Adjoint model based deep learning for efficient pointwise residence time estimation in coastal environments

Ismail Oubarka[a,b,*], Imad Kissami[a], Eric Deleersnijder[c]

[a]*Mohammed VI Polytechnic University, College of Computing, Lot 660, Ben Guerir, 43150, Morocco*
[b]*Sorbonne Paris Nord University, LAGA, Villetaneuse, 93430, France*
[c]*Université catholique de Louvain, IMMC and ELI, B-1348, Louvain-la-Neuve, 1348, Belgium*

Abstract

Residence time is an essential quantitative diagnosis for evaluating water renewal and pollutant retention in coastal environments. However, its computation remains challenging due to the spatial and temporal variability of the coastal flows. Traditional techniques are usually based on simplifying hypotheses that fail to capture the dynamic conditions accurately. We introduce a new deep learning framework that uses a convolutional Long Short-Term Memory network (ConvLSTM) to solve the adjoint advection-diffusion problem that leads to spatially and temporally resolved residence time without the necessity to store extensive hydrodynamic histories. Our approach utilizes high-resolution data generated by solving the shallow water equations using a non-homogeneous Riemann solver (SRNH). We use a data reduction approach to address the substantial computing difficulties posed by full-resolution data. Initially, the ConvLSTM is trained using full data for the initial waiting period of the simulation to establish robust recognition of hydrodynamic models. After that, temporal and spatial resolutions are progressively reduced to preserve the essential characteristics of the model. As a result, the computational cost has been reduced from 48 hours using the traditional data storage method to 3 hours using the ConvLSTM-trained model. Additionally, the memory required to store the hydrodynamics data has been reduced from 14 TB to 800 MB, demonstrating the efficiency of the proposed method in both computational and memory usage.

Keywords: Shallow water equations,, Pointwise, Residence time, Adjoint model, SRNH, ConvLSTM, Coastal environments,

1. Introduction

Residence time is a key concept in hydrodynamics and environmental studies, representing the average time water or dissolved substances stay within a specific area before leaving it. This metric is essential for understanding processes such as water renewal, pollutant dispersion, and ecological health in aquatic environments [1, 2, 3]. Accurate calculation of residence time is crucial for managing coastal and semi-enclosed water systems, where human activities and natural dynamics interact closely [4]. However, traditional methods for calculating residence time often rely on oversimplifications or require computationally expensive simulations, making them unsuitable for capturing the complexity of real-world systems [5, 6].

Conventional approaches, such as tracer-based or hydrodynamic models, typically assume stationarity, perfect mixing, or neglect diffusion processes [7]. These simplifications fail to represent the variability and complexity of natural systems, where flow dynamics change over time and space. For example, water parcels released at different times or locations encounter different hydrodynamic conditions, resulting in widely varying residence times [8]. Computing residence time distributions for an entire domain usually involves forward simulations for each release point, which becomes computationally impractical in large-scale, time-dependent systems [5, 9, 10].

*Corresponding author
Email address: ismail.oubarka@um6p.com (Ismail Oubarka)

A further challenge arises in addressing complex, three-dimensional flows influenced by tides, wind, and seasonal changes [11, 12, 13]. These flows involve intricate interactions between advection, diffusion, and boundary conditions, which are difficult to handle with traditional methods. Additionally, existing approaches often struggle with interpreting boundary conditions, such as the re-entry of water parcels into the domain, and fail to distinguish between first-exit and final-exit times. Variable hydrodynamic conditions, driven by external factors like tides or wind, further complicate residence time computation by requiring dynamic solutions that adapt to changing environmental states.

Adjoint modeling has emerged as a powerful tool to address some of these challenges. By solving the advection-diffusion problem backward in time, adjoint methods eliminate the need for forward simulations at multiple release points, reducing computational costs while producing detailed spatial and temporal distributions of residence time [5]. However, these methods require the storage of extensive hydrodynamic data for backward computations, which can be infeasible for large and dynamic domains. To overcome these limitations, this study introduces a novel approach that integrates deep neural networks (DNNs) into the adjoint modeling framework. While recent studies have shown the effectiveness of deep learning–particularly convolutional and recurrent architectures–for modeling hydrodynamic behavior [14, 15, 16], these approaches are typically designed for forward predictions (e.g., water level, current fields, or particle tracking) and are not used for adjoint equations.

By contrast, our method is, to the best of our knowledge, the first to use DNNs to approximate the solution of the adjoint advection-diffusion equation for the residence time, enabling backward-in-time computation of residence time without storing complete hydrodynamic histories. This approach drastically reduces memory usage and computational cost, while retaining the capacity to model nonlinear and time-varying dynamics typical of coastal systems. The trained network generalizes well from sparse or compressed input sequences, making it a scalable and efficient solution for residence time estimation in complex and dynamic environments. The proposed framework is applied to the Nador Lagoon, Morocco, a nearly enclosed coastal system subject to strong tidal and seasonal forcing, to demonstrate its ability to reconstruct residence time distributions under realistic conditions. The results show an accurate representation of tidal dynamics, residual flows, and spatial variability, with significant gains in computational performance. This contribution advances the modeling of residence time toward more practical and adaptable tools for environmental management, pollutant transport analysis, and coastal sustainability.

The remainder of the paper is organized as follows. Section 2 presents the theoretical background of residence time in coastal environments, the adjoint modeling approach, and challenges associated with large-scale hydrodynamic simulations. Section 3 introduces the proposed deep learning framework, detailing the ConvLSTM-based model architecture, training methodology, and the adaptive data reduction strategy for hydrodynamic input compression. Section 4.2 describes the experimental setup and application of the method to the Nador Lagoon, including the data used, simulation configuration, and implementation details. Section 4.2 presents also the results, including model performance metrics, comparisons with traditional adjoint solutions, and analysis of spatial-temporal residence time patterns. Section 5 concludes the study with key findings and outlines directions for future work.

2. Mathematical models

2.1. Residence time model

In this study, we compute the residence time to characterize the transport processes and water renewal within a shallow coastal lagoon connected to the open sea via a single inlet, which serves simultaneously as the point of inflow and outflow. The residence time, denoted by $\theta_r(x, y, t)$, is defined as the duration that a parcel of water, initially located inside the lagoon, takes to exit the domain for the first time. This diagnostic timescale is especially relevant for evaluating the retention of water masses and associated tracers in semi-enclosed aquatic systems.

We adopt a depth-averaged, two-dimensional formulation of the residence time equation, consistent with the shallow water assumption. The governing equation is written as:

$$\frac{\partial(h\theta_r)}{\partial\tau} + \nabla\cdot(h(-\mathbf{u})\theta_r) = \nabla\cdot(h\kappa\nabla\theta_r) + h, \tag{1}$$

where $h(x, y, t)$ is the total water depth, $\mathbf{u}(x, y, t)$ is the depth-averaged horizontal velocity components, and κ is the horizontal diffusivity coefficient. The right-hand side includes a source term h, reflecting the continuous accumulation of residence time within the domain.

Since the equation is derived from the adjoint of the original water transport equation, it must be integrated backward in time, i.e., with $\tau = T - t$, starting from the final time T of the hydrodynamic simulation. The backward integration allows for computing the first-exit time of water parcels without tracking individual trajectories.

At the open boundary, where water exits the lagoon domain, we impose a homogeneous Dirichlet condition:

$$\theta_r = 0 \quad \text{on} \quad \Gamma_{\text{open}},$$

indicating that water parcels reaching this boundary are assumed to have completed their residence within the lagoon. Along the closed coastal boundaries, a no-flux Neumann condition is applied:

$$(\nabla\theta_r \cdot \mathbf{n}) = 0 \quad \text{on} \quad \Gamma_{\text{coast}},$$

where $\mathbf{n}$ is the outward normal vector. This reflects the impermeability of solid boundaries. The backward-time integration is initialized with a zero field $\theta_r(t = T) = 0$, and the simulation is continued over a spin-up period of sufficient length (typically twice the expected residence time) to ensure convergence of the steady-state solution. After this transient period, the residence time field becomes independent of the initial condition and accurately reflects the influence of the lagoon hydrodynamics and geometry on water exchange.

2.2. Hydrodynamics model

The hydrodynamic behavior of the domain is described by the two dimensional shallow water equations (SWE), which are derived by vertically integrating the incompressible Navier-Stokes equations under the assumption of hydrostatic pressure and negligible vertical acceleration. The resulting system accounts for gravitational forces, wind stress, bottom friction, Coriolis acceleration, and horizontal momentum diffusion. The model reads:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial (hu)}{\partial x} + \dfrac{\partial (hv)}{\partial y} = 0, \\ \dfrac{\partial (hu)}{\partial t} + \dfrac{\partial}{\partial x}\left(hu^2 + \dfrac{1}{2}gh^2\right) + \dfrac{\partial (huv)}{\partial y} = -gh\dfrac{\partial Z}{\partial x} + \dfrac{1}{\rho_w}(\tau_{wx} - \tau_{bx}) + fhv + \mathcal{D}_x, \\ \dfrac{\partial (hv)}{\partial t} + \dfrac{\partial (huv)}{\partial x} + \dfrac{\partial}{\partial y}\left(hv^2 + \dfrac{1}{2}gh^2\right) = -gh\dfrac{\partial Z}{\partial y} + \dfrac{1}{\rho_w}(\tau_{wy} - \tau_{by}) - fhu + \mathcal{D}_y, \end{cases} \tag{2}$$

where $h(x, y, t)$ is the water depth, $u(x, y, t)$ and $v(x, y, t)$ are depth-averaged horizontal velocity components, $Z(x, y)$ represents the bottom elevation, and g is the gravitational acceleration. The Coriolis parameter f is given by

$$f = 2\omega \sin(\phi), \tag{3}$$

where ω is the angular velocity of Earth and ϕ is the latitude. The wind stress components at the surface are computed using a quadratic drag law:

$$\tau_{wx} = \rho_a C_d w_x \sqrt{w_x^2 + w_y^2}, \qquad \tau_{wy} = \rho_a C_d w_y \sqrt{w_x^2 + w_y^2}, \tag{4}$$

where ρ_a is the air density, w_x and w_y are the wind velocity components at 10 meters above the surface, and C_d is the wind drag coefficient. The bottom shear stress is represented by:

$$\tau_{bx} = \rho_w C_b u \sqrt{u^2 + v^2}, \qquad \tau_{by} = \rho_w C_b v \sqrt{u^2 + v^2}, \tag{5}$$

where ρ_w is the water density and C_b is the bed friction coefficient, which may be parameterized by Manning's formula:

$$C_b = \frac{g}{C_z^2}, \qquad C_z = \frac{h^{1/6}}{n_m}, \tag{6}$$

with C_z as the Chezy coefficient and n_m the Manning roughness coefficient. The horizontal momentum diffusion terms $\mathcal{D}_x$ and $\mathcal{D}_y$ account for viscous effects and are derived from second-order expansions of the Navier-Stokes equations

