Environmental Technology & Innovation 42 (2026) 104920

Contents lists available at ScienceDirect

Environmental Technology & Innovation

journal homepage: www.elsevier.com/locate/eti

A generalized cell-based model framework for contaminant dispersion in marine environments

Ying Ma *, Meichen Guo, Bart De Schutter

Delft Center for Systems and Control, Delft University of Technology, 2628 CD, Delft, The Netherlands

ARTICLE INFO

Keywords:
Contaminant dispersion
Marine pollution
Cell-based model (CBM)
Three-dimensional (3D) transport
Parameter identification

ABSTRACT

Dispersion modeling is crucial for marine environmental modeling and management. However, operational applications require a practical balance between model accuracy and computational efficiency. To address this challenge, we develop and validate a generalized cell-based model (CBM) framework for contaminant dispersion. The framework enhances physical realism through a novel three-dimensional (3D) transport model and a formulation for chemical reactions. Additionally, a new discretization-based approach is proposed to robustly relate the CBM's diffusion coefficient to its partial differential equation counterpart, improving performance in scenarios with sharp gradients of the concentration level. The proposed framework's favorable trade-off between accuracy and efficiency is demonstrated in a comparative simulation study, where the 3D CBM reduces computation time from 14.72 s to 0.06 s compared to finite-element methods (FEM), with a relative Root Mean Square Error (RMSE) of 7.67%. To demonstrate its practical applicability, the proposed framework is validated using ocean current and nitrate concentration data from the Copernicus Marine Environment Monitoring Service. After identifying a key model parameter from the data, the model's forward predictions accurately reproduce the observed nitrate concentration patterns, confirming its suitability for operational scenarios.

1. Introduction

Marine contaminants released from various sources, such as industrial discharge and accidental spills, are increasingly present in coastal and ocean waters (Leon and Bellas, 2023). These contaminants increase the concentration of the harmful substances in marine ecosystems, which threatens ecological balance and human health. In addition, the accumulation of contaminants can cause long-term biodiversity loss and decrease the resilience of marine ecosystems (Câmara et al., 2021). In this context, reliable models that capture the contaminant dispersion process in marine environments are a prerequisite for effectively assessing, predicting, and mitigating these impacts.

Dispersion modeling techniques for marine contaminants can broadly be classified into two groups: first-principles-based and data-driven approaches. While first-principles-based models are physically interpretable, they often suffer from high computational costs (Paul and Webb, 2008) or stability issues (Hirsch, 2007). In addition, they rely on accurate physical parameters (e.g., diffusion coefficients), whose identification from observational data requires consideration of data heterogeneity and the fact that concentration levels may span multiple orders of magnitude. Data-driven approaches can be efficient in the inference stage once trained, but their performance is highly dependent on the quality of the training datasets, which are scarce in the context of marine pollution

* Corresponding author.
E-mail addresses: y.ma-4@tudelft.nl (Y. Ma), m.guo@tudelft.nl (M. Guo), b.deschutter@tudelft.nl (B. De Schutter).

https://doi.org/10.1016/j.eti.2026.104920
Received 2 November 2025; Received in revised form 30 January 2026; Accepted 31 March 2026
Available online 1 April 2026

modeling (Guerrini et al., 2021). Moreover, most data-driven models operate as black boxes, lacking the full physical interpretability that is essential for risk assessment and decision making (Reichstein et al., 2019).

To address the respective limitations of the modeling approaches in real-world marine contaminant dispersion scenarios, this paper develops a modeling framework with a systematic workflow including parameter identification from observational data. This generalized cell-based framework builds on our previous work on the two-dimensional (2D) cell-based model (CBM) (Ma et al., 2025), which discretizes the 2D computational domain into uniform cells and characterizes the contaminant transport process by incorporating both water flow-driven convection and diffusion effects. The CBM has already demonstrated its advantages over conventional methods in 2D simulations (Ma et al., 2025): it is less affected by the Courant–Friedrichs–Lewy (CFL) conditions (De Moura and Kubrusly, 2013) compared to the finite-difference method (FDM), and it provides a favorable balance between computation time and accuracy compared to the FEM.

The current work makes the following main contributions:

1. To enhance the model's physical realism, we develop a three-dimensional (3D) framework that captures vertical transport, overcoming the limitation of the 2D model. This involves complex 3D flux calculations across 26 neighboring cells, incorporating key physical details such as depth-dependent diffusion coefficients. A simulation study is conducted to compare the solution of the 3D CBM with that of the finite-element method (FEM), revealing a balanced trade-off between computational efficiency and accuracy. This trade-off is particularly pronounced in computationally demanding 3D simulations, where first-principles-based models often suffer from high computational costs.
2. We develop a formulation for chemical reactions within the 3D CBM framework by discretizing the governing rate equations, enabling the model to simulate essential degradation processes, such as photodegradation, which are a critical component of contaminant transport.
3. A new discretization-based approach is proposed to relate the diffusion coefficient to its partial differential equation counterpart. This approach captures more detailed dynamic information by using a finer grid configuration that accounts for diagonal neighbors, demonstrating improved performance in scenarios with sharp gradients of the concentration level.
4. The framework's practical applicability in real-world scenarios is demonstrated through a systematic parameter identification and forward prediction workflow. This workflow is specifically designed to overcome the inherent challenges of oceanographic data from the Copernicus Marine Environment Monitoring Service (Von Schuckmann et al., 2018); for instance, it employs a relative Root Mean Square Error (RMSE) loss function to counteract model bias caused by concentration levels spanning several orders of magnitude and integrates preprocessing strategies to resolve mismatches between multi-resolution datasets (e.g., mismatches between ocean current data grids and biogeochemical observations).

This paper is organized as follows. Section 3 presents the theoretical formulation for the 3D CBM, which incorporates 3D convection and diffusion with dynamic coefficients, as well as chemical reactions. In Section 4, a comparative case study is conducted between the 3D CBM and the FEM. In Section 5, we propose a new approach for relating the diffusion coefficient formulation of the CBM with that of the partial differential equation counterpart, incorporating more dynamic information than the previous discretization-based approach. Section 6 develops the CBM parameter identification workflow and validates it using Copernicus datasets. Section 7 discusses the limitations of the CBM framework. Finally, Section 8 summarizes the conclusions and discusses future work.

2. Related work

As indicated in Section 1, marine contaminant dispersion modeling techniques can be broadly grouped into first-principles-based and data-driven approaches. This section briefly reviews the state-of-the-art literature in both categories and discusses parameter identification from observational data.

In first-principles-based approaches, the dispersion process is described by physical models, typically in the form of partial differential equations (PDEs), to capture both temporal and spatial evolution. To obtain numerical solutions, modeling methods such as the FEM and the finite-difference method (FDM) are widely applied. For example, the FEM has been used in an adaptive semi-Lagrangian scheme to simulate the pollution transport in the Mediterranean Sea (El-Amrani et al., 2024), and in a streamline-upwind Petrov–Galerkin method to model the concentration distribution in the Han River (Lee and Seo, 2007). Likewise, the FDM has been employed in a compact scheme for groundwater pollution transport in Li et al. (2020), and in a MacCormack method with a numerical Hansen filter for simulating dam-break flows through urban environments (Adityawan et al., 2023). Moreover, to capture the vertical transport and mixing process in real-world marine environments, some researchers have extended traditional numerical modeling methods to 3D domains. For example, Maljutenko et al. (2021) adopt a hydrodynamic-tracer framework based on the general estuarine transport model to investigate the basin-wide transport and vertical mixing of shipborne contaminants in the Baltic Sea. On the other hand, data-driven approaches, particularly those based on neural networks (NNs), learn to approximate the dynamics directly from observational data, rather than solving PDEs. For instance, a convolutional neural network model is used for eutrophication status classification in Lee et al. (2024). Wullems et al. (2023) use a long short-term memory model for forecasting estuarine salt intrusion in the Rhine–Meuse delta.

Both the FEM and FDM modeling methods face several issues. A primary issue with the FEM is its high computational cost, which arises from assembling the equations of the individual elements into a global system of equations. The computation time grows rapidly, a problem that becomes particularly pronounced in 3D simulations. The FDM offers a more straightforward discretization

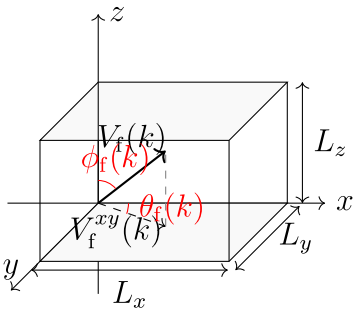


Fig. 1. Illustration of cell dimensions and flow direction in 3D coordinates.

compared to the FEM, but faces limitations in stability. Explicit schemes of the FDM are severely constrained by the CFL conditions, which state that a particle's displacement within one time step must remain within a spatial grid size of the scheme. This requires a smaller time step to maintain stability for the same spatial grid size. Practical implementations often employ stabilization techniques, such as compact schemes (Li et al., 2020) or numerical filters (Adityawan et al., 2023), hereby compromising the method's computational simplicity.

The predictive accuracy of first-principles-based models is highly dependent on key physical parameters. This task is typically formulated as an inverse problem, where parameters are identified by minimizing the difference between model outputs and field observations (Gasca-Ortiz et al., 2021). However, it faces several practical issues. A primary issue lies in the choice of the objective function (Schneider et al., 2020). The conventional method, using the standard Root Mean Square Error (RMSE), is known to be sensitive to outliers and high-magnitude values (Willmott and Matsuura, 2005). In large-scale dispersion scenarios, where contaminant concentration levels can span several orders of magnitude, this sensitivity can introduce a model bias, resulting in poor model performance in regions of low concentration. A further issue arises from the nature of oceanographic data. For instance, models often integrate multi-resolution datasets from various sources with different spatial and temporal resolutions. Moreover, the release of chemical substances along coastal borders, such as from rivers, cities, and industries, can introduce unknown contributions to the boundary conditions.

3. CBM extensions

3.1. Convection process

Building upon the two-dimensional cell-based model (CBM) convection process described in Ma et al. (2025), we extend the formulation to three dimensions. Considering the marine environment as a three-dimensional domain, we discretize it into hyperbox cells with side lengths L_x and L_y in the horizontal plane, and L_z in the vertical direction, and adopt T as the temporal step size.

