Modelling and simulation of pollution transport in the Mediterranean Sea using enriched finite element method

Mofdi El-Amrani[1,*], Abdelouahed Ouardghi[2], Mohammed Seaid[3]

[1] *Dpto. Matemáticas Applicada, Ciencia e Ingeniería de Materiales y Tecnología Electrónica, Universidad Rey Juan Carlos, 28933 Móstoles-Madrid, Spain. Email: mofdi.elamrani@urjc.es*
[2] *International Water Research Institute, University Mohammed VI Polytechnic, Benguerir, Morocco. Email: abdelouahed.ouardghi@um6p.ma*
[3] *Department of Engineering, University of Durham, South Road, Durham DH1 3LE, United Kingdom. Email: m.seaid@durham.ac.uk*

Abstract

This paper presents a novel numerical method for simulating the transport and dispersion of pollutants in the Mediterranean sea. The governing mathematical equations consist of a barotropic ocean model with friction terms, bathymetric forces, Coriolis and wind stresses coupled to an advection-diffusion equation with anisotropic dispersion tensor and source terms. The proposed numerical solver uses a multilevel adaptive semi-Lagrangian finite element method that combines various techniques, including the modified method of characteristics, finite element discretization, coupled projection scheme based on a rotational pressure correction algorithm, and an adaptive L^2-projection. The approach employs the gradient of the concentration as an error indicator for enrichment adaptations and increasing the number of quadrature points where needed without refining the mesh. The method is shown to provide accurate and efficient simulations for pollution transport in the Mediterranean sea. The proposed approach distinguishes itself from the well-established adaptive finite element methods for incompressible viscous flows by retaining the same structure and dimension of linear systems during the adaptation process.

Keywords. Mediterranean Sea, Incompressible Navier-Stokes equations, Enriched finite elements, Semi-Lagrangian method, L^2 projection, Adaptive algorithm

1 Introduction

The Mediterranean Sea is a semi-enclosed basin covering a volume of 3.75 million km^3 and surrounded by 22 countries. It has a length of approximately 4000 km, an average depth of about 1538 m, and a maximum depth of around 5120 m, see for example [21]. The Mediterranean sea is a relatively isolated water system that is renewed from the large Atlantic Ocean every 89 to 90 years through the Gibraltar strait [21]. This natural connection allows for dynamic exchanges of water, salt, heat, and other properties between the Mediterranean and North Atlantic basins, while also limiting them. Despite only occupying 0.82% of the ocean's surface, the Mediterranean sea harbors more than 18% of the known marine species, 21% of which are endangered, see [29] among others. In addition, the Mediterranean climate is characterized by warm temperatures, winter-dominated rainfall, and dry summers, compare [28, 42] and further references are therein. On the other hand, the construction of dams has led to a reduction in the seasonal high flows of major rivers resulting in a decrease in the continental shelf's ability to clear up pollutant deposits. Moreover, it has been demonstrated in [4] that evaporation exceeds rainfall and river flow which negatively impact the marine trophic chain. The Mediterranean Sea connects three continents namely, Africa, Asia, and Europe, and it is a vital shipping route accounting for 15% of worldwide marine trade stopovers. Consequently, maritime operations contribute to

*Corresponding author

marine pollution as stated in [27] among others. Water flow exchange through the strait, wind stress, and buoyant pressures at the surface owing to freshwater and heat fluxes, lead to the transport and dispersion of contaminants throughout the sea [36]. Although, there have been several recent investigations into the environmental elements of the Mediterranean sea, including water quality, currents, vegetation, wildlife, fishing, and aquaculture, no studies have yet been conducted on the numerical modeling of pollution transport in the sea. Numerical simulations are essential for maintaining a safe environment, as they provide the possibility to explore various scenarios and predict the interaction between water flow and pollution transport without risking the real world. Therefore, numerical tools can be developed to investigate the physical environment of the Mediterranean sea and evaluate development plans to decrease pollution risks in the seawater.