[17]:

$$\mathcal{D}_x = 2\nu\frac{\partial}{\partial x}\left[h\left(2\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right]+2\nu\frac{\partial}{\partial y}\left(h\frac{\partial u}{\partial y}\right), \tag{7}$$

$$\mathcal{D}_y = 2\nu\frac{\partial}{\partial x}\left(h\frac{\partial v}{\partial x}\right)+2\nu\frac{\partial}{\partial y}\left[h\left(\frac{\partial u}{\partial x}+2\frac{\partial v}{\partial y}\right)\right], \tag{8}$$

where ν is the kinematic viscosity of water. This model captures the dominant dynamics of SWE under the influence of tides, wind, and topography. It is especially suitable for modeling coastal lagoons and semi-enclosed basins [18, 19, 20], where exchanges with the open sea and internal circulation govern ecological and physical processes.

3. ConvLSTM-based surrogate model for hydrodynamic prediction

3.1. Mathematical justification for using ConvLSTM in the adjoint model

Solving the residence time equation backward in time requires full knowledge of the hydrodynamic fields (u, v, h) over the entire simulation period. Traditionally, this involves storing the complete outputs from the forward simulation of the SWE, which becomes memory-intensive and computationally expensive, especially for large domains or long simulation times.

To address this, we replace the data storage step with a learned approximation of the hydrodynamic fields using a ConvLSTM network, which has been shown to effectively capture complex spatiotemporal dynamics in physical systems [14], making it suitable for learning time-resolved surrogate models of PDE-based simulations. The ConvLSTM model $\mathcal{F}_\theta$ is trained to reproduce the time evolution of the hydrodynamic variables. At each time step t, we approximate:

$$(u(t), v(t), h(t)) \approx \mathcal{F}_\theta(u(t-\Delta t), \ldots, u(t-k\Delta t); v(t-\Delta t), \ldots; h(t-\Delta t), \ldots), \tag{9}$$

where k is the length of the time window and θ represents the trained model parameters. On the first day of the simulation, k is fixed at $k = 1$. After that, k increases step by step as $k \in \{1, 2, \ldots, n\}$. The value of n depends on the duration of the numerical simulation and should not surpass one full day of the simulation. Instead of storing the entire hydrodynamic history (u, v, h), we reconstruct these fields during the backward integration of the adjoint Eq. (1) by querying $\mathcal{F}_\theta$ with previously generated values. The approximate backward-time solution of the adjoint model is then defined as:

$$\theta_r^{n+1} = \mathcal{A}(\theta_r^n; \mathcal{F}_\theta(\mathbf{H}^n)), \tag{10}$$

where θ_r^n is the solution of the adjoint equation at time t^n, $\mathcal{F}_\theta(\mathbf{H}^n)$ provides the reconstructed hydrodynamic inputs at t^n, and $\mathcal{A}$ is the numerical integration operator of the adjoint model.

This formulation eliminates the need to store (u, v, h) over the entire simulation window, while preserving the ability to solve the adjoint problem backward in time with sufficient accuracy. The model $\mathcal{F}_\theta$ is trained once using high-fidelity numerical simulation data and then used recursively during the backward integration.

3.2. Mathematical formulation of ConvLSTM

ConvLSTM extends the classical LSTM architecture by replacing fully connected layers with convolutional layers, which preserves spatial structure while modeling temporal dynamics [21, 22]. The update equations are given by:

$$\begin{cases} i_t = \sigma(W_i * X_t + U_i * H_{t-1} + b_i), \\ f_t = \sigma(W_f * X_t + U_f * H_{t-1} + b_f), \\ o_t = \sigma(W_o * X_t + U_o * H_{t-1} + b_o), \\ \tilde{C}_t = \tanh(W_C * X_t + U_C * H_{t-1} + b_C), \\ C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \\ H_t = o_t \odot \tanh(C_t). \end{cases} \tag{11}$$

where X_t is the input at time t, H_t is the hidden state, C_t is the cell state, and W, U are learnable convolutional kernels. The symbols $*$ and $\odot$ denote convolution and element-wise multiplication, respectively. Activation functions σ and tanh refer to sigmoid and hyperbolic tangent.

3.3. Architecture of the ConvLSTM model

The architecture of the ConvLSTM model employed in this study consists of multiple layers of convolutional LSTM units, followed by a final convolutional layer that maps the learned representations to the target hydrodynamic variables. The complete training pipeline is outlined in Algorithm 1. While $\tau = 0.3$, this value was selected to reduce the amount of data provided to the ConvLSTM while preserving accurate predictions in both space and time. Smaller values of τ result in the selection of a larger number of spatial locations, increasing the computational cost without a significant gain in accuracy, whereas larger values lead to insufficient sampling of dynamically important regions. The chosen value provides a good balance by focusing the training on areas with strong spatial and temporal variations, while using fewer samples in smoother regions.

Algorithm 1 ConvLSTM-Based Surrogate Model Training for SWE Solutions

Require: Numerical solution dataset of SWE variables $\{h_t, u_t, v_t\}_{t=1}^{T}$
Require: Spatial domain discretized into elements $\{T_i\}$, adaptive sampling threshold τ
Require: Sequence length k, model parameters θ
Ensure: Trained ConvLSTM surrogate that reproduces dynamics within time window $[1, T]$

Step 1: Adaptive Sampling for Data Reduction

for each spatial element T_i **do**
  Compute velocity gradient norm: $G_i = \|\nabla u(T_i)\|$
end for
Normalize: $C^n(T_i) = \frac{G_i}{\max_j G_j}$
Define reduced domain $\Omega_{\text{red}} \leftarrow \{T_i \mid C^n(T_i) \geq \tau\}$

Step 2: Input-Output Pair Construction from Simulation Data

for each timestep $t = 1$ to $T - k$ **do**
  Extract input sequence:

$$X_t = [h_t, u_t, v_t], \ldots, [h_{t+k-1}, u_{t+k-1}, v_{t+k-1}]$$

  Extract corresponding reference sequence (same as input): $Y_t = X_t$
  Apply spatial masking using Ω_{red} to both X_t and Y_t
  Store training pair (X_t, Y_t)
end for

Step 3: ConvLSTM Surrogate Model Training

while loss not converged **do**
  for each mini-batch $(X^{(b)}, Y^{(b)})$ **do**
    Forward pass: $\hat{Y}^{(b)} \leftarrow \text{ConvLSTM}_\theta(X^{(b)})$
    Compute reconstruction loss:

$$\mathcal{L} = \lambda_{Data}\mathcal{L}_{Data} + \lambda_{PDE}\mathcal{L}_{PDE}$$

    Update parameters θ using gradient descent
  end for
end while

Step 4: Surrogate Evaluation

Compare model output $\hat{Y}$ to ground truth Y

The model is trained using the mean squared error (MSE) loss function, which quantifies the difference between predicted and true hydrodynamic fields with the residual loss from the governing PDEs:

$$\mathcal{L}_{Data} = \frac{1}{N}\sum_{i=1}^{N}\left(h_i^{true} - h_i^{pred}\right)^2 + \left(u_i^{true} - u_i^{pred}\right)^2 + \left(v_i^{true} - v_i^{pred}\right)^2 \tag{12}$$

$$\mathcal{L}_{PDE} = \frac{1}{N}\sum_{i=1}^{N}\Big(\mathcal{R}_h^2(t_i, x_i, y_i) + \mathcal{R}_u^2(t_i, x_i, y_i) + \mathcal{R}_v^2(t_i, x_i, y_i)\Big), \tag{13}$$

where N denotes the number of spatial points in the training set, and the superscripts *true* and *pred* represent the ground truth and predicted values, respectively. The residuals $\mathcal{R}_h$, $\mathcal{R}_u$, and $\mathcal{R}_v$ correspond to the mass conservation equation and the momentum equations in the x- and y-directions of the SWEs see Eq. (2). To compute the spatial derivatives in $\mathcal{L}_{PDE}$, we use automatic differentiation [23], a built-in feature in the PyTorch library. This method calculates derivatives directly from the predicted outputs of thr ConvLSTM, so there's no need to use finite difference methods. respectively. The coefficients λ_{Data} and λ_{PDE} are weighting parameters that balance the relative contributions of the data-driven and physics-informed losses, and their respective values are 0.8 and 0.2. These values were chosen because the main objective of the study is to reconstruct hydrodynamic data obtained from the numerical solution of the shallow water equations. A larger weight is therefore assigned to the data-driven loss to ensure that the ConvLSTM model accurately captures the dominant hydrodynamic patterns present in the numerical data. At the same time, a smaller weight is assigned to the physics-informed loss in order to guide the training in regions where data are sparse or unavailable, without overly constraining the model. Since the training data are generated using a finite-volume method with the SRNH scheme [24], which is known to introduce numerical diffusion, assigning equal weights to the data and PDE losses may lead to training instability. In such a case, the physics-based loss would attempt to enforce a less diffusive solution, while the data-driven loss reflects the numerical diffusion present in the data, resulting in competing objectives during training.