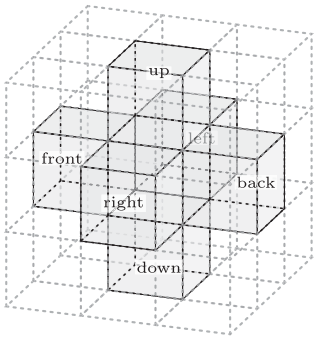
The water flow velocity in each cell is characterized by its magnitude and two angles $\phi_{\mathrm{f},i,j,h}(k)$ and $\theta_{\mathrm{f},i,j,h}(k)$ in spherical coordinates. Specifically, $\phi_{\mathrm{f},i,j,h}(k)$ denotes the angle between the positive z-axis and the velocity vector $V_{\mathrm{f},i,j,h}(k)$, while $\theta_{\mathrm{f},i,j,h}(k)$ is the angle between the positive x-axis and the projection of $V_{\mathrm{f},i,j,h}(k)$ on the xy plane, written as $V^{xy}_{\mathrm{f},i,j,h}(k)$. Accordingly, the three-dimensional flow direction can be fully specified by $(\phi_{\mathrm{f},i,j,h}(k), \theta_{\mathrm{f},i,j,h}(k))$, with $\phi_{\mathrm{f},i,j,h}(k) \in [0,\pi]$ and $\theta_{\mathrm{f},i,j,h}(k) \in [0,2\pi)$. The cell dimensions and flow direction in a cell $C_{i,j,h}$ are illustrated in Fig. 1.

Let $J_{i,j,h}(k)$ be the contaminant concentration in cell $C_{i,j,h}$ at time step k. During the period $[kT,(k+1)T)$, the concentration in $C_{i,j,h}$ is influenced by upstream neighbors contributing inflow of contaminants and downstream neighbors receiving outflow. The concentration update over one time step is

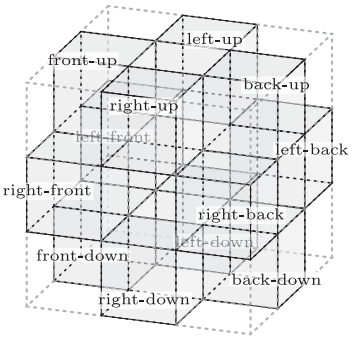
$$\begin{aligned} J_{i,j,h}(k+1) = J_{i,j,h}(k) &+ \sum_{(u,v,w)\in N_{\mathrm{upstream,i,j,h}}(k)} \frac{\alpha^{i,j,h}_{u,v,w}(k)}{L_x L_y L_z} J_{u,v,w}(k) \\ &- \sum_{(u,v,w)\in N_{\mathrm{downstream,i,j,h}}(k)} \frac{\alpha^{i,j,h}_{u,v,w}(k)}{L_x L_y L_z} J_{i,j,h}(k), \end{aligned} \tag{1}$$

where $N_{\mathrm{upstream,i,j,h}}(k)$ and $N_{\mathrm{downstream,i,j,h}}(k)$ are the upstream and downstream neighbor sets, which are determined by the flow direction angles $\phi_{\mathrm{f},i,j,h}(k)$ and $\theta_{\mathrm{f},i,j,h}(k)$. In addition, $\alpha^{i,j,h}_{u,v,w}(k)$ represents the interaction volume terms between cell $C_{i,j,h}$ and neighbor $C_{u,v,w}$. To further compute $\alpha^{i,j,h}_{u,v,w}(k)$, we first decompose $V_{\mathrm{f},i,j,h}(k)$ into its projections along the x, y, and z axes, which yields

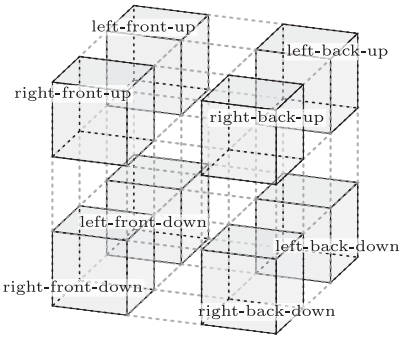
$$\begin{aligned} V^x_{\mathrm{f},i,j,h}(k) &= V_{\mathrm{f},i,j,h}(k)\,\sin(\phi_{\mathrm{f},i,j,h}(k))\,\cos(\theta_{\mathrm{f},i,j,h}(k)) \\ V^y_{\mathrm{f},i,j,h}(k) &= V_{\mathrm{f},i,j,h}(k)\,\sin(\phi_{\mathrm{f},i,j,h}(k))\,\sin(\theta_{\mathrm{f},i,j,h}(k)) \\ V^z_{\mathrm{f},i,j,h}(k) &= V_{\mathrm{f},i,j,h}(k)\,\cos(\phi_{\mathrm{f},i,j,h}(k)). \end{aligned} \tag{2}$$



(a) 6 axial neighbors highlighted.



(b) 12 planar-diagonal neighbors highlighted.



(c) 8 space-diagonal neighbors highlighted.

Fig. 2. Illustration of cell neighbors.

Based on these velocity projections, the interactions between a cell and its neighbors can be determined. Each cell has a potential of 26 neighboring cells, categorized as 6 axial neighbors (front, back, left, right, up, and down), 12 planar-diagonal neighbors (e.g., front-up, back-down, left-up, right-down, front-left, etc.), and 8 space-diagonal neighbors (e.g., front–up–left, back–down–right, etc.), as illustrated in Fig. 2. Driven by a given velocity vector $V_{\mathrm{f},i,j,h}(k)$, the concentration outflow from $C_{i,j,h}$ over one time step is distributed to at most seven downstream neighbors, collected in $N_{\mathrm{downstream},i,j,h}(k)$: three axial neighbors, three planar-diagonal neighbors, and one space-diagonal neighbor. Symmetrically, inflow into $C_{i,j,h}$ comes from the corresponding upstream set $N_{\mathrm{upstream},i,j,h}(k)$.

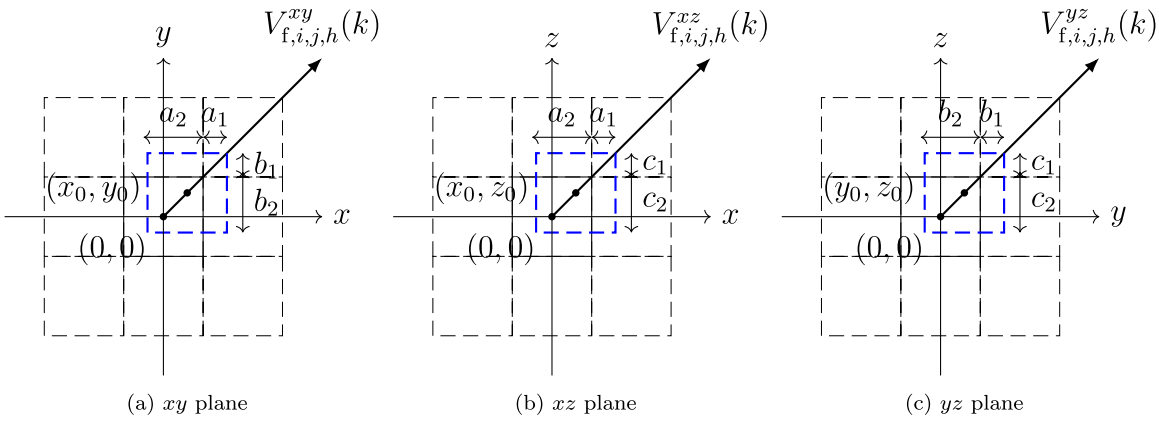
While Fig. 2 provides the three-dimensional neighbor structure, the velocity-driven displacement of the cell center can be more clearly derived through two-dimensional projections. For example, as shown in Fig. 3, the velocity vectors $V^{xy}_{\mathrm{f},i,j,h}(k)$, $V^{xz}_{\mathrm{f},i,j,h}(k)$, and $V^{yz}_{\mathrm{f},i,j,h}(k)$ are constructed as the vector sums of their respective coordinate-axis projections defined in (2). In each projection plane, the relative center point $(0,0)$ of the current cell is displaced to (x_0, y_0) under the influence of the velocity vectors. To ensure this displacement is numerically stable, the Courant–Friedrichs–Lewy (CFL) condition in three dimensions requires that De Moura and Kubrusly (2013)

$$\max\left(\left|TV^{x}_{\mathrm{f},i,j,h}(k)\right|, \left|TV^{y}_{\mathrm{f},i,j,h}(k)\right|, \left|TV^{z}_{\mathrm{f},i,j,h}(k)\right|\right) \le \min(L_x, L_y, L_z). \tag{3}$$

We define the interaction sides a_1, a_2 along the x-axis, b_1, b_2 along the y-axis, and c_1, c_2 along the z-axis, representing the projected movement displacements. Specifically, in the xy plane, the interaction sides are $a_1 = \left|TV^{x}_{\mathrm{f},i,j,h}(k)\right|$, $a_2 = L_x - a_1$, $b_1 = \left|TV^{y}_{\mathrm{f},i,j,h}(k)\right|$, and $b_2 = L_y - b_1$. Similarly, in the z direction, the interaction sides are expressed as $c_1 = \left|TV^{z}_{\mathrm{f},i,j,h}(k)\right|$ and $c_2 = L_z - c_1$.

The interaction volumes are then obtained as the products of these side lengths. For instance, the interaction volumes are $a_1 b_2 c_2$ for the right neighbor cell $(u = i+1, v = j, w = h)$, $a_2 b_1 c_2$ for the front neighbor cell $(u = i, v = j+1, w = h)$, and $a_2 b_2 c_1$ for the up neighbor cell $(u = i, v = j, w = h+1)$. Similarly, the volumes are $a_1 b_1 c_2$ for the right-front neighbor cell $(u = i+1, v = j+1, w = h)$, $a_1 b_2 c_1$ for the right-up neighbor cell $(u = i+1, v = j, w = h+1)$, and $a_2 b_1 c_1$ for the front-up neighbor cell $(u = i, v = j+1, w = h+1)$. In addition, the volume $a_1 b_1 c_1$ corresponds to the right-front-up neighbor cell $(u = i+1, v = j+1, w = h+1)$.

Although the above derivation example assumes a flow velocity with a certain direction, the expressions of a_1, a_2, b_1, b_2, and c_1, c_2 are general, as they depend only on the magnitudes of the displacement vector. The signs of the velocity merely determine the upstream or downstream neighbor sets without altering these absolute values, so the formulation of calculating $\alpha^{i,j,h}_{u,v,w}(k)$ applies to arbitrary flow directions. Moreover, in practice, the neighbor structure may vary at physical boundaries such as the sea surface, seabed, and coastal areas. At the sea surface, the boundary in the z direction, no neighbor exists above. At the seabed, the boundary in the z direction, no neighbor exists below. At the coastal boundaries, cells lack neighbors in the horizontal xy plane on the land side. All of these reduce the number of valid neighbors considered in (1).



(a) xy plane (b) xz plane (c) yz plane

Fig. 3. The concentration movement of the current cell under the influence of flow velocity on the xy, xz, and yz planes.

3.2. Diffusion process

In the diffusion process, contaminants spread from areas of higher concentration to those of lower concentration under the driving force of the entropy increase principle, leading to the spatial expansion of concentration in three dimensions. During the time period $[kT, (k+1)T)$, the concentration in a given cell $C_{i,j,h}$ is influenced by the exchange of contaminants with its neighboring cells, as indicated in Fig. 2.