In the current study, the governing equations are based on the well-known barotropic ocean model which includes friction effects, bathymetric forces, Coriolis effects, and wind stresses, as well as eddy diffusion. To model pollution transport in the Mediterranean sea, we consider an advection-diffusion equation with an anisotropic dispersion tensor and a source term for the pollution release. The coupled flow and transport model yields a system in which convective terms dominate diffusive terms, particularly when the Reynolds number is large. In this work, we aim to develop a stable, accurate, and efficient numerical algorithm capable of approximating the transport and dispersion of pollutants in the Mediterranean sea. The Eulerian-based finite element methods for incompressible viscous flows have been widely investigated in the literature, see for example [46, 45, 25, 26, 11, 35, 7, 40, 10]. These methods typically use weighting techniques to stabilize the spatial discretization of convection-dominated flows, but truncation errors can increase drastically and impose significant constraints on the time steps employed in the numerical calculations. This is mainly due to the standard Courant-Friedrichs-Lewy (CFL) stability requirements. Moreover, the convective terms can cause computing challenges and nonphysical oscillations at high Reynolds numbers, making the Eulerian finite element methods vulnerable to sharp fronts, boundary layers, vertex shading, and shocks. To address these challenges, we investigate the use of semi-Lagrangian finite element methods which have been widely used in the literature to solve several problems in physical and engineering applications, see for instance [18, 17, 15, 16]. These methods avoid some of the limitations in the Eulerian-based finite element methods by following the trajectories of fluid particles rather than using a fixed grid. The present study aims to implement a robust semi-Lagrangian finite element method for the convection-dominated equation with anisotropic dispersion modelling pollution transport problems in the Mediterranean sea. Notice that the semi-Lagrangian approximations are a popular choice in computational fluid dynamics due to their ability to efficiently handle advection terms. These techniques utilize the Lagrangian coordinates defined by the characteristic curves to reformulate the governing equations. By doing so, the semi-Lagrangian approach combines the advection term and time derivative into a directional derivative along the characteristic curves resulting in a characteristic time-stepping method. One key advantage of the semi-Lagrangian schemes is their strong stability with no need for CFL conditions. This means that large time steps can be used in the simulations, which reduce the time truncation errors and save execution time compared to Eulerian-based methods.

Adaptive enriched semi-Lagrangian finite element methods have shown a good balance in accurately and efficiently solving convection-diffusion problems, incompressible Navier-Stokes equations and coupled flow-transport problems as evidenced in previous studies [31, 13, 32]. This class of methods introduces the concept of incorporating enrichment points into the L^2-projection procedure to capture the local features of the problem such as singularities, discontinuities, or high gradients which cannot be accurately captured by the conventional semi-Lagrangian finite element methods. Furthermore, incorporating quadrature points as enrichment points lead to well-conditioned linear systems which is crucial in ensuring accurate and efficient simulations. It should also be stressed that, the well-known partition of unity enriched finite element methods which incorporates enrichment functions into the approximation space have also shown to increase the accuracy of the results and reduce the computational costs, see for example [20, 39]. However, the major drawback of these approaches is that they result in dense and ill-conditioned systems that must be solved at each time step since they inject additional degrees of freedom to those previously constructed in the discretization.

The objective of the current study is to propose a novel approach to numerically simulate the transport and dispersion of pollutants in the Mediterranean sea. More precisely, we are investigating a semi-Lagrangian finite element method with multilevel adaptive enrichments. Here, the semi-Lagrangian method tracks the

Figure 1: Location and schematic description of the Mediterranean sea.