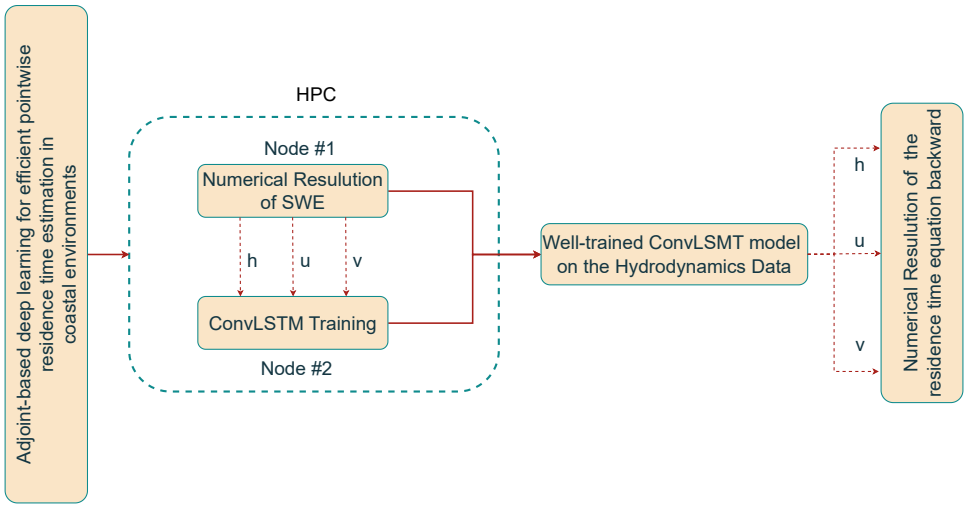


Figure 1: Schematic of the proposed hybrid framework integrating a finite volume solver (FVM-SRNH) with a ConvLSTM network for forward simulation and backward residence time computation. HPC refers to High Performance Computing.

Fig. 1 illustrates the main steps of our method. The process begins by solving the SWE using the SRNH numerical solver to generate the hydrodynamic data (u, v, h). During the initial waiting period Δt_w (i.e., from t_0 to $t_0 + \Delta t_w$), the solver runs independently without any training, in order to accumulate a sufficient sequence of data needed for ConvLSTM input.

Once the initial buffer is filled, the ConvLSTM model starts its training. From this point onward, the numerical solver and the ConvLSTM training run in parallel. The simulation continues to produce new data, which are transferred in real time to the training node via a shared memory buffer. This setup involves two computational nodes on the HPC system: one for solving the SWE and another for training the ConvLSTM model.

At the end of the simulation, the ConvLSTM continues training for an additional period to ensure it covers the full duration of the simulation, including the final time steps. The result of this procedure is a trained ConvLSTM model that accurately reproduces the hydrodynamic fields required for backward integration of the residence time equation.

3.4. Training data and reduction strategy

The training dataset is generated from high-resolution numerical solutions of the SWE using the SRNH scheme. However, directly using full-resolution outputs for training poses substantial computational and memory challenges. To address this, we employ a data reduction strategy that preserves the essential dynamics while significantly reducing the dataset size [25].

A straightforward uniform downsampling approach was initially considered for data reduction. However, it proved to be insufficient, as it occasionally omitted critical information, particularly in regions where rapid changes in water velocity and depth occur. Since these regions are essential for capturing the non-linear dynamics of shallow water flows, a more adaptive approach was required to ensure that the training dataset retained the most informative hydrodynamic states.

To achieve this, we enforced an adaptation technique based on the normalized gradient of the water velocity. This method allows for an adaptive selection of training data points, ensuring that regions with significant flow variations, such as strong gradients, are well represented. The adaptation criterion is defined as:

$$C^n(T_i) = \frac{\|\nabla u(T_i)\|}{\max_{T_j} \|\nabla u(T_j)\|} \tag{14}$$

where $\|\nabla u(T_i)\|$ represents the Euclidean norm of the velocity gradient within a given triangular element T_i. The normalization ensures that the adaptation criterion values remain within the interval $[0, 1]$, enabling a systematic selection of data points based on their relative importance to the hydrodynamic dynamics. A threshold-based mechanism is used to retain data from elements with strong gradients, while reducing redundancy in smoother regions. This ensures that the ConvLSTM model is trained on hydrodynamically relevant inputs. Fig. 2 illustrates the effect of this adaptive sampling compared to using all points.

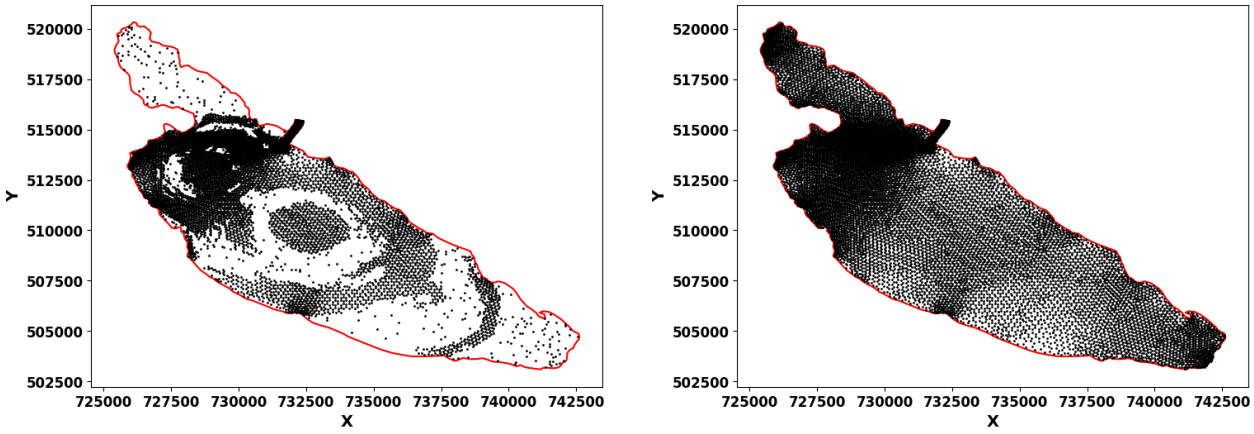


Figure 2: Data reduction points are chosen to reduce training cost, with adaptation points aligned along the current patterns. The right figure shows the selected points after adaptation, while the left figure displays the full mesh points before reduction.

4. Numerical results

4.1. Numerical experiment on a 2D Channel

In this test, we consider a two-dimensional channel with unidirectional flow, featuring both an inlet and an outlet, to validate the ability of the ConvLSTM model to reproduce results with accuracy comparable to classical methods that rely on data storage. The channel spans the domain $[0, 1] \times [0, 0.5]$; see Fig. 3. Initially, the residence time is set to zero, and Dirichlet boundary conditions with a value of zero are imposed at both the inlet and the outlet. A constant flow velocity of 1 m/s is prescribed. The simulation is performed for three different diffusion coefficients: $\kappa = 0.001$, 0.01, and 0.1 with $\Delta t_w = 0.2s$. The problem is solved using both a traditional data storage approach and

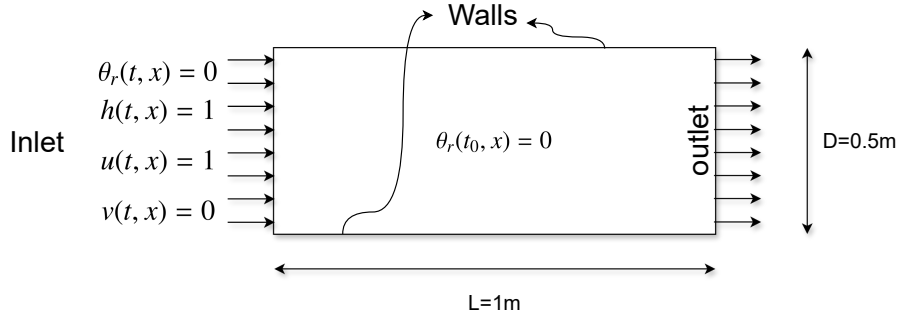


Figure 3: Illustration of the 2D channel.

the ConvLSTM model. The results show a strong agreement between the two methods. This is illustrated in Fig. 4, which presents a cross-sectional comparison of the computed residence times. Table 1 summarizes the relative error in the L_2 norm with respect to the analytical solution, as well as the CPU time and memory requirements for both the data storage approach and the ConvLSTM model. The analytical solution for the residence time is given by:

$$\theta_r(x) = \frac{L}{U}\left[(1-\sigma) - \frac{e^{-\mathrm{Pe}\,\sigma} - e^{-\mathrm{Pe}}}{1 - e^{-\mathrm{Pe}}}\right], \tag{15}$$

where $\sigma = \frac{x}{L}$ is the dimensionless spatial coordinate and $\mathrm{Pe} = \frac{UL}{K}$ is the Peclet number. Here, L denotes the length of the domain, U is the constant along-channel velocity, and K is the diffusion coefficient.

Table 1: Relative error in the L_2 norm of the residence time for different diffusion coefficients in the 2D channel test case.

Diffusion Coefficients κ	**0.001**	**0.01**	**0.1**	**CPU Time**
Backward with Data	4.350×10^{-3}	1.875×10^{-3}	1.431×10^{-3}	4.2 minutes
Backward with ConvLSTM	9.977×10^{-3}	9.783×10^{-3}	8.214×10^{-3}	2.6 minutes

4.2. Application to the Nador Lagoon

4.2.1. Numerical setup

The Nador Lagoon, located along Morocco's Mediterranean coast near 35°N, serves as the main focus of this study see Fig. 5(a). It covers a surface area of approximately 115 km^2 and holds a volume of about 542 Mm^3, making it a key site for investigating regional hydrodynamics. The computational mesh see Fig. 5(b) comprises 73,322 elements and 36,612 nodes, with a higher node density concentrated within the lagoon to improve spatial resolution. Particular care was taken to refine the coastline representation, ensuring smooth boundaries and more accurate treatment of boundary conditions. To solve the SWE defined in Eq. (2), a large computational domain was selected, encompassing the lagoon and part of the adjacent Mediterranean Sea. This broader domain, shown in Fig. 5(a), allows the imposition of realistic open boundary conditions. These were derived from the TPXO model [26], which also provides the initial conditions for the simulation. Meshing was performed using Gmsh [27].