The exchange of contaminants between cell $C_{i,j,h}$ and its neighbors is driven by the concentration difference, which yields

$$J_{i,j,h}(k+1) = J_{i,j,h}(k) + \sum_{(u,v,w)\in N(i,j,h)} \left(J_{u,v,w}(k) - J_{i,j,h}(k)\right) \frac{\beta^{i,j,h}_{u,v,w}(k)}{L_{\mathrm{d},x} L_{\mathrm{d},y} L_{\mathrm{d},z}}, \tag{4}$$

where $\beta^{i,j,h}_{u,v,w}(k)$ denotes the interaction volume between the current cell $C_{i,j,h}$ and its neighboring cell $C_{u,v,w}$ at time step k. Moreover, $L_{\mathrm{d},x}$, $L_{\mathrm{d},y}$ and $L_{\mathrm{d},z}$ are the side lengths of the cell after expansion in the x, y and z directions, respectively, denoted as

$$\begin{aligned} L_{\mathrm{d},x} &= (1 + T\lambda_{\mathrm{c},xy}(z))L_x \\ L_{\mathrm{d},y} &= (1 + T\lambda_{\mathrm{c},xy}(z))L_y \\ L_{\mathrm{d},z} &= (1 + T\lambda_{\mathrm{c},z}(z))L_z, \end{aligned} \tag{5}$$

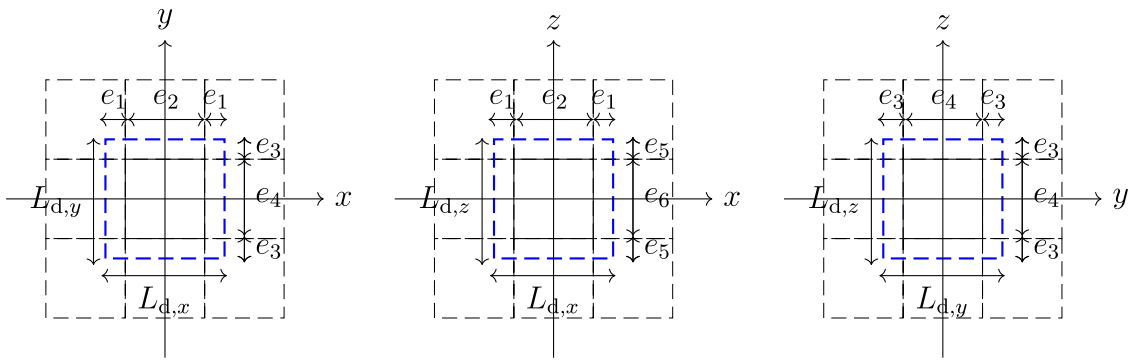
where $\lambda_{\mathrm{c},xy}(z)$ and $\lambda_{\mathrm{c},z}(z)$ represent the depth-dependent diffusion coefficients of the CBM in the horizontal and vertical directions, respectively, which decrease with depth increase (Canuto et al., 2011; Ferreira et al., 2005). We denote the values of $\lambda_{\mathrm{c},xy}$ and $\lambda_{\mathrm{c},z}$ at the surface of the computational domain by $\lambda_{\mathrm{c},xy}(0)$ and $\lambda_{\mathrm{c},z}(0)$, respectively, and their decay rates by H_{xy} and H_z. Accordingly, the diffusion coefficients are then parameterized as

$$\begin{aligned} \lambda_{\mathrm{c},xy}(z) &= \lambda_{\mathrm{c},xy}(0)e^{-\frac{z}{H_{xy}}} \\ \lambda_{\mathrm{c},z}(z) &= \lambda_{\mathrm{c},z}(0)e^{-\frac{z}{H_z}}. \end{aligned} \tag{6}$$

The derivation of the interaction volumes $\beta^{i,j,l}_{u,v,w}$ is performed first by projecting the three-dimensional space onto two dimensions, as illustrated in Fig. 4. The interaction side lengths are denoted as e_1 and e_2 in the x direction, e_3 and e_4 in the y direction, and as e_5 and e_6 in the z direction. Specifically, e_1, e_3, and e_5 are derived from the diffusion coefficients of the CBM, while e_2, e_4, and e_6 correspond to the side lengths of the hyperbox cell in the x, y, and z directions, respectively, which yields

$$\begin{aligned} e_1 &= TL_x\lambda_{\mathrm{c},xy} \\ e_2 &= L_x \\ e_3 &= TL_y\lambda_{\mathrm{c},xy} \\ e_4 &= L_y \\ e_5 &= TL_z\lambda_{\mathrm{c},z} \\ e_6 &= L_z. \end{aligned} \tag{7}$$

Then, $\beta^{i,j,h}_{u,v,w}$ is calculated from the products of the side lengths. Specifically, the expression $e_1e_4e_6$, $e_3e_2e_6$, and $e_5e_2e_4$ corresponds to the interaction volumes with the right/left, front/back, and up/down neighboring cells, with indices $(u = i \pm 1, v = j, w = h)$ and $(u = i, v = j \pm 1, w = h)$, and $(u = i, v = j, w = h \pm 1)$, respectively. Moreover, The expression $e_1e_3e_6$, $e_1e_5e_4$, and $e_3e_5e_2$ correspond to the interaction volumes with the right-front, left-front, right-back, and left-back cells in the xy plane with indices $(u = i \pm 1, v = j \pm 1, w = h)$; with the right-up, right-down, left-up, and left-down cells in the xz plane with indices



(a) xy plane (b) xz plane (c) yz plane

Fig. 4. The current cell expansion under the influence of concentration on the xy, xz, and yz planes.

$(u = i \pm 1, v = j, w = h \pm 1)$; and with the front-up, front-down, back-up, and back-down cells in the yz plane with indices $(u = i, v = j \pm 1, w = h \pm 1)$, respectively. Finally, the expression $e_1 e_3 e_5$ corresponds to the interaction volumes with the neighboring cells in which all three indices change simultaneously, i.e., up-right-front, up-right-back, up-left-front, up-left-back, down-right-front, down-right-back, down-left-front, and down-left-back, with indices $(u = i \pm 1, v = j \pm 1, w = h \pm 1)$.

3.3. Chemical reaction process

In addition to flow-driven convection and concentration gradient-driven diffusion, chemical reaction is also an essential component of contaminant transport in the ocean, particularly through photodegradation (Baena-Nogueras et al., 2017). We incorporate this reaction process into the CBM framework as follows. Specifically, contaminants are modeled to degrade under external influences such as sunlight. This process is purely local, without interactions with neighboring cells, and is therefore directly represented as a decay term within each cell, which yields

$$J_{i,j,h}(k+1) = J_{i,j,h}(k) - \mu(z)\, J_{i,j,h}(k), \tag{8}$$

where $\mu(z) = T\mu_{\text{photo}}(z)$ is the dimensionless decay factor that represents the fraction of contaminant degraded within one time step at depth z. Moreover, $\mu_{\text{photo}}(z)$ denotes the photodegradation rate, which is assumed to decrease with depth, expressed as

$$\mu_{\text{photo}}(z) = \mu_{\text{photo}}(0)\, e^{-\frac{z}{H_{\text{photo}}}}, \tag{9}$$

where $\mu_{\text{photo}}(0)$ represents the photodegradation rate at the surface of the computational domain, and H_{photo} is the decay rate.

The photodegradation process described above is a specific type of chemical reaction. It can be represented within a broader chemical reaction framework. More generally, a chemical reaction can be expressed as (Atkins et al., 2023)

$$n_A A + n_B B \longrightarrow n_C C + n_D D, \tag{10}$$

where A and B denote the reactants, and C and D denote the transformation products. The parameters n_A, n_B, n_C, and n_D are the stoichiometric coefficients of the substances A, B, C, and D, respectively. In the present study, we denote the concentration of the contaminant by J. Accordingly, the general reactant A in (10) corresponds to J in our formulation.

In general, environmental factors (e.g., temperature, pressure, or sunlight intensity) can also influence the reaction process. However, for the sake of simplicity, these factors are not explicitly included in the general form of (10). The extension is straightforward, and the photodegradation process above provides an illustration of including the effect of sunlight on the reaction rate.

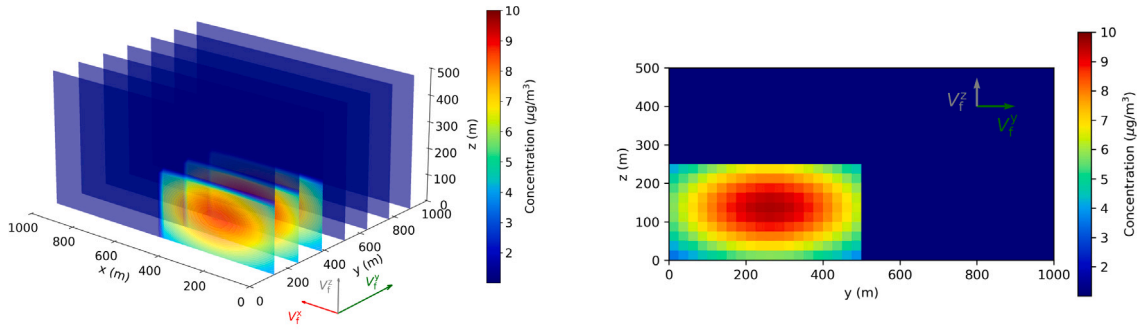
Assuming that reactant B is present in sufficient quantities,[1] the definition of the reaction rate is

$$v = -\frac{1}{n_A}\frac{dJ}{dT}. \tag{11}$$

On the other hand, the reaction rate can also be determined experimentally and is commonly expressed by a rate law. Under the pseudo-first-order approximation, it takes the following form (Atkins et al., 2023):

$$v = k_{\text{r,eff}} J, \tag{12}$$

[1] If B is not present in sufficient quantities, the model has to be extended to also include the concentration of B in the same way as we model the concentration of A.



(a) 3D view (b) At $x = 200$ m, yz plane view

Fig. 5. Initial concentration distribution at $t = 0$ s.

where $k_{\mathrm{r,eff}}$ is the effective rate constant, given by $k_{\mathrm{r,eff}} = k_{\mathrm{r}}[B]_0^2$. Here, k_{r} is the intrinsic rate constant for the reaction, dependent on the temperature, and $[B]_0$ represents the initial concentration value of B.

Combining the definition of the reaction rate (11) and the rate law (12), we get

$$\frac{dJ}{dT} = -n_A k_{r,\mathrm{eff}} J. \tag{13}$$

By discretizing (13) by the forward Euler scheme, the concentration update for J is given by

$$J(k+1) = J(k) - \mu_{\mathrm{reaction}} J(k), \tag{14}$$

where $\mu_{\mathrm{reaction}} = T n_A k_{\mathrm{r,eff}}$, represents the consumption rate.