characteristics backwards at a time step to the departure points of a discrete set of virtual Lagrangian particles that arrive at a regular set of gridpoints. Note that previous works in [13, 31, 32] have used a second-order extrapolation based on the midpoint rule to evaluate these departure points. However, these solvers are not suitable for computing departure points in barotropic flows due to the rapid expansion of invariants in long time computations. This sensitivity to the evaluation of departure points can significantly impact the overall accuracy of the semi-Lagrangian schemes. To address this issue, we propose in this study the use of symplectic integrators proposed in [5] which are based on the Hamilton variational principle to evaluate departure points. These methods are considered to be structure-preserving and can handle the rapid growth of Hamiltonian even in long time simulations, ensuring correct streamline tracking. This approach improves the overall accuracy and efficiency of the semi-Lagrangian approximations for the considered flow problems as it offers an additional benefit in comparison to the methods studied in [13, 31, 32]. Another advantage of the proposed method is related to the efficiency of the search-locate algorithm utilized to locate the mesh elements containing the departure points within the computational domain. In this study, the search-locate algorithm has been enhanced to handle the large number of points that require identification within the discretized domain in an efficient manner. Furthermore, the proposed method uses a coupled projection scheme based on a rotational pressure correction presented in [19] to solve the Stokes problem along with a second-order backward difference formula for the time integration. The main advantage of this algorithm over the direct gradient conjugate approach used in [13, 32] is the improved stability and accuracy of the numerical solution. Indeed, the rotational pressure correction method utilizes a projection step that enforces the solenoidal condition on the velocity field and a correction step that removes any non-zero divergence from the pressure field. This approach ensures that the velocity and pressure fields are consistent with each other, leading to more accurate and physically meaningful results. In contrast, the direct gradient conjugate approach relies on an iterative procedure that can lead to numerical instabilities and suboptimal convergence. In addition, the rotational pressure correction scheme is computationally efficient and can handle complex geometries and unstructured meshes with ease, making it a preferred method for a wide range of flow simulations. Adaptive enrichment improves the accuracy of the numerical solution in regions where the solution exhibits sharp gradients or singularities. By adding more quadrature points in these regions, adaptive enrichment can capture the features of the solution more accurately leading to more precise results. Moreover, adaptive enrichments can reduce the computational cost by allowing the proposed method to use a coarser mesh then increase the number of quadrature points where it is needed. It should be noted that these methods use a class of gradient-based error indicators to identify domain areas

that need to be enriched. A gradient-based error indicator can provide a more accurate estimate of errors in the solution compared to other error estimators, see for instance [8, 34, 30, 33, 1, 9, 38, 44]. This is mainly because the gradient measures how rapidly the solution is changing, and regions where the gradient is large are likely to be the regions where the numerical error is also large. As a result, the number of enrichment points is refined only in regions where the solution is changing rapidly or where the error is largest. Two numerical test examples for transport and dispersion problems, including the transport and dispersion of pollutants in the Mediterranean sea, are used to test the performance of the proposed techniques. These enriched approaches are examined for various levels of enrichment and mesh refinements whereas the obtained results are compared to those obtained using the conventional method and against each other.

The paper is organized as follows. In section 2, we present the mathematical equations considered for modelling transport and dispersion of pollutants in the Mediterranean sea. The proposed enriched semi-Lagrangian finite element method is formulated in section 3. This section also includes the formulation of both the conventional and enriched semi-Lagrangian finite element methods, as well as the coupled projection scheme based on a rotational pressure correction used to solve the generalized Stokes stage. Section 4 introduces the multilevel adaptive enrichment procedure and the criteria used for mesh adaptation. In Section 5, we evaluate the numerical performance of the proposed method using two examples of transport and dispersion problems including a problem of pollutant transport in the Mediterranean sea. The results show that our novel technique provides the required efficiency, accuracy, and stability. Finally, section 6 presents the concluding remarks and summarizes the main contributions of the paper.

2 Mathematical models

The focus in this study is on transport of pollutants in the Mediterranean sea using a barotropic model and an advection-dispersion equation. Here, the barotropic model allows studying the water flow without the complexity of the vertical mixing of water masses, see for example [43]. The advection-dispersion equation, on the other hand, is a fundamental equation that describes the transport of a concentration in a fluid. This equation takes into account the effects of both advection which is the transport of the concentration due to the flow field, and dispersion which is the random movement of particles in the fluid. Thus, the governing equations consist of coupling a barotropic ocean model for flow along with an advection-dispersion equation for the concentration transport as

$$\begin{aligned} \nabla \cdot \boldsymbol{U} &= 0, \\ \frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{U} \cdot \nabla \boldsymbol{U} + \nabla p - \nu \Delta \boldsymbol{U} &= f \boldsymbol{U}^{\perp} - \gamma \boldsymbol{U} + \frac{\tau}{H} + \beta \left(C - C_{\infty}\right) \boldsymbol{e}, \\ \frac{\partial C}{\partial t} + \boldsymbol{U} \cdot \nabla C - \nabla \cdot (\boldsymbol{D} \nabla C) &= S, \end{aligned} \tag{1}$$

