below the threshold are tightly clustered, indicating regions with slower dynamics and less temporal variability. Conversely, eigenvalues of cells above the threshold are more spread out, representing regions with higher activity and greater dynamic variability. Figure 10 illustrates these differences in the extrapolation regime by comparing the overall eigenvalue spectrum of the xDMD surrogate to the eigenvalue spectra for cells below and above the threshold. This analysis highlights the ability of xDMD to capture and differentiate dynamic regions within the domain, providing interpretability into how different regions contribute to the system's overall behavior.

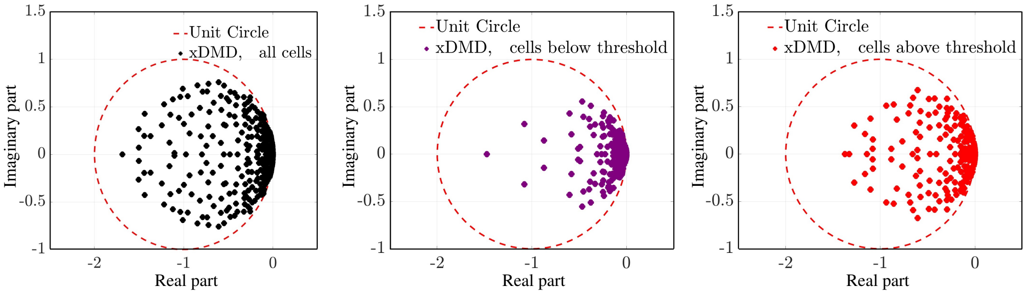

FIG. 10: Eigenvalue analysis of the xDMD surrogate in the extrapolation regime.

4. CONCLUSION

We demonstrated xDMD to be a cheap, efficient, and robust surrogate of process-based models of coastal groundwater systems, addressing the pressing challenges posed by rising sea levels and intensified storm surges. By building on the validated PFLOTRAN model for the Beaver Creek site in Washington, we examined xDMD's ability to overcome the computational and operational limitations of traditional process-based models. Our findings demonstrate that xDMD surrogates can accurately reproduce salinity distributions and total salt mass dynamics.

We found xDMD to be robust in handling problems with repeated and low rank dynamics, such as extrapolating pressure distribution across both temporal and spatial domains, with reliable accuracy. However, xDMD performed poorly with long-term salinity extrapolation where it accumulates, yielding nonphysical results such as negative concentrations. To address this issue, we implemented domain decomposition, which successfully improved xDMD's accuracy, stability and robustness. Given xDMD interpretability, we effectively differentiated slower and faster dynamic regions through eigenvalue analysis. Our analysis revealed that the overlapping domain approach outperforms the non-overlapping method, offering superior stability and a better fit for learning salt accumulation in the floodplain.

In our experiments, xDMD significantly accelerated simulations (1780 times faster) and reduced storage requirements. The xDMD surrogate requires 800 times less storage for salinity and pressure modeling and over 2000 times less storage when reconstructing missing snapshots, making it highly efficient for long-term modeling, such as 42-year period. These features of xDMD make them a scalable and efficient alternative for real-world scenarios.

Although xDMD shows significant promise for modeling coastal groundwater systems, it is not well-suited for handling highly non-linear or long-term dynamic systems. Future work will focus on advancing domain decomposition techniques to incorporate dynamic boundary conditions and enhance the stability and reliability of xDMD surrogates for effective coastal groundwater management. Another direction is to explore surrogates that encompass a broader parameter space, which will complement the current xDMD surrogate and enhance its applicability for scenario development such as sea level rise and climate change.

ACKNOWLEDGMENT

This research was funded in part by the Strategic Environmental Research and Development Program (SERDP) of the Department of Defense under award RC22-3278, by the Air Force Office of Scientific Research under grant FA9550- 21-1-0381, by the National Science Foundation under award 2100927, and by the Office of Advanced Scientific Computing Research (ASCR) within the Department of Energy Office of Science under award number DE-SC0023163. The authors declare no real or perceived financial conflicts of interests.

REFERENCES

Aria, M., Cuccurullo, C., and Gnasso, A., A Comparison Among Interpretative Proposals for Random-Forests, *Machine Learning with Applications*, vol. **6**, p. 100094, 2021.

Azevedo de Almeida, B. and Mostafavi, A., Resilience of Infrastructure Systems to Sea-Level Rise in Coastal Areas: Impacts, Adaptation Measures, and Implementation Challenges, *Journal of Infrastructure Systems*, vol. **27**, no. 4, p. 04021045, 2021.