4. Simulation study of the three-dimensional CBM

4.1. Case study of transport process

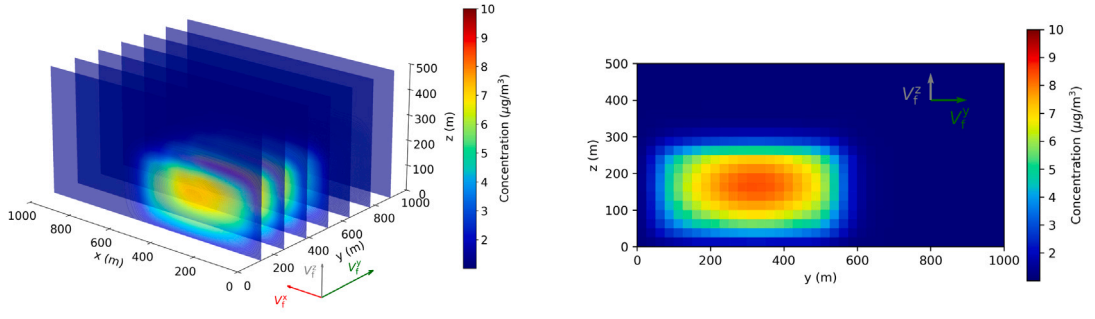
To demonstrate the applicability of the proposed three-dimensional (3D) CBM scheme, we conduct a simulation study of contaminant transport, taking the convection process as a representative case. First, the transport process is simulated based on the CBM scheme implemented in Python, then the resulting solutions are compared with those of a 3D stabilized FEM solver, which is implemented in Python-COMSOL.

The physical domain is defined as a hyperbox of 1000 m × 1000 m × 500 m, with the background concentration set to 1.1 μg/m^3. The initial contaminant follows a Gaussian distribution within a subregion of 500 m × 500 m × 250 m, with a peak concentration of 10 μg/m^3, located near the origin $(0, 0, 0)$ at the down-right-front corner of the computational domain. The water flow projection in each axis is $V_{\mathrm{f}}^x = 1$ m/s, $V_{\mathrm{f}}^y = 1$ m/s, and $V_{\mathrm{f}}^z = 0.5$ m/s, respectively. As shown in Fig. 5, the initial concentration field is shown in a 3D view and in the yz-plane view at $x = 0$ m. In addition, the colored arrows indicate the flow directions: red for the V_{f}^x, green for the V_{f}^y, and blue for the V_{f}^z.

The simulation is performed for a duration of 60 s, with a time step of $T = 1$ s and spatial resolutions of $L_x = L_y = L_z = 25$ m. The evolution of the contaminant concentration field is illustrated in Fig. 6, where the contaminant expands outward from the initial region under the effects of convection. Moreover, the process is simulated using FEM with the same physical parameters, and the corresponding solution is shown in Fig. 7.

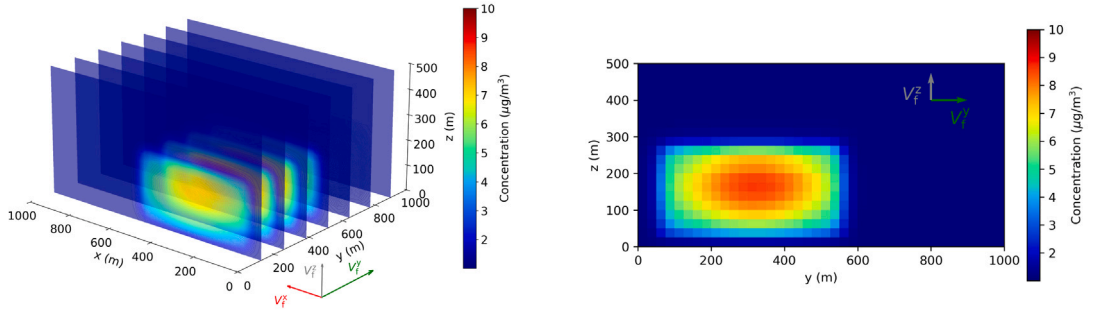
For the FEM benchmark solution (COMSOL via Python interface), we employ a structured hexahedral (brick) mesh with first-order (i.e., linear) elements. To apply the same background boundary treatment as in the CBM, the FEM computational domain is extended by a one-cell buffer layer, resulting in a mesh of 41 × 41 × 21 hexahedral elements and 42 × 42 × 22 nodes (mesh statistics reported by COMSOL). Importantly, the quantitative comparisons and visualizations are evaluated on the interior physical domain 1000 m × 1000 m × 500 m, excluding the buffer layer. The time-dependent FEM solver uses an implicit Backward Differentiation Formula (BDF) scheme with strict stepping constrained by $T = 1$ s and a relative tolerance of 0.005, together with a stabilized convection–diffusion formulation built in COMSOL.

The overall relative RMSE value between the CBM and FEM at $t = 60$ s is 7.67%. In addition, we also record the CPU time; the experiments were all run on a Lenovo ThinkPad P16 Gen 2 with an Intel Core i9-13950HX. The CBM required 0.06 s of CPU time, whereas the FEM required 14.72 s.



(a) 3D view (b) At $x = 200$ m, yz plane view

Fig. 6. Concentration evolution of the CBM at $t = 60$ s.



(a) 3D view (b) At $x = 200$ m, yz plane view

Fig. 7. Concentration evolution of the FEM at $t = 60$ s.

Table 1
Comparison of the CBM and the FEM solutions at representative depths.

	$z = 0$ m	$z = 50$ m	$z = 100$ m
CBM solution	0.3610	1.0918	1.0999
FEM solution	0.3739	1.0922	1.0999
Relative RMSE	3.45%	0.04%	0.00%

4.2. Validation of the chemical reaction process

Since the chemical reaction module is formulated as a purely local decay process without inter-cell interactions, we validate it in an independent reaction-only simulation study. Specifically, we consider photodegradation as a representative reaction process, where the reaction rate decreases exponentially with depth, as defined in (9). For this validation, we use a stabilized time-dependent FEM solution computed in Python-COMSOL as the benchmark, and compare it with the CBM solution from our Python implementation. Both solvers use identical geometry, parameters, initial conditions, and output times.

The computational domain is a hyperbox of $1\text{ m} \times 1\text{ m} \times 100\text{ m}$, initialized with a uniform concentration $J_0 = 1.1\ \mu\text{g/m}^3$. We set $\mu_{\text{photo}}(0) = 5.56 \times 10^{-3}\ \text{s}^{-1}$ and $H_{\text{photo}} = 10\,\text{m}$, and run the simulation over $[0, 200]\,\text{s}$ with a time step $T = 1$ s and spatial resolutions $L_x = L_y = L_z = 1$ m. We compare the CBM and the FEM concentration values at representative depths in the shallow ($z = 0\,\text{m}$), intermediate ($z = 50\,\text{m}$), and deep ($z = 100\,\text{m}$) layers, and calculate their relative RMSE, as shown in Table 1. The slightly higher relative error at the surface ($z = 0\,\text{m}$) is expected, as the degradation rate is strongest there, leading to larger concentration changes and accumulated error. Overall, it validates the accuracy of the proposed CBM scheme for simulating chemical reactions.

(a) CBM grid configuration (b) FDM grid configuration

Fig. 8. Coarser and finer grid configurations.

5. Improved framework for relating the diffusion coefficients

The diffusion parameter λ_c in the CBM is not directly equivalent to the PDE coefficient λ_f. To ensure physical consistency, we establish a systematic relation between λ_c and λ_f. In our previous work (Ma et al., 2025), two approaches were introduced to identify this relation. While these approaches provided an initial framework, their applicability is limited in more general scenarios. One strategy, based on equal grid configurations, only considers the horizontal and vertical neighbors, which leads to some loss of precision. Another strategy relies on identification from simulation data, which achieves higher accuracy but requires higher computational cost.

To overcome these limitations and to enhance the robustness of the strategies, we here propose an alternative formulation based on finer grid configurations, where diagonal neighbors are also incorporated in the discretization. The three strategies will be systematically compared in a simulation example to analyze their performance from the accuracy and computational efficiency perspectives.

5.1. Relation under finer grid configurations

For the sake of simplicity, we consider the two-dimensional case in this section. However, note that the approach can also be extended to the three-dimensional case.

To capture more detailed dynamic information, we set the spatial and temporal steps as $L_c = 2L_f$ and $T_c = 2T_f$, where the subscripts c and f denote the CBM and FDM discretizations, respectively. In the CBM, we assume that $C_{i,j}$'s orthogonal neighbors ($C_{i+1,j}$, $C_{i-1,j}$, $C_{i,j+1}$, and $C_{i,j-1}$) share the same concentration $J^{\text{orth}}_{i,j}$, while diagonal neighbors ($C_{i+1,j+1}$, $C_{i-1,j+1}$, $C_{i+1,j-1}$, and $C_{i-1,j-1}$) are assigned a common value $J^{\text{diag}}_{i,j}$, as illustrated in 8(a). In the FDM, each coarse CBM cell is represented by a 2×2 block of finer cells, where the four cells in the block are assumed to share the same concentration value as their CBM counterpart, as shown in Fig. 8(b).

For the diffusion component of the CBM, based on the two-dimensional specialization of (4), the CBM concentration update at cell $C_{i,j}$ over one coarse time step T_c is expressed as

$$\begin{aligned} J(k+1) &= J(k) + 4m_1\left(J^{\text{orth}}(k) - J(k)\right) + 4m_2\left(J^{\text{diag}}(k) - J(k)\right) \\ &= (1 - 4m_1 - 4m_2)J(k) + 4m_1 J^{\text{orth}}(k) + 4m_2 J^{\text{diag}}(k). \end{aligned} \tag{15}$$

The coefficients m_1 and m_2 represent the concentration contributions from orthogonal and diagonal neighbors, respectively, which yields

$$m_1 = \frac{T_c\lambda_c}{(1+T_c\lambda_c)^2}, \qquad m_2 = \left(\frac{T_c\lambda_c}{1+T_c\lambda_c}\right)^2. \tag{16}$$

For the FDM, we first recall the general discretization form of the diffusion component, which yields

$$\begin{aligned} J_{i,j}(k+1) = J_{i,j}(k) + \lambda_f\left(\frac{T_f}{L_f^2}\right)(J_{i-1,j}(k) + J_{i+1,j}(k) \\ + J_{i,j-1}(k) + J_{i,j+1}(k) - 4J_{i,j}(k)). \end{aligned} \tag{17}$$

Based on (17), for the central cell in Fig. 6(b), after one time step T_f, its concentration becomes

$$J(k+1) = J(k) + 2n(J^{\text{orth}}(k) - J(k)), \tag{18}$$

where $n = \lambda_f \frac{T_f}{L_f^2}$. This expression can be equivalently rewritten as $n = 2\lambda_f \frac{T_c}{L_c^2}$, which serves as a simplified notation in the following derivations.