where $\boldsymbol{U} = (U, V)^{\top}$ is the depth-averaged horizontal velocity field defined by

$$\boldsymbol{U}(t, \boldsymbol{x}) = \frac{1}{H} \int_{-H}^{0} \mathbf{u}(t, x, y, z); dz,$$

with $\boldsymbol{U}$ is the velocity field, and H denotes the depth of the water measured from the undisturbed water surface. In (1), p is the pressure, f the Coriolis parameter, γ the the bottom friction coefficient, ν the kinematic viscosity, τ the wind stress acting on the upper surface, C the pollutant concentration, $S(t, \boldsymbol{x})$ the source term, $\boldsymbol{U}^{\perp} = (V, -U)^{\top}$, and $\boldsymbol{D}$ is the diffusion tensor defined as

$$\mathbf{D} = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix}, \tag{2}$$

where the coefficients D_{11}, D_{12}, D_{21}, and D_{22} are defined by

$$\begin{aligned} D_{11} &= D_m + \frac{\alpha_L U^2 + \alpha_T V^2}{\sqrt{U^2+V^2}}, \\ D_{12} &= D_{21} = (\alpha_L - \alpha_T)\frac{UV}{\sqrt{U^2+V^2}}, \\ D_{22} &= D_m + \frac{\alpha_L V^2 + \alpha_T U^2}{\sqrt{U^2+V^2}}, \end{aligned} \tag{3}$$

where D_m is the molecular dispersion coefficient, α_L and α_T are the longitudinal and transverse dispersion coefficients, respectively. It should be noted that equations (1) are solved in a bounded domain $\Omega \subset \mathbb{R}^2$ with boundary Γ and equipped with appropriate boundary and initial conditions to define a well-posed mathematical problem.

To numerically solve the equations (1), we first discretize the space domain $\overline{\Omega} = \Omega \cup \Gamma$ into a quasi-uniform partition $\Omega_h \subset \Omega$ consisting of triangular elements $\mathcal{T}_k$ such that $\overline{\Omega} = \cup_{k=1}^{N_e} \mathcal{T}_k$, where N_e is the number of elements in Ω_h and h is a space discretization parameter. For the conforming finite element spaces for the velocity/concentration and pressure, we use the mixed Taylor-Hood finite elements P_2-P_1, which means that we use quadratic elements for the velocity/concentration solutions and linear elements for the pressure solution. It is well known that for the mixed finite element formulation, the discrete velocity and pressure fields satisfy the inf-sup condition, see for example [14]. The finite element spaces associated with the mixed formulation are defined as

$$V_h = \Big\{ U_h \in C^0(\Omega) :\ U_h\big|_{\mathcal{T}_k} \in P_2(\mathcal{T}_k),\ \forall \mathcal{T}_k \in \Omega_h \Big\}, \qquad \mathcal{P}_h = \Big\{ p_h \in C^0(\Omega) :\ p_h\big|_{\mathcal{T}_k} \in P_1(\mathcal{T}_k),\ \forall \mathcal{T}_k \in \Omega_h \Big\},$$

where $P_1(\mathcal{T}_k)$ and $P_2(\mathcal{T}_k)$ are polynomial spaces of degree 1 and 2, respectively, defined in the element $\mathcal{T}_k$. Note that since the velocity field is a vector of two dimensions, the associated finite element space is defined as $\mathbf{V}_h = V_h \times V_h$.

Next, for the time discretization, we divide the time interval $[0, T]$ into sub-intervals $[t_n, t_{n+1}]$ of length $\Delta t = t_{n+1} - t_n$ for $n \geq 0$. We use the notation $w_h^n := w(t_n, \boldsymbol{x}_h)$ to denote the value of an arbitrary function w at time t_n in the position $\boldsymbol{x}_h$. The approximate values $\boldsymbol{U}_h^n(\boldsymbol{x}) \in \mathbf{V}_h$, $p_h^n(\boldsymbol{x}) \in \mathcal{P}_h$, and $C_h^n(x) \in V_h$ are then formulated as