Although the SWE are solved over the entire domain, only the Nador Lagoon–partitioned into eight blocks as shown in Fig. 5(b) is used during the training phase of the ConvLSTM model. This restriction stems from the backward integration of the residence time Eq. (1), which is only required within the region of interest.

The numerical model was implemented within the Manapy framework[1] [28], a parallel, Python-based platform for solving partial differential equations using an object-oriented finite-volume solver on unstructured meshes. The

[1] `https://github.com/imadki/manapy`

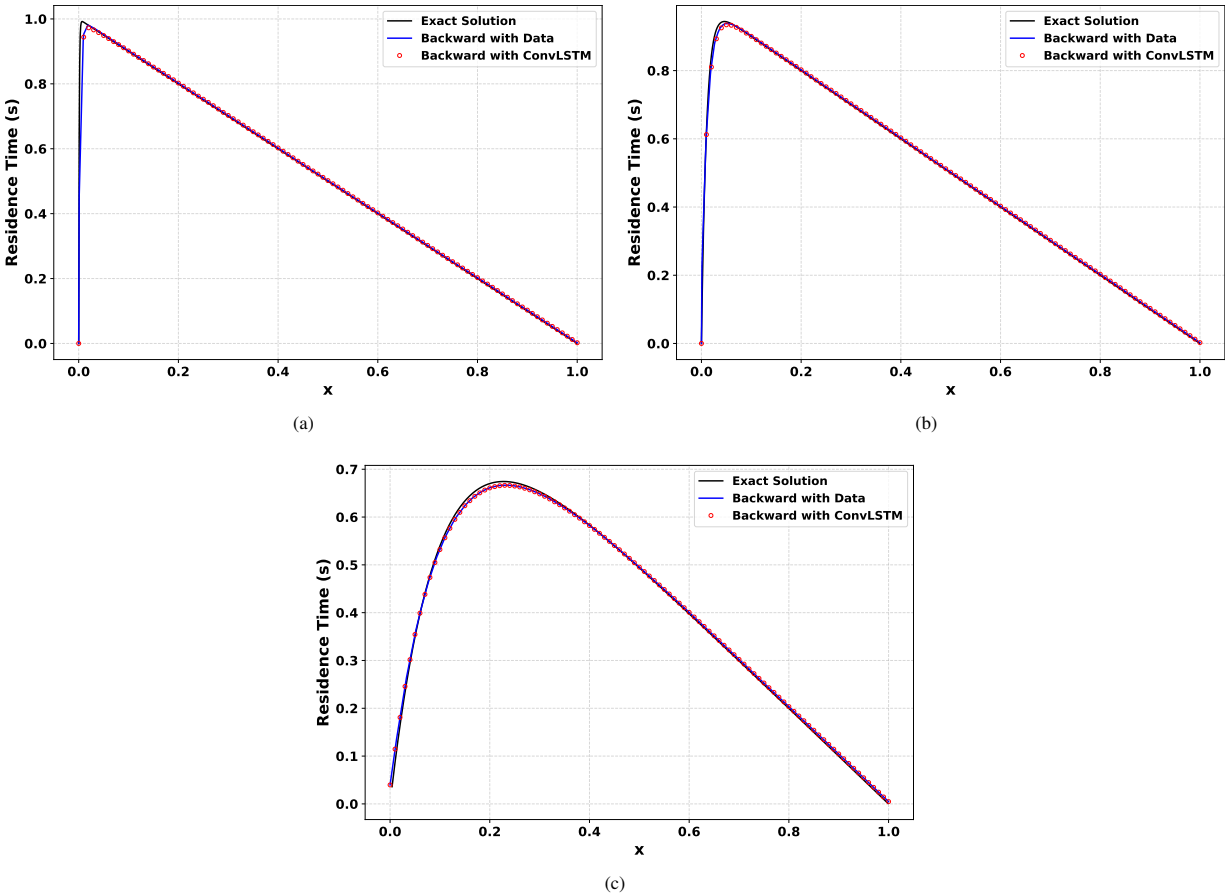


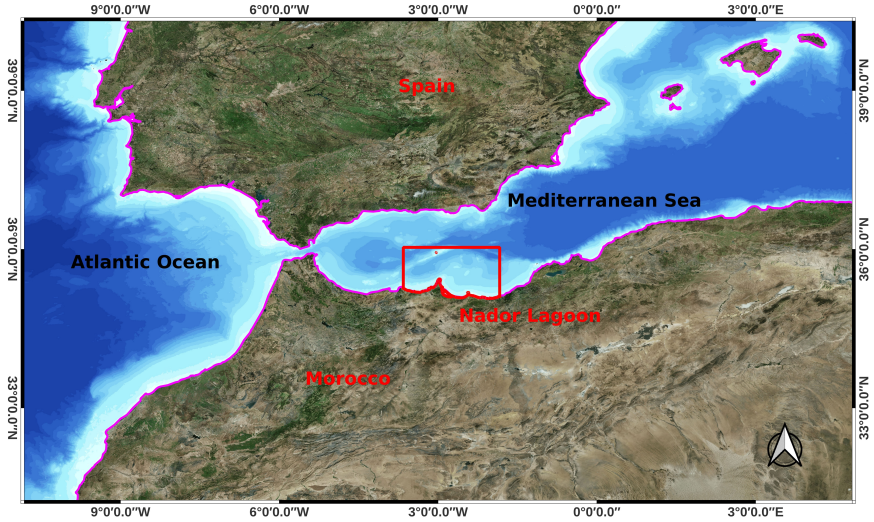
Figure 4: Spatial distribution of the residence time along the cross-section $y = 0.25$ for different diffusion coefficients: (a) $\kappa = 0.001$, (b) $\kappa = 0.01$, and (c) $\kappa = 0.1$.

training of the model was carried out on the Toubkal Supercomputer [29] at Mohammed VI Polytechnic University (UM6P). The network architecture consists of two main components: a convolutional encoder and a recurrent temporal processor. The encoder includes six successive convolutional layers, each with 128 neurons, which progressively extract increasingly abstract and complex spatial features from the input data. These features represent key hydrodynamic variables, including the velocity components u, v, and the water surface elevation h. The extracted spatial features are then passed into a stack of six ConvLSTM layers, which are specifically designed to model the temporal evolution of spatial patterns. By maintaining both memory and spatial coherence across time steps, the ConvLSTM layers effectively capture short- and long-term temporal dependencies. This makes the model particularly suitable for forecasting and data assimilation tasks in dynamic environmental systems. The final decoder layer transforms the learned temporal features into the target output variables, such as velocity fields and water surface elevation.

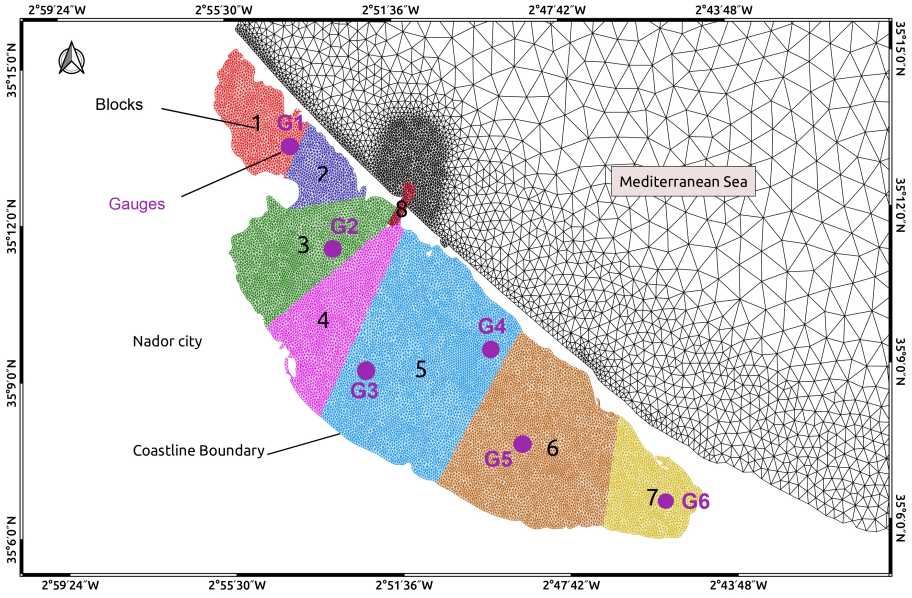
4.2.2. Validation of hydrodynamics and residence time

We next evaluate the ability of the ConvLSTM model to reproduce the hydrodynamic fields required for backward integration of the residence time equation. Fig. 6 shows a comparison of the velocity components in the x and y directions between the reference numerical simulation and the ConvLSTM predictions at different locations within the Nador Lagoon. The results indicate good agreement, demonstrating that the surrogate model can effectively reconstruct the high-resolution fields generated by the SWE solver.

Fig. 7 shows the isovalues of the current velocities from the numerical simulation (a), the ConvLSTM prediction (b), and the absolute error between the two approaches (c). The error is highest in the region near the channel connecting the lagoon to the Mediterranean Sea, where the flow is more complex and changes rapidly. This complexity makes it



((a)) Satellite-based visualization of the computational domain encompassing Nador Lagoon. The map was generated using QGIS and high-resolution satellite imagery from Bing Maps. Bathymetry data were extracted from EMODnet, and the 10-meter resolution coastline from OpenStreetMap.



((b)) Zoomed-in view of the computational mesh over Nador Lagoon, generated using the open-source software Gmsh.