Since the concentration update of the central cell at the second time step requires its orthogonal neighbors' value from the first step, we also compute their evolution, which yields

$$J^{\mathrm{orth}}(k+1) = J^{\mathrm{orth}}(k) + n(J^{\mathrm{diag}}(k) - J^{\mathrm{orth}}(k)) + n(J(k) - J^{\mathrm{orth}}(k)). \tag{19}$$

Then, proceeding to the second time step, the concentration of the central cell is updated as

$$J(k+2) = J(k+1) + 2n(J^{\mathrm{orth}}(k+1) - J(k+1)). \tag{20}$$

Substituting the expressions of $J(k+1)$ from (18) and $J^{\mathrm{orth}}(k+1)$ from (19) into (20), we obtain

$$J(k+2) = \left((1-2n)^2 + 2n^2\right)J(k) + 2n^2 J^{\mathrm{diag}}(k) + 4n(1-2n)J^{\mathrm{orth}}(k). \tag{21}$$

By equating the coefficients of corresponding terms in (15) and (21), the following relations between λ_{f} and λ_{c} are obtained:

$$1 - 4m_1 - 4m_2 = (1-2n)^2 + 2n^2, \tag{22}$$

$$4m_2 = 2n^2, \tag{23}$$

$$4m_1 = 4n(1-2n). \tag{24}$$

After simplifying (22), (23) and (24), respectively, the following expressions are obtained:

$$m_1 + m_2 = n - \frac{3}{2}n^2, \tag{25}$$

$$m_2 = \frac{1}{2}n^2, \tag{26}$$

$$m_1 = n - 2n^2. \tag{27}$$

Since (25) is derived from (26) and (27), the subsequent analysis focuses on solving (26) and (27) to establish the relationship between λ_{c} and λ_{f}. We present four approaches: direct transformation, solving a quadratic equation, averaging the root values, and minimizing the total error.

Direct transformation

Substituting the expressions for m_2 and n into (26), we obtain

$$\left(\frac{T_{\mathrm{c}}\lambda_{\mathrm{c}}}{1+T_{\mathrm{c}}\lambda_{\mathrm{c}}}\right)^2 = \frac{1}{2}(2\lambda_{\mathrm{f}}\frac{T_{\mathrm{c}}}{L_{\mathrm{c}}^2})^2. \tag{28}$$

Rearranging the factors in (28) and simplifying, the relationship between λ_{c} and λ_{f} is derived as

$$\lambda_{\mathrm{c}} = \frac{2\lambda_{\mathrm{f}}}{\sqrt{2}L_{\mathrm{c}}^2 - 2\lambda_{\mathrm{f}}T_{\mathrm{c}}} =: \lambda_{\mathrm{c}}^{\mathrm{direct}}. \tag{29}$$

Solving a quadratic equation

Substituting the expressions of m_1 and n into (27), and introducing $n - 2n^2 = \gamma_1$ to simplify notation, we have

$$\frac{T_{\mathrm{c}}\lambda_{\mathrm{c}}}{(1+T_{\mathrm{c}}\lambda_{\mathrm{c}})^2} = \gamma_1. \tag{30}$$

Rearranging and simplifying the above equation, we obtain the following quadratic equation with respect to λ_{c}:

$$\gamma_1 T_{\mathrm{c}}^2\lambda_{\mathrm{c}}^2 + (2\gamma_1 T_{\mathrm{c}} - T_{\mathrm{c}})\lambda_{\mathrm{c}} + \gamma_1 = 0. \tag{31}$$

By solving this quadratic equation, we derive the relationship between λ_{c} and γ_1, where γ_1 involves λ_{f}. Through the quadratic formula and considering the physically meaningful positive root, we obtain

$$\lambda_{\mathrm{c}} = \frac{1 - 2\gamma_1 \pm \sqrt{1-4\gamma_1}}{2\gamma_1 T_{\mathrm{c}}} =: \lambda_{\mathrm{c}}^{\mathrm{quadratic}}. \tag{32}$$

Averaging the root values

The relation between λ_{c} and λ_{f} in is obtained from different derivation approaches. Averaging these roots yields a new solution value of λ_{c}, given as

$$\lambda_{\mathrm{c}}^{\mathrm{avg}} = \frac{\lambda_{\mathrm{c}}^{\mathrm{direct}} + \lambda_{\mathrm{c}}^{\mathrm{quadratic}}}{2}. \tag{33}$$

Minimizing the total error

By minimizing the total error of (26) and (27), we obtain the minimization problem with respect to λ_c. To simplify the calculation, we introduce $\gamma_2 = \frac{n^2}{2}$, which leads to

$$\min_{\lambda_c}((m_2 - \gamma_2)^2 + (m_1 - \gamma_1)^2). \tag{34}$$

Taking the derivative of (34) with respect to λ_c, and setting it to zero, we obtain

$$(m_2 - \gamma_2)\frac{dm_2}{d\lambda_c} + (m_1 - \gamma_1)\frac{dm_1}{d\lambda_c} = 0, \tag{35}$$

where the derivatives $\frac{dm_2}{d\lambda_c}$ and $\frac{dm_1}{d\lambda_c}$ are given as

$$\begin{aligned} \frac{dm_2}{d\lambda_c} &= \frac{2T_c^2\lambda_c}{(1+T_c\lambda_c)^3}, \\ \frac{dm_1}{d\lambda_c} &= \frac{T_c(1-T_c\lambda_c)}{(1+T_c\lambda_c)^3}. \end{aligned} \tag{36}$$

Substituting the expressions of the above derivatives, m_1 and m_2 into (35), we get

$$\left(\frac{(T_c\lambda_c)^2}{(1+T_c\lambda_c)^2} - \gamma_2\right)\frac{2T_c^2\lambda_c}{(1+T_c\lambda_c)^3} + \left(\frac{T_c\lambda_c}{(1+T_c\lambda_c)^2} - \gamma_1\right)\frac{T_c(1-T_c\lambda_c)}{(1+T_c\lambda_c)^3} = 0. \tag{37}$$

Eliminating the denominator and rearranging the terms of the above equation, yields

$$2\left((T_c\lambda_c)^2 - \gamma_2(1+T_c\lambda_c)^2\right)T_c^2\lambda_c + \left(T_c\lambda_c - \gamma_1(1+T_c\lambda_c)^2\right)\left(T_c - T_c^2\lambda_c\right) = 0. \tag{38}$$

Finally, after simplifying, we obtain a cubic equation with respect to λ_c as follows:

$$(2 - 2\gamma_2 + \gamma_1)T_c^3\lambda_c^3 + (-4\gamma_2 + \gamma_1 - 1)T_c^2\lambda_c^2 + (-2\gamma_2 - \gamma_1 + 1)T_c\lambda_c - \gamma_1 = 0. \tag{39}$$

5.2. Case study

We conduct numerical experiments under the three strategies to validate the performance of the proposed framework for linking λ_c and λ_f. Specifically, Approach 1, based on an equal grid configuration, and Approach 3, which relies on identification from simulation data, were originally introduced in our previous work (Ma et al., 2025). Approach 2, the finer grid configuration that incorporates diagonal neighbors into the discretization, is newly proposed in this paper. For Approach 2, Approach 2.1, Approach 2.2, Approach 2.3, and Approach 2.4 refer to the methods presented earlier in this section: direct transformation, solving a quadratic equation, averaging the root values, and minimizing the total error, respectively.

A broad range of diffusion coefficients λ_f is investigated, ranging from $1.00 \cdot 10^{-3}\,\mathrm{m}^2/\mathrm{s}$ to $10\,\mathrm{m}^2/\mathrm{s}$. The simulations are performed in a $1000\,\mathrm{m} \times 1000\,\mathrm{m}$ square domain, with a uniform initial background concentration of $1.1\,\mu\mathrm{g}/\mathrm{m}^3$. A Gaussian contaminant source is located at the center, covering $500\,\mathrm{m} \times 500\,\mathrm{m}$, and peaking at $10\,\mu\mathrm{g}/\mathrm{m}^3$. The CBM is discretized with a spatial resolution of $L_c = 10\,\mathrm{m}$ and a temporal resolution of $T_c = 1\,\mathrm{s}$, and each simulation runs for 600 s. For a more accurate reference, the FDM is solved on a finer grid with $L_f = 5\,\mathrm{m}$ and time step size $T_f = 0.5\,\mathrm{s}$, and mapped to the CBM grid by spatial averaging.

Table A.1 in the Appendix shows the RMSE of the three strategies over a wide range of λ_f. The identification-based strategy yields the lowest RMSE in all cases. Between the two discretization-based strategies, the finer-grid formulation (Approach 2) performs better at smaller λ_f, whereas the equal-grid formulation (Approach 1) becomes slightly more accurate at larger λ_f. In general, the finer-grid formulation (Approach 2) is preferable in cases with sharper gradients of the concentration level, while the equal-grid formulation (Approach 1) is more suitable for shallow gradient scenarios. The identification-based (Approach 3) consistently gives the highest accuracy, with a higher computational cost.

6. Parameter identification in the CBM framework

In practice, the diffusion coefficient is often difficult to measure, as it is influenced by spatiotemporal intermittency and turbulence arising from multiple interacting processes across different scales. The diffusion coefficient is assumed to be constant in both time and space (Olivé Abelló et al., 2023). Fortunately, ocean current data are widely available through various techniques, such as satellite sensing (Dohan, 2017), surface drifters (Aksamit et al., 2019), and acoustic Doppler instruments (Mercier et al., 2021). Given this data availability, in the CBM framework, the diffusion coefficient is regarded as an unknown physical parameter, whereas the ocean current is treated as a known input. In this section, we identify this parameter by minimizing the difference between the CBM model outputs and the observed concentration data based on the least-squares method, thereby validating the CBM on real oceanographic datasets.

Table 2
Overall prediction errors on training and test datasets.

Solver	Dataset	Absolute RMSE	Relative RMSE (%)
CBM	Training	0.0331	7.58%
	Test	0.0340	7.71%

6.1. Case study: Parameter identification from copernicus dataset

The ocean current (Lellouche et al., 2018) and nitrate concentration data (Cossarini et al., 2021) of the Mediterranean Sea used in this study are extracted from the Copernicus Marine Environment Monitoring Service, and available at https://doi.org/10.48670/moi-00016 and https://doi.org/10.25423/cmcc/medsea_multiyear_bgc_006_008_medbfm3 , respectively. The parameter identification process consists of three main stages: data preprocessing, least-squares-based estimation, and forward model evaluation.

6.1.1. Data preprocessing

Since the original spatial resolutions of the ocean current and concentration datasets differ (8 km $\times$ 8 km versus 4 km $\times$ 4 km, respectively), we apply bilinear interpolation (Kirkland, 2010) to resample the ocean current data onto a 4 km $\times$ 4 km grid, ensuring consistency. Similarly, to satisfy the CFL condition, we employ linear interpolation to align the temporal resolutions of both datasets to hourly intervals.

To ensure that parameter estimation is restricted to reliable regions, we apply two filtering operations to the original M $\times$ N matrix datasets, which include both land and ocean regions. First, an ocean mask operator is used to exclude land, retaining only valid ocean grid points. Second, a 15-cell buffer zone is removed along the domain boundaries to mitigate uncertain boundary effects.