Batzle, M. and Wang, Z., Seismic Properties of Pore Fluids, *Geophysics*, vol. **57**, no. 11, pp. 1396–1408, 1992.

Bedient, P.B., Huber, W.C., and Vieux, B.E., *Hydrology and Floodplain Analysis*, 5th Edition, Pearson Education, 2013.

Carsel, R.F. and Parrish, R.S., Developing Joint Probability Distributions of Soil Water Retention Characteristics, *Water Resources Research*, vol. **24**, no. 5, pp. 755–769, 1988.

Hemati, M.S., Williams, M.O., and Rowley, C.W., Dynamic Mode Decomposition for Large and Streaming Datasets, *Physics of Fluids*, vol. **26**, p. 111701, 2014.

Hess, M.W., Quaini, A., and Rozza, G., A Data-Driven Surrogate Modeling Approach for Time-Dependent Incompressible Navier-Stokes Equations with Dynamic Mode Decomposition and Manifold Interpolation, *Advances in Computational Mathematics*, vol. **49**, no. 22, 2023.

Klawonn, A., Lanser, M., and Weber, J., Machine Learning and Domain Decomposition Methods - A Survey, *Computational Science and Engineering*, vol. **1**, no. 1, p. 2, 2024.

Kutz, J.N., Brunton, S.L., Brunton, B.W., and Proctor, J.L., *Dynamic Mode Decomposition*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016.

Libero, G., Chiofalo, A., Ciriello, V., and Tartakovsky, D.M., Extended Dynamic Mode Decomposition for Model Reduction in Fluid Dynamics Simulations, *Physics of Fluids*, vol. **36**, no. 6, p. 061911, 2024.

Lu, H. and Tartakovsky, D.M., Prediction Accuracy of Dynamic Mode Decomposition, *SIAM Journal on Scientific Computing*, vol. **42**, no. 3, pp. A1639–A1662, 2020.

Lu, H. and Tartakovsky, D.M., Extended Dynamic Mode Decomposition for Inhomogeneous Problems, *Journal of Computational Physics*, vol. **444**, p. 110550, 2021.

Lu, H. and Tartakovsky, D.M., DRIPS: A Framework for Dimension Reduction and Interpolation in Parameter Space, *Journal of Computational Physics*, vol. **493**, p. 112455, 2023.

Lucia, D.J., Beran, P.S., and Silva, W.A., Reduced-Order Modeling: New Approaches for Computational Physics, *Progress in Aerospace Sciences*, vol. **40**, no. 1-2, pp. 51–117, 2004.

Moseley, B., Markham, A., and Nissen-Meyer, T., Finite Basis Physics-Informed Neural Networks (FBPINNS): A Scalable Domain Decomposition Approach for Solving Differential Equations, *Advances in Computational Mathematics*, vol. **49**, p. 62, 2023.

PFLOTRAN Development Team, 2024. PFLOTRAN: User Manual. Version 5.0, www.documentation.pflotran.org.

UNESCO, *Background Papers and Supporting Data on the Practical Salinity Scale 1978*, UNESCO Technical Papers in Marine Science No. 37, 1981.

Yabusaki, S., Myers-Pigg, A., Ward, N., Waichler, S., Sengupta, A., Hou, Z., Chen, X., Fang, Y., Duan, Z., Serkowski, J., Indivero, J., Moore, C., and Gunn, C., Floodplain Inundation and Salinization from a Recently Restored First-Order Tidal Stream, *Water Resources Research*, vol. **56**, no. 7, p. e2019WR026850, 2020.

Zhang, H., Rowley, C.W., Deem, E.A., and Cattafesta, L.N., Online Dynamic Mode Decomposition for Time-Varying Systems, *SIAM Journal on Applied Dynamical Systems*, vol. **18**, no. 3, pp. 1586–1609, 2019.

Zhou, Z. and Tartakovsky, D.M., Markov Chain MonteCarlo with Neural Network Surrogates: Application to Contaminant Source Identification, *Stochastic Environmental Research and Risk Assessment*, vol. **35**, no. 3, pp. 639–651, 2021.

Zhou, Z., Zabaras, N., and Tartakovsky, D.M., Deep Learning for Simultaneous Inference of Hydraulic and Transport Properties, *Water Resources Research*, vol. **58**, no. 10, p. e2021WR031438, 2022.