Figure 5: Geospatial and numerical representation of the Nador Lagoon domain.

challenging for the ConvLSTM model to accurately capture the hydrodynamics in that area. Nevertheless, the overall error remains low around 10^{-3}, demonstrating that the ConvLSTM can effectively learn the hydrodynamic patterns using only a small subset of the simulation data. This significantly reduces the need to store full spatiotemporal outputs from the forward simulation (14 TB), enabling a much faster and more efficient backward computation of residence time compared to traditional methods.

The results shown in Figs. 8-9 illustrate the distribution of residence time in the lagoon. Fig. 8 presents the

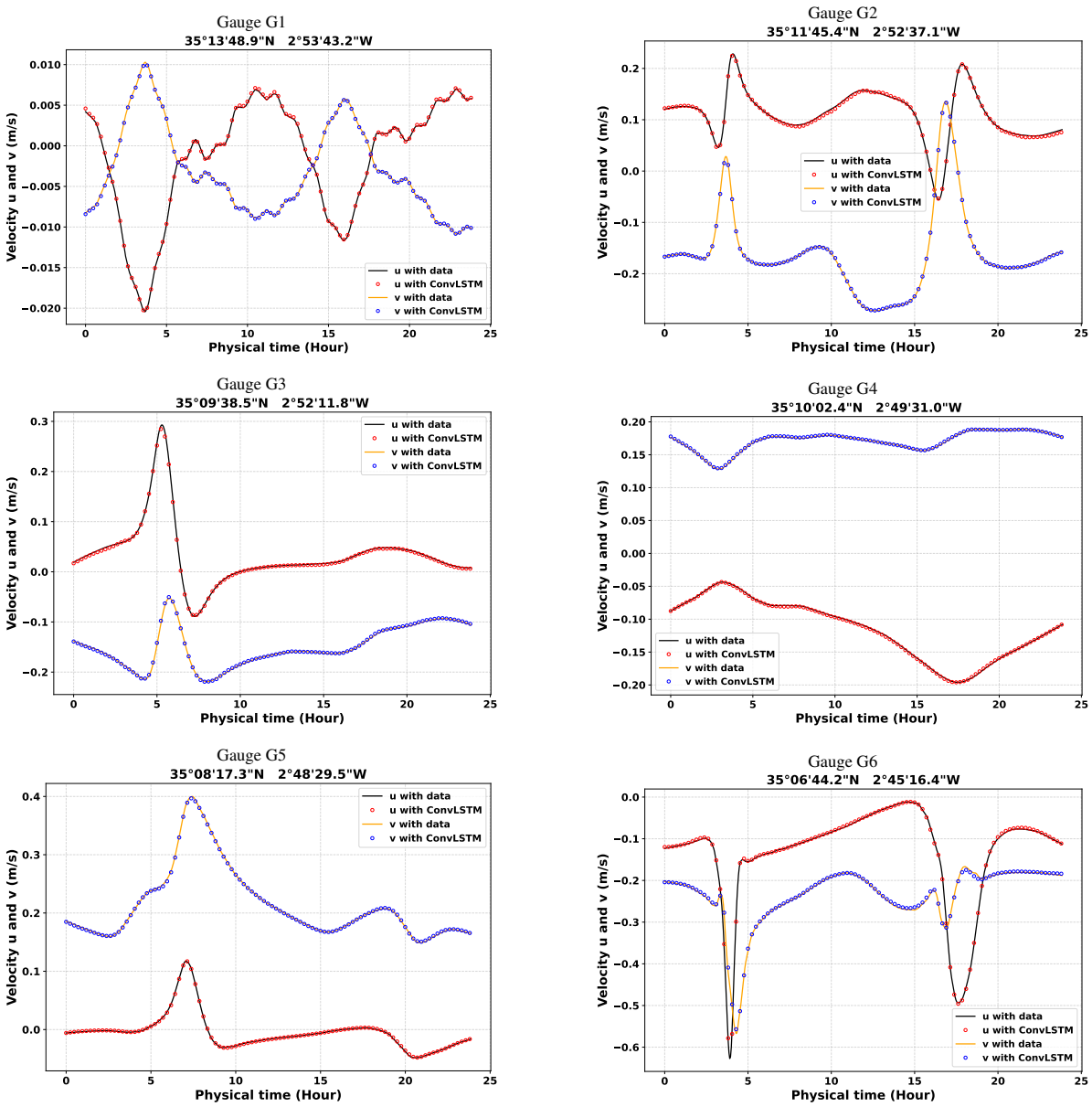
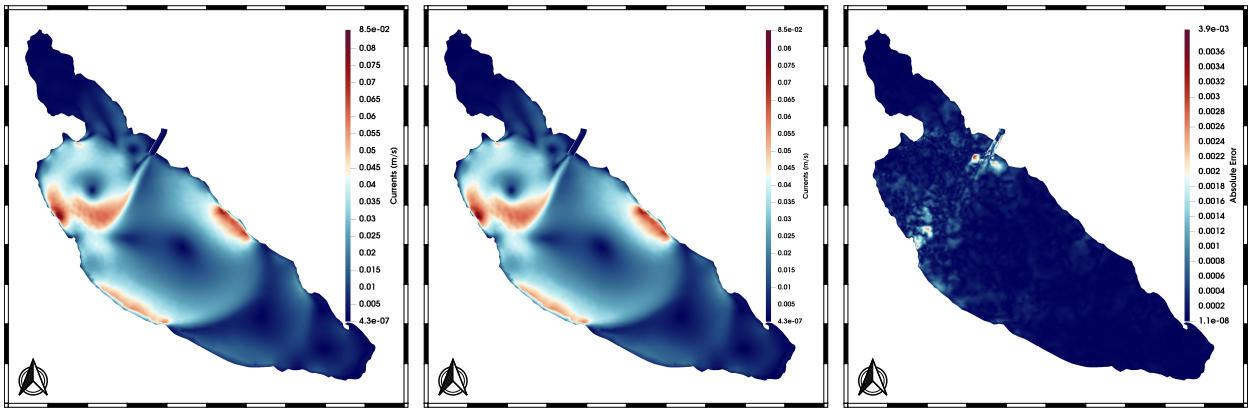


Figure 6: Comparison between the data storage solution and the ConvLSTM prediction of hydrodynamics at different locations in the Nador Lagoon.

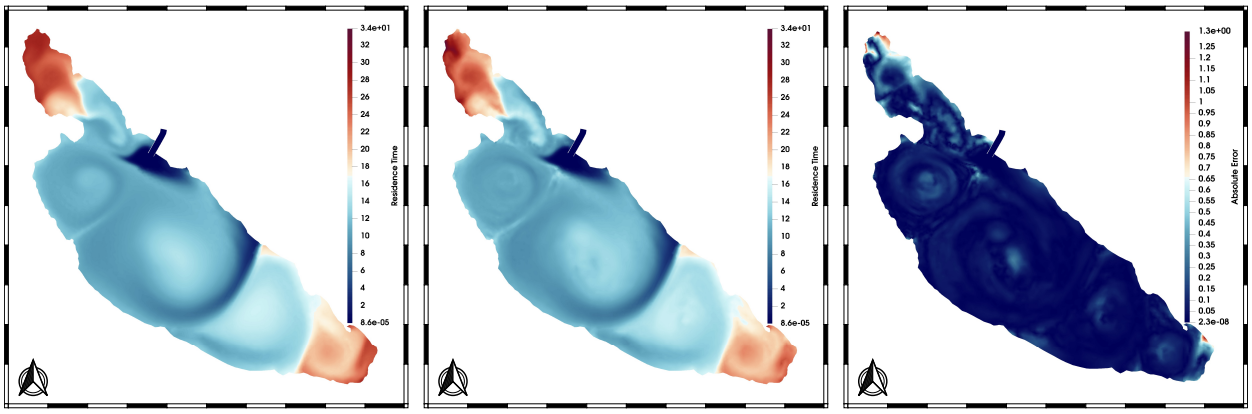
residence time after 25 days of physical simulation, prior to full convergence. At this stage, the error is primarily concentrated in regions with high residence time and low velocity. Nevertheless, both the numerical method and the ConvLSTM model yield closely aligned results. Fig. 9 displays the residence time distribution after full convergence. Although the ConvLSTM and numerical results remain in good agreement, the error increases slightly due to accumulation in high residence time regions. However, this error remains within acceptable limits, as a difference of approximately three days in residence time is negligible compared to the significant computational savings from two days of simulation time to just three hours using ConvLSTM.

The result shown in the Fig. 10 represents the residence time evolution over time, showing the good matching between the numerical methods and the ConvLSTM model, except at some point in which there is some difference between



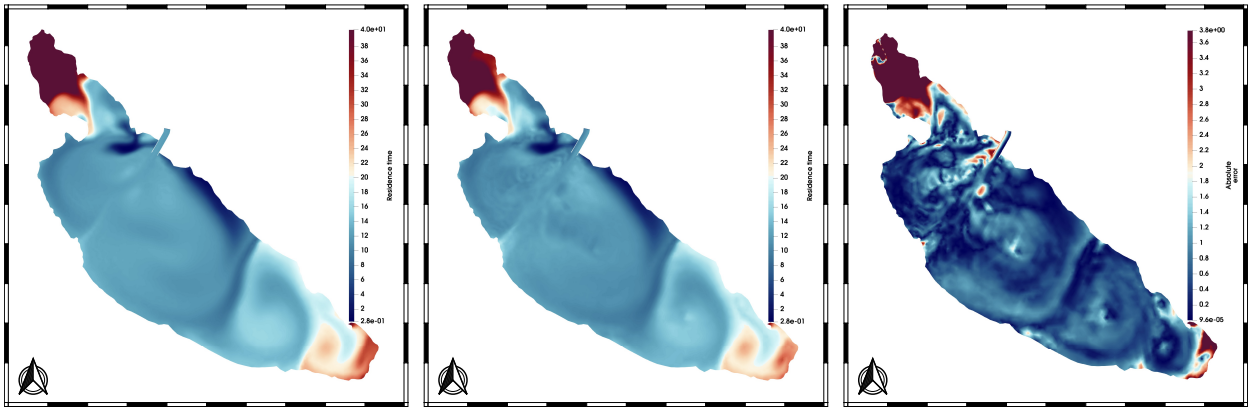
(a) (b) (c)

Figure 7: Comparison of current isovalues: (a) numerical solution, (b) ConvLSTM prediction, and (c) absolute error.