Finally, four days of data (96 time steps) are used for training, with the initial condition (nitrate concentration) reset at the start of each day. The subsequent day (24 time steps) serves as the test set, as a single day captures a complete tidal cycle. Moreover, this daily reset also helps limit the accumulation of numerical errors.

6.1.2. Parameter identification via least squares

We formulate the estimation of the diffusion coefficient as a parameter optimization problem, which is solved using a nonlinear least-squares approach. Specifically, we define the objective function as the sum of squared relative differences between the CBM-simulated concentrations and the observed data across all valid ocean grid points and time steps. Using a relative, rather than an absolute, difference is a critical choice for handling concentration values that span several orders of magnitude, as it ensures that the parameter identification is not biased by high-concentration regions. More specifically, the optimization problem is formulated as

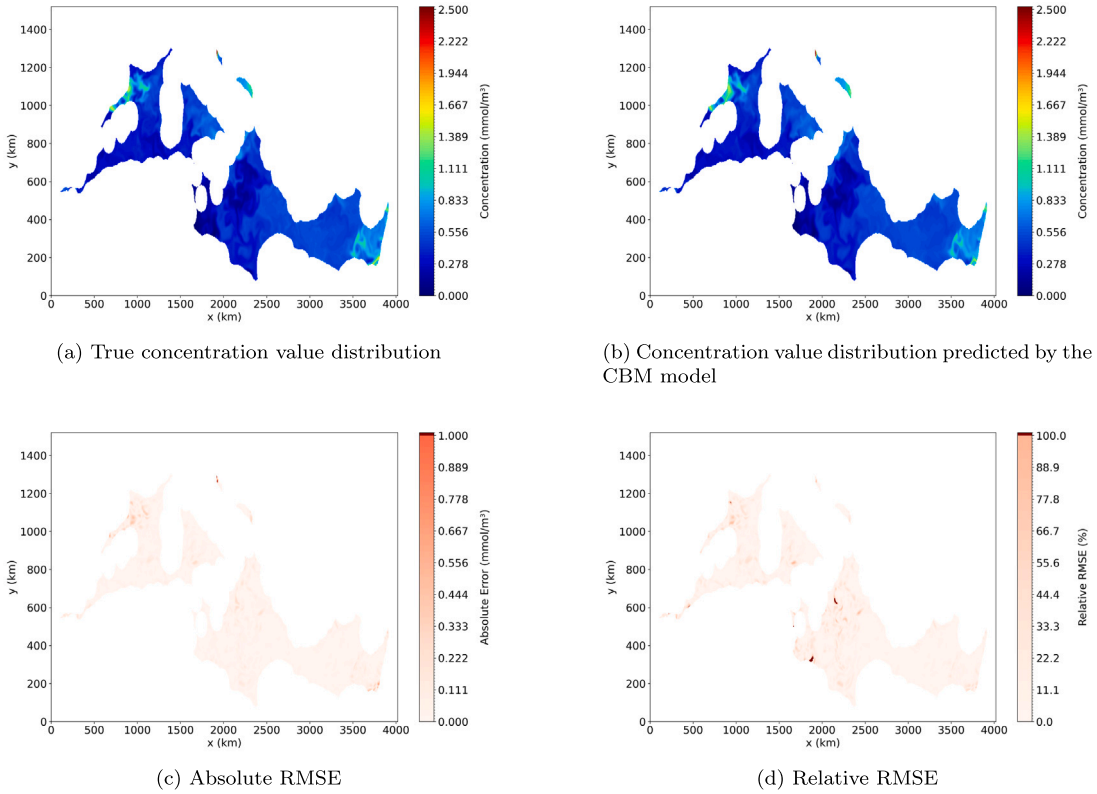
$$\min_{\lambda_c} \sum_{k=1}^{N} \sum_{(i,j)\in G} \left(\frac{J_{\mathrm{CBM},i,j}(k,\lambda_c) - J_{\mathrm{Copernicus},i,j}(k)}{\sqrt{J_{\mathrm{Copernicus},i,j}(k)^2 + \epsilon}} \right)^2, \tag{40}$$

where G denotes the set of valid ocean grid points, ϵ is a small positive constant to avoid division by zero, $J_{\mathrm{Copernicus},i,j}(k)$ represents the observed nitrate concentration at location (i,j) and time step k, extracted from the Copernicus dataset, and $J_{\mathrm{CBM},i,j}(k,\lambda_c)$ denotes the corresponding simulated concentration computed by the CBM, based on three inputs: the ocean current field $V_{\mathrm{Copernicus},i,j}(k)$, the initial nitrate concentration $J_{\mathrm{Copernicus},i,j}(0)$, and the diffusion coefficient λ_c.

6.1.3. Forward prediction and performance evaluation

After identifying the optimized diffusion coefficient, we assess model performance by comparing its forward simulation with observed nitrate concentrations in the test dataset. This evaluation includes both qualitative and quantitative perspectives. To examine the model's spatial prediction accuracy, we compare the predicted and observed nitrate concentration fields at the final time step of the test set, illustrated in Fig. 9. To further quantify model performance, we compute metrics over the entire spatiotemporal domain, as shown in Table 2.

As shown in Fig. 9, the localized relative RMSE values reaching 100% primarily occur in low-concentration regions, where the ground truth concentration is close to zero. In these areas, the relative metric becomes highly sensitive, i.e., a small absolute error can result in a large relative percentage due to the small denominator. For example, at the location ($x = 1676$ km, $y = 476$ km), the true value is $J = 0.006$ mmol/m^3 and predicted value is $\hat{J} = 0.013$ mmol/m^3. The absolute error is only 0.007 mmol/m^3, while the relative error exceeds 100%. In addition, some localized relative RMSE values are observed near the physical domain boundaries, where boundary-related uncertainties have a strong influence. For instance, at ($x = 3792$ km, $y = 176$ km), which lies at the edge



(a) True concentration value distribution

(b) Concentration value distribution predicted by the CBM model

(c) Absolute RMSE

(d) Relative RMSE

Fig. 9. Concentration prediction at final test time step.

of the domain along the y-axis, the true value is $J = 0.638$ mmol/m^3, while the predicted value is $\hat{J} = 1.401$ mmol/m^3, indicating uncertain cross-boundary exchange.

In addition to the accuracy, we also record the CPU time to highlight the efficiency of the proposed framework. In this Copernicus case study, the computational domain is represented by a 380×1005 grid, yielding 381 900 total grid points. After land–sea masking, 140 888 points are valid ocean grid points (36.89% coverage), and the CBM updates are performed on these ocean points. The evaluation covers 120 time steps (96 for training and 24 for testing). The experiment was run on a Lenovo ThinkPad P16 Gen 2 with an Intel Core i9-13950HX. The CBM completes the forward simulation in 4.56 s for the training period and 1.09 s for the test period. This demonstrates the computational efficiency of the proposed framework for practical oceanographic applications.

7. Discussion

In this section, we discuss the limitations of the proposed CBM and potential directions for improvement. Despite the favorable performance demonstrated in the current study, several limitations remain for broader application.

First, regarding the parameter identification of the diffusion coefficient, sensitivity experiments show that removing the diffusion term in Section 6 alters the test dataset RMSE by only 0.02 percentage points (from 7.71% to 7.73%), indicating that the study case is convection-dominated. Thus, a constant diffusion coefficient is a robust and sufficient choice for the current setting. However, in scenarios where diffusion plays a more dominant role, a spatiotemporally varying diffusion coefficient is necessary. In such cases, deep learning approaches could be employed to learn the spatiotemporal variation of the coefficient (Bolton and Zanna, 2019), which we plan to explore in future work.

Second, concerning the model's applicability to more complex scenarios, the current framework is validated using nitrate as a representative tracer. When dealing with more complex contaminants, additional processes such as sedimentation may need to be incorporated. In addition, in highly heterogeneous environments, spatially varying diffusion coefficients may be necessary to better capture local variability and improve prediction accuracy. These represent important directions for future work.

Finally, with respect to boundary handling strategies in coastal regions, during data preprocessing, a 15-cell buffer zone along the domain boundaries is removed to mitigate boundary effects. One reason is that transport processes across domain boundaries

are not fully constrained by the available data. Another reason is that contaminant dynamics in coastal regions are often more complex than in open water, involving factors such as localized source releases and tides. Therefore, for future operational coastal applications, a finer nearshore resolution, together with data assimilation using near-real-time observations to better constrain nearshore dynamics (Chen et al., 2025), will be considered rather than simply excluding these regions.

8. Conclusions and future work

We have proposed a generalized three-dimensional cell-based framework for contaminant transport modeling in marine environments, demonstrating a favorable balance between accuracy and computational efficiency. This framework comprehensively models the main physical processes, including convection, diffusion, and chemical reactions. Comparative simulation studies show that the framework achieves higher computational speed than a traditional finite-element method (FEM) solver, while the concentration evolution difference remains within an acceptable range. Additionally, the discretization-based strategy employs a finer grid configuration to physically link the diffusion coefficient formulation of the cell-based model (CBM) with the finite-difference method (FDM). A comparative simulation study demonstrates that this new formulation is more suitable than existing approaches for scenarios with sharper gradients of the concentration level. Together with existing equal-grid and identification-based strategies, they ensure applicability across different concentration field conditions. Furthermore, for parameter identification from real oceanographic data, we implement a systematic data preprocessing and parameter identification workflow, which is validated using ocean current and nitrate concentration data from the Copernicus dataset for the Mediterranean Sea. The validation results demonstrate the framework's potential for practical deployment in operational marine pollution assessment.

Expanding upon this work, future directions will investigate parameter identification of a spatiotemporally varying diffusion coefficient, improved coastal boundary handling, and extensions to more complex contaminants and processes. Moreover, building on the CBM framework, our future research will focus on the challenge of sparse observations in marine environments by developing a physics-informed neural network (PINN) approach. This approach will integrate deep learning with first-principles numerical models, such as our CBM or a traditional FDM.

CRediT authorship contribution statement

Ying Ma: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Meichen Guo:** Writing – review & editing, Supervision, Conceptualization. **Bart De Schutter:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Code availability

The source code for the proposed CBM model, along with the validation scripts used in this study (including the chemical reaction module and the FEM benchmark), is available in the GitHub repository at: https://github.com/ONE-BLUE-EU/cbm-marine-dispersion-3d/.

Acknowledgments

Funded by the European Union under grant agreement 101134929 (ONE-BLUE). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

See Table A.1.

Table A.1
Related λ_c values and absolute RMSE for different strategies and λ_f values. *Note: The bold font indicates the λ_c value that yields the lowest RMSE between Approach 1 and 2.*

$\lambda_f(m^2/s)$	Approach	$\lambda_c(1/s)$	RMSE
(a) λ_f from 1.0×10^{-3} to 7.85×10^{-2}			
$1.0000 \cdot 10^{-3}$	Approach 1	$1.0000 \cdot 10^{-5}$	$4.8143 \cdot 10^{-3}$
	Approach 2.1	$1.4142 \cdot 10^{-5}$	$2.7848 \cdot 10^{-3}$
	Approach 2.2	$2.0000 \cdot 10^{-5}$	$1.5841 \cdot 10^{-4}$
	Approach 2.3	$1.7071 \cdot 10^{-5}$	$1.3617 \cdot 10^{-3}$
	Approach 2.4	$\mathbf{2.0000 \cdot 10^{-5}}$	$1.5841 \cdot 10^{-4}$
	Approach 3	$1.9864 \cdot 10^{-5}$	$1.4421 \cdot 10^{-4}$
$1.6200 \cdot 10^{-3}$	Approach 1	$1.6201 \cdot 10^{-5}$	$7.6132 \cdot 10^{-3}$
	Approach 2.1	$2.2911 \cdot 10^{-5}$	$4.3699 \cdot 10^{-3}$
	Approach 2.2	$3.2400 \cdot 10^{-5}$	$3.0495 \cdot 10^{-4}$
	Approach 2.3	$2.7655 \cdot 10^{-5}$	$2.1060 \cdot 10^{-3}$
	Approach 2.4	$\mathbf{3.2400 \cdot 10^{-5}}$	$3.0495 \cdot 10^{-4}$
	Approach 3	$3.2056 \cdot 10^{-5}$	$2.5787 \cdot 10^{-4}$
$2.6400 \cdot 10^{-3}$	Approach 1	$2.6401 \cdot 10^{-5}$	$1.1927 \cdot 10^{-2}$
	Approach 2.1	$3.7337 \cdot 10^{-5}$	$6.7587 \cdot 10^{-3}$
	Approach 2.2	$5.2800 \cdot 10^{-5}$	$6.4875 \cdot 10^{-4}$
	Approach 2.3	$4.5068 \cdot 10^{-5}$	$3.1805 \cdot 10^{-3}$
	Approach 2.4	$\mathbf{5.2800 \cdot 10^{-5}}$	$6.4875 \cdot 10^{-4}$
	Approach 3	$5.1902 \cdot 10^{-5}$	$5.0276 \cdot 10^{-4}$
$4.2800 \cdot 10^{-3}$	Approach 1	$4.2804 \cdot 10^{-5}$	$1.8159 \cdot 10^{-2}$
	Approach 2.1	$6.0532 \cdot 10^{-5}$	$1.0078 \cdot 10^{-2}$
	Approach 2.2	$8.5600 \cdot 10^{-5}$	$1.4573 \cdot 10^{-3}$
	Approach 2.3	$7.3066 \cdot 10^{-5}$	$4.5660 \cdot 10^{-3}$
	Approach 2.4	$\mathbf{8.5600 \cdot 10^{-5}}$	$1.4573 \cdot 10^{-3}$
	Approach 3	$8.3251 \cdot 10^{-5}$	$1.0467 \cdot 10^{-3}$
$6.9500 \cdot 10^{-3}$	Approach 1	$6.9510 \cdot 10^{-5}$	$2.6663 \cdot 10^{-2}$
	Approach 2.1	$9.8298 \cdot 10^{-5}$	$1.4297 \cdot 10^{-2}$
	Approach 2.2	$1.3900 \cdot 10^{-4}$	$3.3304 \cdot 10^{-3}$
	Approach 2.3	$1.1865 \cdot 10^{-4}$	$6.1129 \cdot 10^{-3}$
	Approach 2.4	$\mathbf{1.3900 \cdot 10^{-4}}$	$3.3304 \cdot 10^{-3}$
	Approach 3	$1.3278 \cdot 10^{-4}$	$2.2451 \cdot 10^{-3}$
$1.1300 \cdot 10^{-2}$	Approach 1	$1.1303 \cdot 10^{-4}$	$3.6943 \cdot 10^{-2}$
	Approach 2.1	$1.5983 \cdot 10^{-4}$	$1.8704 \cdot 10^{-2}$
	Approach 2.2	$2.2600 \cdot 10^{-4}$	$7.3965 \cdot 10^{-3}$
	Approach 2.3	$1.9292 \cdot 10^{-4}$	$7.4877 \cdot 10^{-3}$
	Approach 2.4	$\mathbf{2.2600 \cdot 10^{-4}}$	$7.3965 \cdot 10^{-3}$
	Approach 3	$2.0940 \cdot 10^{-4}$	$4.6598 \cdot 10^{-3}$
$1.8300 \cdot 10^{-2}$	Approach 1	$1.8307 \cdot 10^{-4}$	$4.6704 \cdot 10^{-2}$
	Approach 2.1	$2.5887 \cdot 10^{-4}$	$2.1506 \cdot 10^{-2}$
	Approach 2.2	$3.6600 \cdot 10^{-4}$	$1.5206 \cdot 10^{-2}$
	Approach 2.3	$\mathbf{3.1243 \cdot 10^{-4}}$	$9.1804 \cdot 10^{-3}$
	Approach 2.4	$3.6600 \cdot 10^{-4}$	$1.5206 \cdot 10^{-2}$
	Approach 3	$3.2233 \cdot 10^{-4}$	$8.6961 \cdot 10^{-3}$
$2.9800 \cdot 10^{-2}$	Approach 1	$2.9818 \cdot 10^{-4}$	$5.1850 \cdot 10^{-2}$
	Approach 2.1	$4.2161 \cdot 10^{-4}$	$2.0513 \cdot 10^{-2}$
	Approach 2.2	$5.9600 \cdot 10^{-4}$	$2.8227 \cdot 10^{-2}$
	Approach 2.3	$\mathbf{5.0881 \cdot 10^{-4}}$	$1.4889 \cdot 10^{-2}$
	Approach 2.4	$5.9600 \cdot 10^{-4}$	$2.8226 \cdot 10^{-2}$
	Approach 3	$4.8360 \cdot 10^{-4}$	$1.3689 \cdot 10^{-2}$
$4.8300 \cdot 10^{-2}$	Approach 1	$4.8347 \cdot 10^{-4}$	$4.7978 \cdot 10^{-2}$
	Approach 2.1	$\mathbf{6.8353 \cdot 10^{-4}}$	$1.6829 \cdot 10^{-2}$
	Approach 2.2	$9.6600 \cdot 10^{-4}$	$4.5366 \cdot 10^{-2}$
	Approach 2.3	$8.2476 \cdot 10^{-4}$	$2.6904 \cdot 10^{-2}$
	Approach 2.4	$9.6600 \cdot 10^{-4}$	$4.5366 \cdot 10^{-2}$
	Approach 3	$7.0025 \cdot 10^{-4}$	$1.6546 \cdot 10^{-2}$

(*continued on next page*)

Table A.1 (*continued*).

λ_f (m^2/s)	Approach	λ_c (1/s)	RMSE
$7.8500 \cdot 10^{-2}$	Approach 1	$7.8623 \cdot 10^{-4}$	$3.5732 \cdot 10^{-2}$
	Approach 2.1	$\mathbf{1.1114 \cdot 10^{-3}}$	$1.9828 \cdot 10^{-2}$
	Approach 2.2	$1.5700 \cdot 10^{-3}$	$6.3704 \cdot 10^{-2}$
	Approach 2.3	$1.3407 \cdot 10^{-3}$	$4.2225 \cdot 10^{-2}$
	Approach 2.4	$1.5700 \cdot 10^{-3}$	$6.3704 \cdot 10^{-2}$
	Approach 3	$1.0056 \cdot 10^{-3}$	$1.4389 \cdot 10^{-2}$
(b) λ_f from 1.27×10^{-1} to 1.0×10^{1}			
$1.2700 \cdot 10^{-1}$	Approach 1	$\mathbf{1.2732 \cdot 10^{-3}}$	$2.2534 \cdot 10^{-2}$
	Approach 2.1	$1.7993 \cdot 10^{-3}$	$3.0119 \cdot 10^{-2}$
	Approach 2.2	$2.5399 \cdot 10^{-3}$	$8.0989 \cdot 10^{-2}$
	Approach 2.3	$2.1696 \cdot 10^{-3}$	$5.7175 \cdot 10^{-2}$
	Approach 2.4	$2.5399 \cdot 10^{-3}$	$8.0988 \cdot 10^{-2}$
	Approach 3	$1.4744 \cdot 10^{-3}$	$9.2928 \cdot 10^{-3}$
$2.0700 \cdot 10^{-1}$	Approach 1	$\mathbf{2.0786 \cdot 10^{-3}}$	$1.3707 \cdot 10^{-2}$
	Approach 2.1	$2.9360 \cdot 10^{-3}$	$4.1339 \cdot 10^{-2}$
	Approach 2.2	$4.1398 \cdot 10^{-3}$	$9.8377 \cdot 10^{-2}$
	Approach 2.3	$3.5379 \cdot 10^{-3}$	$7.1745 \cdot 10^{-2}$
	Approach 2.4	$4.1397 \cdot 10^{-3}$	$9.8374 \cdot 10^{-2}$
	Approach 3	$2.2591 \cdot 10^{-3}$	$5.5317 \cdot 10^{-3}$
$3.3600 \cdot 10^{-1}$	Approach 1	$\mathbf{3.3828 \cdot 10^{-3}}$	$8.3934 \cdot 10^{-3}$
	Approach 2.1	$4.7744 \cdot 10^{-3}$	$5.2442 \cdot 10^{-2}$
	Approach 2.2	$6.7191 \cdot 10^{-3}$	$1.1701 \cdot 10^{-1}$
	Approach 2.3	$5.7468 \cdot 10^{-3}$	$8.6785 \cdot 10^{-2}$
	Approach 2.4	$6.7188 \cdot 10^{-3}$	$1.1700 \cdot 10^{-1}$
	Approach 3	$3.5392 \cdot 10^{-3}$	$3.6370 \cdot 10^{-3}$
$5.4600 \cdot 10^{-1}$	Approach 1	$\mathbf{5.5204 \cdot 10^{-3}}$	$4.5216 \cdot 10^{-3}$
	Approach 2.1	$7.7817 \cdot 10^{-3}$	$6.4640 \cdot 10^{-2}$
	Approach 2.2	$1.0916 \cdot 10^{-2}$	$1.3888 \cdot 10^{-1}$
	Approach 2.3	$9.3488 \cdot 10^{-3}$	$1.0398 \cdot 10^{-1}$
	Approach 2.4	$1.0915 \cdot 10^{-2}$	$1.3885 \cdot 10^{-1}$
	Approach 3	$5.6301 \cdot 10^{-3}$	$2.5488 \cdot 10^{-3}$
$8.8600 \cdot 10^{-1}$	Approach 1	$\mathbf{9.0206 \cdot 10^{-3}}$	$1.8559 \cdot 10^{-3}$
	Approach 2.1	$1.2689 \cdot 10^{-2}$	$7.9487 \cdot 10^{-2}$
	Approach 2.2	$1.7703 \cdot 10^{-2}$	$1.6627 \cdot 10^{-1}$
	Approach 2.3	$1.5196 \cdot 10^{-2}$	$1.2524 \cdot 10^{-1}$
	Approach 2.4	$1.7697 \cdot 10^{-2}$	$1.6618 \cdot 10^{-1}$
	Approach 3	$9.0195 \cdot 10^{-3}$	$1.8557 \cdot 10^{-3}$
$1.4400 \cdot 10^{0}$	Approach 1	$\mathbf{1.4830 \cdot 10^{-2}}$	$5.2113 \cdot 10^{-3}$
	Approach 2.1	$2.0788 \cdot 10^{-2}$	$9.9549 \cdot 10^{-2}$
	Approach 2.2	$2.8723 \cdot 10^{-2}$	$2.0255 \cdot 10^{-1}$
	Approach 2.3	$2.4755 \cdot 10^{-2}$	$1.5353 \cdot 10^{-1}$
	Approach 2.4	$2.8699 \cdot 10^{-2}$	$2.0226 \cdot 10^{-1}$
	Approach 3	$1.4548 \cdot 10^{-2}$	$1.3708 \cdot 10^{-3}$
$2.3400 \cdot 10^{0}$	Approach 1	$\mathbf{2.4564 \cdot 10^{-2}}$	$1.3626 \cdot 10^{-2}$
	Approach 2.1	$3.4225 \cdot 10^{-2}$	$1.2864 \cdot 10^{-1}$
	Approach 2.2	$4.6452 \cdot 10^{-2}$	$2.5059 \cdot 10^{-1}$
	Approach 2.3	$4.0339 \cdot 10^{-2}$	$1.9219 \cdot 10^{-1}$
	Approach 2.4	$4.6351 \cdot 10^{-2}$	$2.4965 \cdot 10^{-1}$
	Approach 3	$2.3540 \cdot 10^{-2}$	$9.9880 \cdot 10^{-4}$
$3.7900 \cdot 10^{0}$	Approach 1	$\mathbf{4.1078 \cdot 10^{-2}}$	$3.0578 \cdot 10^{-2}$
	Approach 2.1	$5.6634 \cdot 10^{-2}$	$1.7205 \cdot 10^{-1}$
	Approach 2.2	$7.4207 \cdot 10^{-2}$	$3.0819 \cdot 10^{-1}$
	Approach 2.3	$6.5421 \cdot 10^{-2}$	$2.4269 \cdot 10^{-1}$
	Approach 2.4	$7.3812 \cdot 10^{-2}$	$3.0535 \cdot 10^{-1}$
	Approach 3	$3.8063 \cdot 10^{-2}$	$6.9375 \cdot 10^{-4}$