(a) (b) (c)

Figure 8: Residence time distribution in the Nador Lagoon before the convergence is reached: (a) Data storage results, (b) ConvLSTM prediction results, (c) Absolute error.



(a) (b) (c)

Figure 9: Residence time distribution in the Nador Lagoon at convergence: (a) Stored data results, (b) ConvLSTM prediction results, (c) Absolute error.

the two approaches because of the error accumulation that is illustrated in the Figs. 8-9.

Table 2: Comparison of average residence time [day] across lagoon blocks using ConvLSTM and traditional data storage approaches. The table also reports the CPU time for solving the residence time equation, the memory required for data storage and ConvLSTM parameters, and the training time.

Block / Metric	**Data storing**	**ConvLSTM (Data Reduction)**	**ConvLSTM (without Reduction)**
B1	46.84	42.36	43.92
B2	19.72	19.12	19.02
B3	14.35	13.67	13.62
B4	15.08	15.76	14.21
B5	13.75	13.94	13.35
B6	19.65	19.44	19.10
B7	36.32	35.12	36.17
CPU time	2 days	3 hours	3 hours
Memory	14 TB	800 MB	800 MB
Training time*	–	8 hours	18 hours

** The reported training time is not added separately, as ConvLSTM training is performed in parallel with the SWE resolution (see Fig. 1).*

The results in Table 2 present the residence time computed using the classical numerical method, which requires storing the full simulation data and reusing it during the backward integration of Eq. (1). This approach is computationally expensive, as reflected in the table. Additionally, Table 2 highlights the impact of data reduction during the training process. It compares the use of the full simulation dataset with a reduced dataset. While training with the full dataset is significantly more costly, the results show that using a reduced dataset yields nearly the same level of accuracy in residence time estimation. A quantitative analysis of the relative error in both the L_2 and L_∞ norms was

Table 3: Relative L_2 and L_∞ errors computed at multiple locations within the Nador Lagoon over a 25-hour hydrodynamic time series (h, u, v).

Norm	**Gauge**	h	u	v
L_2	G1	8.031×10^{-3}	1.929×10^{-2}	1.409×10^{-2}
	G2	5.841×10^{-3}	1.929×10^{-2}	1.409×10^{-2}
	G3	1.659×10^{-3}	7.572×10^{-3}	1.050×10^{-2}
	G4	5.317×10^{-3}	1.806×10^{-2}	1.575×10^{-2}
	G5	3.358×10^{-3}	2.595×10^{-2}	5.293×10^{-3}
	G6	1.029×10^{-2}	1.572×10^{-2}	1.332×10^{-2}
L_∞	G1	1.003×10^{-2}	1.896×10^{-2}	2.565×10^{-2}
	G2	9.384×10^{-3}	1.896×10^{-2}	2.565×10^{-2}
	G3	6.275×10^{-3}	1.533×10^{-2}	2.332×10^{-2}
	G4	8.692×10^{-3}	3.468×10^{-2}	2.416×10^{-2}
	G5	1.301×10^{-2}	2.767×10^{-2}	3.830×10^{-3}
	G6	4.304×10^{-3}	2.887×10^{-2}	1.941×10^{-2}

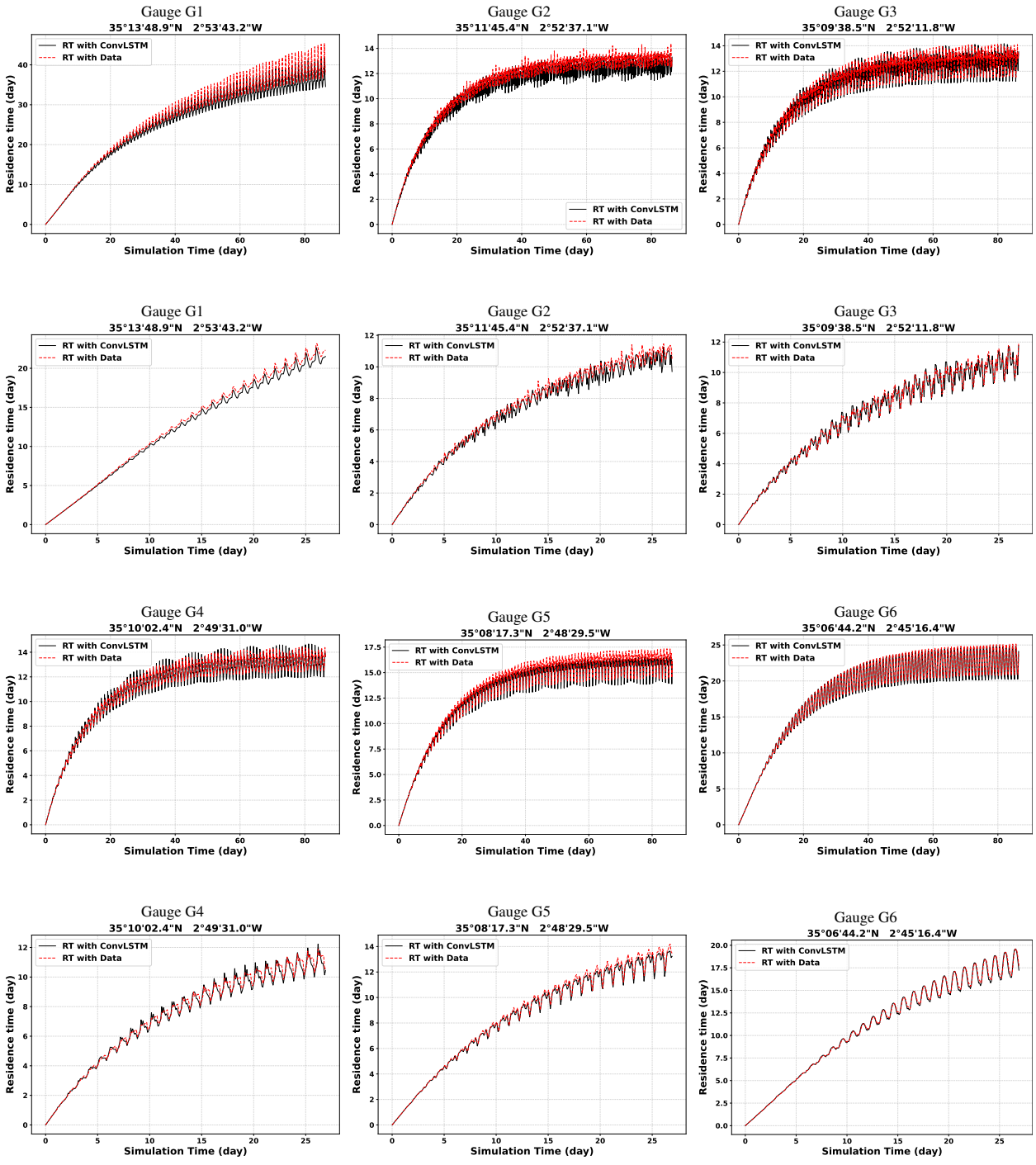


Figure 10: Evolution of residence time over the simulation period using both the classical method and the ConvLSTM approach. The first and third rows show the entire simulation period, while the second and fourth rows provide a zoomed-in view of the first 25 days, highlighting finer details and the close alignment between the two approaches.

performed to evaluate the accuracy of the proposed approach. Tables 3 present the relative L_2 and L_∞ errors computed at multiple locations within the Nador Lagoon over a 25-hour hydrodynamic time series, which corresponds to a full tidal cycle in the sea adjacent to the lagoon. The results indicate consistently low error values, suggesting that the ConvLSTM model employed in this study accurately reproduces the hydrodynamics necessary for the backward integration of the residence time equation. As shown in Table 4, the relative error in residence time across different locations and throughout the time series remains very low, confirming the model's effectiveness in capturing the hydrodynamic behavior relevant to residence time estimation. Furthermore, Table 5 reports the spatially averaged error, which also demonstrates low values, reinforcing the validity and reliability of the proposed approach.

Table 4: Relative error on the time series of residence time evolution, computed at multiple locations within the Nador Lagoon over 86 days, evaluated using the L_2 and L_∞ norms.

Gauge	L_2	L_∞
G1	1.060×10^{-2}	2.220×10^{-2}
G2	4.323×10^{-2}	7.491×10^{-2}
G3	1.343×10^{-2}	7.198×10^{-2}
G4	6.156×10^{-2}	9.260×10^{-2}
G5	3.912×10^{-2}	6.249×10^{-2}
G6	2.503×10^{-2}	4.061×10^{-2}

Table 5: Spatially averaged relative error in the L_2 and L_∞ norms for the velocity components (u, v) and the residence time.

Error	**h**	**u**	**v**	**Residence Time**
L_2	3.879×10^{-2}	7.512×10^{-2}	6.023×10^{-2}	8.231×10^{-2}
L_∞	7.303×10^{-2}	9.638×10^{-2}	8.377×10^{-2}	9.471×10^{-2}

The numerical resolution of the shallow-water equations using the Finite Volume method with the SRNH scheme is implemented within the Manapy framework. Distributed computing is leveraged through OpenMPI to distribute computational tasks across multiple CPUs, enabling high-performance computing. The domain partitioning is performed to ensure that each processor manages an approximately equal volume of control, thus balancing the computational load. However, due to mesh refinement, some processors handle smaller, more refined regions, while others manage larger, coarser regions. Refinement is particularly applied to regions exhibiting significant variations in tides and currents to ensure high accuracy in the numerical simulations.

Tables 6 and 7 highlight the effect of the number of CPUs on computational time, demonstrating that computations performed on 56 CPUs complete more rapidly than those executed on 1, 14, or 28 CPUs.