(*continued on next page*)

Table A.1 (*continued*).

λ_f (m^2/s)	Approach	λ_c ($1/s$)	RMSE
$6.1600 \cdot 10^{0}$	Approach 1	$\mathbf{7.0606 \cdot 10^{-2}}$	$6.7329 \cdot 10^{-2}$
	Approach 2.1	$9.5429 \cdot 10^{-2}$	$2.3703 \cdot 10^{-1}$
	Approach 2.2	$1.1554 \cdot 10^{-1}$	$3.5544 \cdot 10^{-1}$
	Approach 2.3	$1.0548 \cdot 10^{-1}$	$2.9809 \cdot 10^{-1}$
	Approach 2.4	$1.1435 \cdot 10^{-1}$	$3.4884 \cdot 10^{-1}$
	Approach 3	$6.1908 \cdot 10^{-2}$	$4.8429 \cdot 10^{-4}$
$1.0000 \cdot 10^{1}$	Approach 1	$\mathbf{1.2702 \cdot 10^{-1}}$	$1.4697 \cdot 10^{-1}$
	Approach 2.1	$1.6472 \cdot 10^{-1}$	$3.2476 \cdot 10^{-1}$
	Approach 2.2	$1.6204 \cdot 10^{-1}$	$3.1328 \cdot 10^{-1}$
	Approach 2.3	$1.6338 \cdot 10^{-1}$	$3.1904 \cdot 10^{-1}$
	Approach 2.4	$1.6239 \cdot 10^{-1}$	$3.1480 \cdot 10^{-1}$
	Approach 3	$1.0092 \cdot 10^{-1}$	$1.0486 \cdot 10^{-3}$

Data availability

Data will be made available on request.

References

Adityawan, M., Sandi, C., Harlan, D., Farid, M., Kuntoro, A., Chaidar, A., Nugroho, J., et al., 2023. Numerical modeling of dam break induced flow through multiple buildings in an idealized city. Results Eng. 18, 101060.

Aksamit, N., Sapsis, T., Haller, G., 2019. Machine-learning ocean dynamics from Lagrangian drifter trajectories. arXiv preprint arXiv:1909.12895.

Atkins, P., De Paula, J., Keeler, J., 2023. Atkins' Physical Chemistry. Oxford University Press.

Baena-Nogueras, R., González-Mazo, E., Lara-Martín, P., 2017. Degradation kinetics of pharmaceuticals and personal care products in surface waters: Photolysis vs biodegradation. Sci. Total Environ. 590, 643–654.

Bolton, T., Zanna, L., 2019. Applications of deep learning to ocean data inference and subgrid parameterization. J. Adv. Model. Earth Syst. 11, 376–399.

Câmara, J., Montesdeoca-Esponda, S., Freitas, J., Guedes-Alonso, R., Sosa-Ferrera, Z., Perestrelo, R., 2021. Emerging contaminants in seafront zones: Environmental impact and analytical approaches. Separations 8, 95.

Canuto, V., Cheng, Y., Howard, A., 2011. Vertical diffusivities of active and passive tracers. Ocean. Model. 36, 198–207.

Chen, W., Nguyen, T., Pein, J., Jourdin, F., Fablet, R., Staneva, J., 2025. Physics-informed neural data assimilation for high-Resolution Coastal SPM reconstruction from model and satellite data. Appl. Ocean Res. 165, 1.

Cossarini, G., Feudale, L., Teruzzi, A., Bolzon, G., Coidessa, G., Solidoro, C., Di Biagio, V., Amadio, C., Lazzari, P., Brosich, A., et al., 2021. High-resolution reanalysis of the mediterranean sea biogeochemistry (1999–2019). Front. Mar. Sci. 8, 741486.

De Moura, C., Kubrusly, C., 2013. The Courant-Friedrichs-Lewy (CFL) Condition. AMC 10, 45–90.

Dohan, K., 2017. Ocean surface currents from satellite data. J. Geophys. Res.: Ocean. 122, 2647–2651.

El-Amrani, M., Ouardghi, A., Seaid, M., 2024. Modeling and simulation of pollution transport in the mediterranean sea using enriched finite element method. Math. Comput. Simulation 223, 565–587.

Ferreira, D., Marshall, J., Heimbach, P., 2005. Estimating eddy stresses by fitting dynamics to observations using a residual-mean ocean circulation model and its adjoint. J. Phys. Oceanogr. 35, 1891–1910.

Gasca-Ortiz, T., Domínguez-Mota, F., Pantoja, D., 2021. Determination of optimal diffusion coefficients in lake zirahuén through a local inverse problem. Mathematics 9, 1695.

Guerrini, F., Mari, L., Casagrandi, R., 2021. The dynamics of microplastics and associated contaminants: Data-driven Lagrangian and Eulerian modelling approaches in the mediterranean sea. Sci. Total Environ. 777, 145944.

Hirsch, C., 2007. Numerical Computation of Internal and External Flows: the Fundamentals of Computational Fluid Dynamics. Elsevier.

Kirkland, E., 2010. Bilinear interpolation. Springer.

Lee, D., Moon, J., Jung, S., Suh, S., Pyo, J., 2024. Classifying eutrophication spatio-temporal dynamics in river systems using deep learning technique. Sci. Total Environ. 954, 176585.

Lee, M., Seo, I., 2007. Analysis of pollutant transport in the han river with tidal current using a 2D finite element model. J. Hydro-Environment Res. 1, 30–42.

Lellouche, J., Greiner, E., Le Galloudec, O., Garric, G., Regnier, C., Drevillon, M., Benkiran, M., Testut, C., Bourdalle-Badie, R., Gasparin, F., et al., 2018. Recent updates to the copernicus marine service global ocean monitoring and forecasting real-time 1/12 high-resolution system. Ocean. Sci. 14, 1093–1126.

Leon, V., Bellas, J., 2023. Contaminants of emerging concern in the marine environment: Current challenges in marine pollution. Elsevier.

Li, L., Jiang, Z., Yin, Z., 2020. Compact finite-difference method for 2D time-fractional convection–diffusion equation of groundwater pollution problems. Comput. Appl. Math. 39, 142.

Ma, Y., Guo, M., De Schutter, B., 2025. An enhanced cell-based model for contaminant dispersion in marine environments. In: OCEANS 2025 Brest. pp. 1–9.

Maljutenko, I., Hassellöv, I., Eriksson, M., Ytreberg, E., Yngsell, D., Johansson, L., Jalkanen, J., Kouts, M., Kasemets, M., Moldanova, J., et al., 2021. Modelling spatial dispersion of contaminants from shipping lanes in the baltic sea. Marine Poll. Bull. 173, 112985.

Mercier, P., Thiébaut, M., Guillou, S., Maisondieu, C., Poizot, E., Pieterse, A., Thiébot, J., Filipot, J., Grondeau, M., 2021. Turbulence measurements: An assessment of acoustic Doppler current profiler accuracy in rough environment. Ocean Eng. 226, 108819.

Olivé Abelló, A., Pelegrí, J., Machín, F., 2023. A simple method for estimating horizontal diffusivity. J. Atmos. Ocean. Technol. 40, 739–752.

Paul, P., Webb, J., 2008. Reducing computational costs using a multi-region finite element method for electromagnetic scattering. IET Microw. Antennas Propag. 2, 427–433.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., Prabhat, F., 2019. Deep learning and process understanding for data-driven earth system science. Nature 566, 195–204.

Schneider, R., Henriksen, H., Stisen, S., 2020. A robust objective function for calibration of groundwater models in light of deficiencies of model structure and observations. Hydrol. Earth Syst. Sci. Discuss. 2020, 1–26.

Von Schuckmann, K., Le Traon, P., Smith, N., Pascual, A., Brasseur, P., Fennel, K., Djavidnia, S., Aaboe, S., Fanjul, E., Autret, E., et al., 2018. Copernicus marine service ocean state report. J. Oper. Ocean. 11, S1–S142.

Willmott, C., Matsuura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Clim. Res. 30, 79–82.
Wullems, B., Brauer, C., Baart, F., Weerts, A., 2023. Forecasting estuarine salt intrusion in the Rhine–Meuse Delta using an LSTM model. Hydrol. Earth Syst. Sci. 27, 3823–3850.