Table 6: Total CPU time required for the forward resolution of the Shallow Water Equations and the training of the ConvLSTM model using different numbers of CPUs

#CPUs	**1**	**14**	**28**	**56**
time[Hour]	558.48	43.89	21.06	8.28

The ConvLSTM model training employs a similar distributed approach. Instead of training a single model on the entire domain using one processor, which is computationally expensive and slow, the domain is partitioned, and each processor independently trains a ConvLSTM model on its assigned region. At the end of training, each region has its model, which is subsequently utilized in the backward residence time computations. However, this distributed training

Table 7: CPU time required for the backward resolution of the residence time equation using different numbers of CPUs

#CPUs	1	14	28	56	Unit of time
With ConvLSTM	143.83	10.60	5.58	3.10	Hour
With data storing	82*	6.46	3.40	2.00	day

** The CPU time for a single processor was not measured directly, as executing the simulation on one processor would require an impractically long runtime. Instead, the value was approximated by extrapolating from the recorded runtimes obtained using 14, 28, and 56 processors.*

approach introduces inaccuracies at the boundaries between adjacent regions, as each model is trained only on local data specific to its assigned area. To mitigate this issue, we augment the local dataset for each region by incorporating additional data from neighboring regions see Fig. 11. Specifically, neighboring processors exchange data from their common boundaries to include relevant neighboring region information in their training datasets. This exchange of boundary data significantly improves prediction accuracy across processor boundaries.

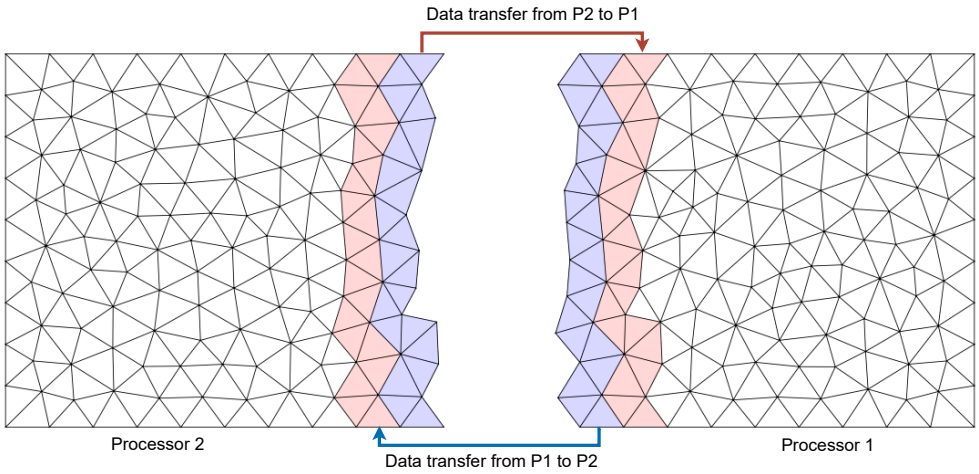


Figure 11: Schematic of the additional data from neighboring regions that should be added to the data that each processor must be trained on to solve the boundaries problem.

5. Conclusion

In this study, we introduced a deep learning-based framework to facilitate the computation of pointwise residence time in coastal environments. The method addresses key limitations of traditional techniques, such as the need for extensive data storage and the high computational cost associated with solving the adjoint problem backward in time. By training a deep neural network on hydrodynamic data, the model learns the underlying flow dynamics without requiring the storage of full spatiotemporal datasets. With the incorporation of data reduction and adaptation techniques, the training cost is significantly optimized while maintaining strong predictive performance.

Once trained, the model can reproduce and predict the hydrodynamic fields necessary for backward-in-time residence time estimation. To validate the method, numerical experiments were conducted by comparing the model's predictions against reference data from high-fidelity numerical simulations. The results show a strong agreement, confirming the effectiveness and accuracy of the proposed approach. The model demonstrates high accuracy in computing pointwise residence time, while also achieving a substantial reduction in computational cost–from approximately two days to just three hours of CPU time.

Despite the promising results, the model must be fine-tuned when applied to different time intervals, boundary conditions, or domains of interest. This can be achieved by initializing the model with previously trained parameters

that already capture the known hydrodynamic patterns, rather than using random initialization, which would require training the model from scratch. As a future direction, we plan to extend this work using physics-informed neural networks (PINNs) [30, 31] to directly solve the hydrodynamics governed by the SWE. This will be particularly challenging due to the nonlinear and hyperbolic nature of the system, but it holds great potential for advancing data-driven modeling in environmental fluid dynamics.

Conflict of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the support and computing resources provided by the Toubkal Supercomputer [29] (https://toubkal.um6p.ma/) at UM6P, Morocco. The authors would also like to express their gratitude to Imad Elmahi, Professor at the National School of Applied Sciences in Oujda, for his guidance and valuable support throughout this study.

Appendix A

Table 8 lists the physical parameters, their mathematical symbols, and reference values used in the numerical simulations to generate the hydrodynamic data required for training the ConvLSTM model. Table 9 summarizes the neural network architecture and training parameters adopted in this study. Together, these parameters provide sufficient information to ensure the reproducibility of the results presented in this paper.

Table 8: Physical parameters and reference quantities used in this study.

Symbol	Description	Reference Value
ρ_w	Density of water	10^3 kg/m^3
ρ_a	Density of air	1.293 kg/m^3
g	Gravitational acceleration	9.81 m/s^2
ν	Kinematic viscosity of water	10^{-6} m^2/s
κ	Molecular diffusivity	2×10^{-9} m^2/s
n_b	Manning's roughness coefficient	2.5×10^{-3} s/m$^{1/3}$
ω	Earth's angular velocity	7.272×10^{-5} rad/s

Table 9: Neural network architecture and training parameters

Parameter	Value
Number of convolutional layers	6
Number of ConvLSTM layers	6
Number of Neurons per layer	128
kernel size	3×3
Activation function (Conv layers)	Tanh
Activation function (ConvLSTM layers)	Tanh, sigmoid
Optimizer	Adam
Learning rate	10^{-3}
Batch size	32
Number of epochs	500

Appendix B

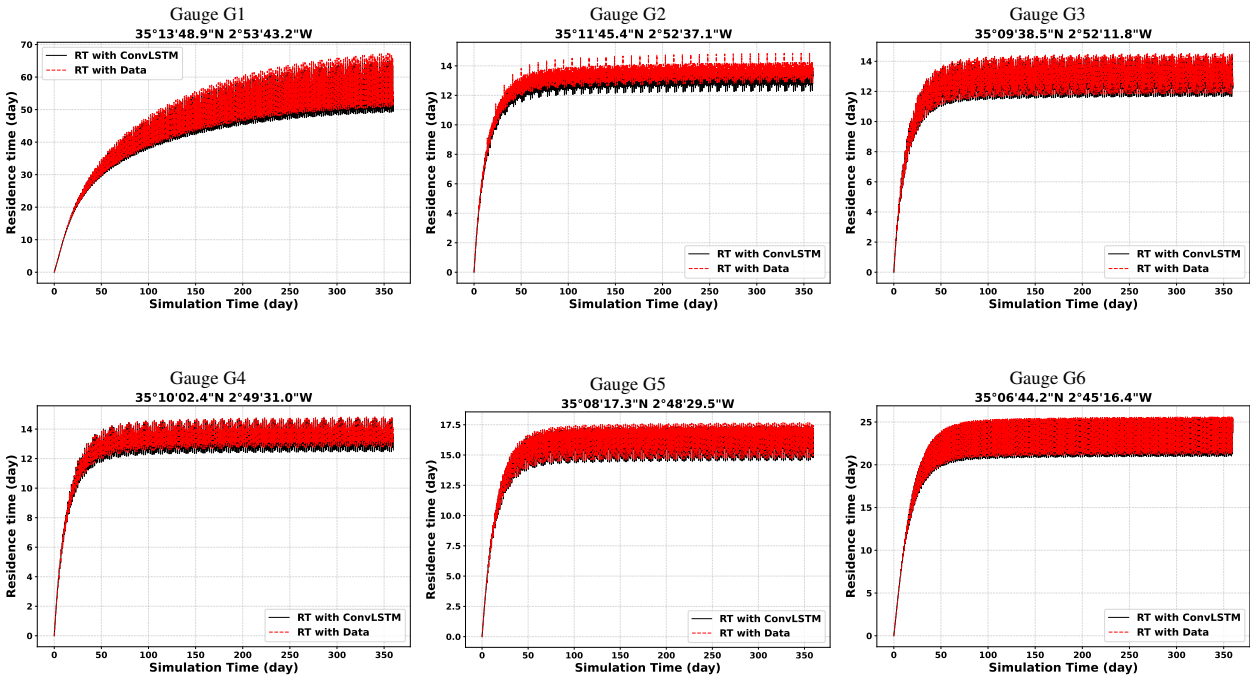
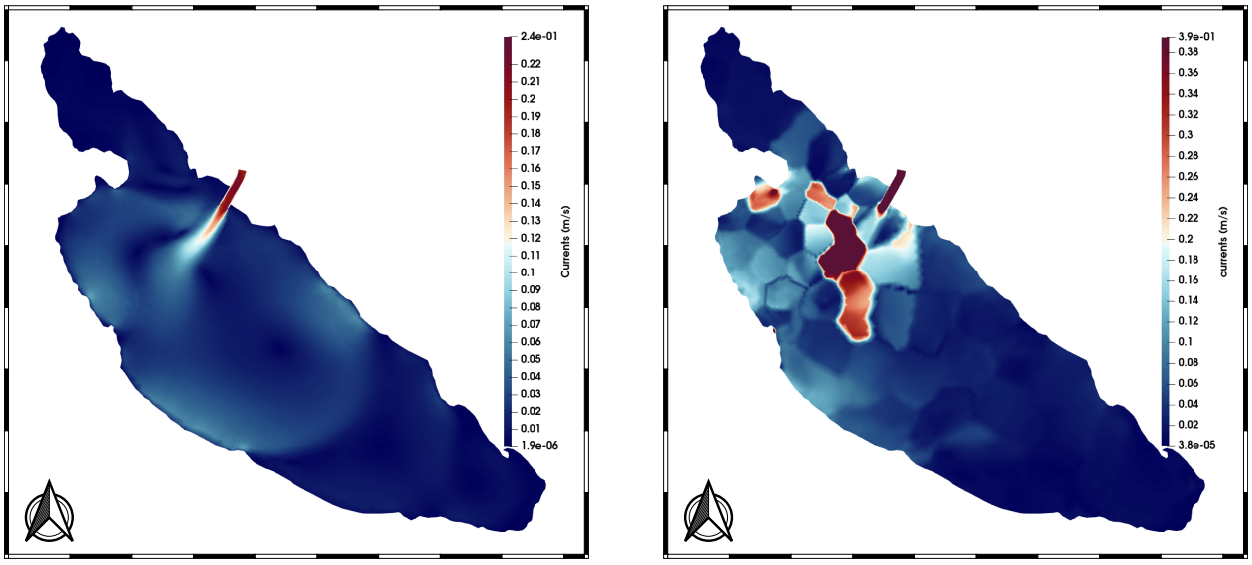


Figure 12: Comparison between the data storage solution and the ConvLSTM prediction of residence time at different locations in the Nador Lagoon over 365 days.

Table 10: Relative error on the time series of residence time evolution, computed at multiple locations within the Nador Lagoon over one year.

Gauge	L_2	L_∞
G1	6.858×10^{-2}	8.672×10^{-2}
G2	1.491×10^{-1}	2.424×10^{-1}
G3	5.532×10^{-2}	9.045×10^{-2}
G4	8.146×10^{-2}	9.859×10^{-2}
G5	5.642×10^{-2}	7.431×10^{-2}
G6	4.235×10^{-2}	5.061×10^{-2}

Fig. 12 illustrates a comparison between the data-storage–based solution and the ConvLSTM prediction of the residence time over one year of simulation at the same locations considered in Fig. 10. This duration corresponds to four times the period required to compute the residence time in the Nador lagoon. Table 10 reports the residence time errors at these locations over the entire simulation year. The long-term simulation is performed to examine whether error accumulation grows unbounded over time. The results demonstrate that, once the residence time converges to its steady value, the prediction error stabilizes and remains bounded at a small constant level, even in regions displaying high accumulation of residence time error, see Fig. 12(G1)–(G2).



(a) (b)

Figure 13: Comparison of current isovalues: (a) numerical solution and (b) ConvLSTM prediction one day beyond the ConvLSTM training period.

Appendix C

Fig. 13 illustrates the current isovalues in the Nador lagoon one day beyond the ConvLSTM training period, highlighting the differences between the ConvLSTM predictions and the numerical solution. Since residence time computation strongly depends on the underlying hydrodynamics, inaccuracies in the hydrodynamic fields directly lead to erroneous residence time estimates. These results indicate that model fine-tuning becomes necessary when the residence time is computed over a time interval that is not included in the training period.

References

[1] Lisa V Lucas and Eric Deleersnijder. Timescale methods for simplifying, understanding and modeling biophysical and water quality processes in coastal aquatic ecosystems: A review. *Water*, 12(10):2717, 2020.

[2] Fayssal Benkhaldoun, Imad Elmahi, and Mohammed Seaïd. Application of mesh-adaptation for pollutant transport by water flow. *Mathematics and Computers in Simulation*, 79(12):3415–3423, 2009. The International Conference on Approximation Methods and numerical Modeling in Environment and Natural Resources.

[3] Feng Zhang, Xiaoxiao Sun, Yan Zhou, Congjiao Zhao, Zhenhong Du, and RenYi Liu. Ecosystem health assessment in coastal waters by considering spatio-temporal variations with intense anthropogenic disturbance. *Environmental Modelling & Software*, 96:128–139, 2017.

[4] V H Hewageegana and M Olabarrieta. Main physical processes affecting the residence times of a micro-tidal estuary. *Journal of Marine Science and Engineering*, 11(7):1333, 2023.

[5] Éric J.M. Delhez, Arnold W. Heemink, and Éric Deleersnijder. Residence time in a semi-enclosed domain from the solution of an adjoint problem. *Estuarine, Coastal and Shelf Science*, 61(4):691–702, 2004.

[6] Bryan D Downing, Brian A Bergamaschi, Carol Kendall, Tamara E C Kraus, Fred Anderson, Megan B Young, and Adina Paytan. Using continuous underway isotope measurements to map water residence time in hydrodynamically complex tidal environments. *Environmental Science & Technology*, 50(17):9361–9371, 2016.

[7] Hisashi Takeoka. Fundamental concepts of exchange and transport time scales in a coastal sea. *Continental Shelf Research*, 3(3):311–326, 1984.

[8] I Oubarka, A Yachouti, I Elmahi, and Eric Deleersnijder. Unstructured finite volume solver for the computation of the residence time in the Nador lagoon, Morocco. In *International Conference on Mathematical And Computational Modelling, Approximation and Simulation*, pages 173–192. Springer, 2023.

[9] Syed Hyder Ali Muttaqi Shah, Arnold Willem Heemink, Ulf Gräwe, and Eric Deleersnijder. Adaptive time stepping algorithm for lagrangian transport models: Theory and idealised test cases. *Ocean Modelling*, 68:9–21, 2013.

[10] Gil Gonçalves, Umberto Andriolo, Luís Pinto, and Filipa Bessa. Mapping marine litter using uas on a beach-dune system: a multidisciplinary approach. *Science of The Total Environment*, 703:134805, 2020.

[11] Edith M van der Lee and Lars Umlauf. Internal wave mixing in the baltic sea: Near-inertial waves in the absence of tides. *Journal of Geophysical Research: Oceans*, 116, 2011.

[12] Ali Haddach, Hassan Smaoui, and Bouchaib Radi. Lattice boltzmann simulation of pollutant transport in shallow water flows: Application to Nador lagoon. *Journal of Computational Science*, 85:102538, 2025.
[13] Ismail Oubarka, Imad Kissami, Imad Elmahi, and Eric Deleersnijder. A robust and well-balanced finite volume solver for investigating the effects of tides on water renewal timescale in the Nador lagoon, Morocco. *Mathematics and Computers in Simulation*, 226:511–523, 2024.
[14] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. 28:802–810, 2015.
[15] Zhiqiang Wei and Adam Davison. A convolutional neural network based model to predict nearshore waves and hydrodynamics. *Coastal Engineering*, 171:103638, 2022.
[16] Chen Yang, You-Kuan Zhang, Xiuyu Liang, Catherine Olschanowsky, Xiaofan Yang, and Reed Maxwell. Accelerating the lagrangian particle tracking of residence time distributions and source water mixing towards large scales. *Computers & Geosciences*, 151:104760, 2021.
[17] Jean-Frédéric Gerbeau and Benoît Perthame. *Derivation of viscous Saint-Venant system for laminar shallow water; numerical validation*. PhD thesis, INRIA, 2000.
[18] Fayssal Benkhaldoun, Salah Daoudi, Imad Elmahi, and Mohammed Seaid. Numerical modelling of sediment transport in the Nador lagoon (Morocco). *Applied Numerical Mathematics*, 62(12):1749–1766, 2012.
[19] Ismail Oubarka, Imad Kissami, Imad Elmahi, and Eric Deleersnijder. Numerical computation of the residence time related to the water renewal in the Nador lagoon. In *AIP Conference Proceedings*, volume 3034, page 090005. AIP Publishing LLC, 2024.
[20] Benjamin de Brye, Anouk de Brauwere, Olivier Gourgue, Eric JM Delhez, and Eric Deleersnijder. Water renewal timescales in the scheldt estuary. *Journal of Marine Systems*, 94:74–86, 2012.
[21] Jeancarlo M Fajardo-Urbina, Yang Liu, Sonja Georgievska, Ulf Gräwe, Herman JH Clercx, Theo Gerkema, and Matias Duran-Matute. Efficient deep learning surrogate method for predicting the transport of particle patches in coastal environments. *Marine Pollution Bulletin*, 209:117251, 2024.
[22] Se-Heon Jeong, Woo Kyoung Lee, Hyosub Kil, Soojeong Jang, Jeong-Heon Kim, and Young-Sil Kwak. Deep learning-based regional ionospheric total electron content prediction long short-term memory (LSTM) and convolutional LSTM approach. *Space Weather*, 22(1):e2023SW003763, 2024.
[23] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18(153):1–43, 2018.
[24] Fayssal Benkhaldoun and Laure Quivy. A non homogeneous riemann solver for shallow water and two phase flows. *Flow, Turbulence and Combustion*, 76(4):391–402, 2006.
[25] Fayssal Benkhaldoun, Imad Elmahi, and Mohammed Seaïd. A new finite volume method for flux-gradient and source-term balancing in shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 199(49-52):3324–3335, 2010.
[26] Gary D Egbert and Svetlana Y Erofeeva. Efficient inverse modeling of barotropic ocean tides. *Journal of Atmospheric and Oceanic technology*, 19(2):183–204, 2002.
[27] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.
[28] Imad Kissami. Manapy: an mpi-based python framework for solving poisson's equation using finite volume on unstructured-grid. In *AIP Conference Proceedings*, volume 3034, page 090004. AIP Publishing LLC, 2024.
[29] Imad Kissami, Robert Basmadjian, Othmane Chakir, and M Riduan Abid. Toubkal: a high-performance supercomputer powering scientific research in africa. *The Journal of Supercomputing*, 81(15):1–45, 2025.
[30] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving pdes. *Journal of Computational Physics*, 378:686–707, 2019.
[31] Alex Bihlo and Roman O Popovych. Physics-informed neural networks for the shallow-water equations on the sphere. *Journal of Computational Physics*, 456:111024, 2022.