$$\boldsymbol{U}_h^n(\boldsymbol{x}) = \sum_{j=1}^{M_v} \boldsymbol{\mathcal{U}}_j^n \phi_j(\boldsymbol{x}), \qquad p_h^n(\boldsymbol{x}) = \sum_{l=1}^{M_p} P_l^n \Psi_l(\boldsymbol{x}), \qquad C_h^n(\boldsymbol{x}) = \sum_{j=1}^{M_v} \mathcal{C}_j^n \phi_j(\boldsymbol{x}), \tag{4}$$

where M_v and M_p are respectively, the number of velocity/concentration and pressure gridpoints in Ω_h. The quantities $\boldsymbol{\mathcal{U}}_j^n = \left(U_j^n, V_j^n\right)^\top$, P_l^n and $\mathcal{C}_j^n$ are the corresponding nodal values of $\boldsymbol{U}_h^n(\boldsymbol{x})$, $p_h^n(\boldsymbol{x})$ and $C_h^n(\boldsymbol{x})$, respectively. These values are evaluated as $\boldsymbol{\mathcal{U}}_j^n = \boldsymbol{U}_h^n(\boldsymbol{x}_j)$, $P_l^n = p_h^n(\boldsymbol{y}_l)$ and $\mathcal{C}_j^n = C_h^n(\boldsymbol{x}_j)$, where $\{\boldsymbol{x}_j\}_{j=1}^{M_v}$ and $\{\boldsymbol{y}_l\}_{l=1}^{M_p}$ are the sets of velocity/concentration and pressure mesh points in Ω_h. It should be noted that $M_p < M_v$ and $\{\boldsymbol{y}_l\}_l \subset \{\boldsymbol{x}_j\}_j$. In (4), $\{\phi_j\}_{j=1}^{M_v}$ and $\{\Psi_l\}_{l=1}^{M_p}$ are respectively, the sets of global nodal basis functions of the velocity and pressure spaces characterized by the property $\phi_i(\boldsymbol{x}_j) = \delta_{ij}$ and $\Psi_i(\boldsymbol{y}_l) = \delta_{il}$, with δ denotes the Kronecker delta.

3 Enriched semi-Lagrangian finite element method

The semi-Lagrangian finite element method is a fractional-step technique that separates the advective part from the Stokes and dispersion parts during the time integration of (1). Thus, during each time step, the velocity, pressure and concentration are updated by solving first the advection equations

$$\begin{aligned} \frac{D\boldsymbol{U}}{Dt} &:= \frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{U} \cdot \nabla \boldsymbol{U} = \boldsymbol{0}, \\ \frac{DC}{Dt} &:= \frac{\partial C}{\partial t} + \boldsymbol{U} \cdot \nabla C = 0, \end{aligned} \tag{5}$$

followed next by these equations

$$\begin{aligned} \nabla \cdot \boldsymbol{U} &= 0, \\ \frac{\partial \boldsymbol{U}}{\partial t} + \nabla p - \nu \Delta \boldsymbol{U} &= f \boldsymbol{U}^{\perp} - \gamma \boldsymbol{U} + \frac{\tau}{H} + \beta \left(C - C_{\infty}\right) \boldsymbol{e}, \\ \frac{\partial C}{\partial t} - \nabla \cdot \left(\boldsymbol{D} \nabla C\right) &= S, \end{aligned} \tag{6}$$

where $\frac{D}{Dt} =: \frac{\partial}{\partial t} + \boldsymbol{U} \cdot \nabla$ is the material (total) derivative in the direction of flow field $\boldsymbol{U}$ which measures the rate of change following the trajectories of the flow particles. The finite element discretization is used for both steps (5) and (6). In the subsequent section, we formulate the techniques involved in the numerical solution of these problems.

3.1 Symplectic scheme for evaluation of departure points

The semi-Lagrangian method is used to transport fields in equations (5), with the property that the transported values remain constant along the trajectory of a fluid particle which travels from a departure point at time t_n to an arrival point at $\boldsymbol{x}_j$ at time t_{n+1}. Here, the characteristic curve is the path taken by the fluid particle, and the departure point at time t_n is denoted by $\boldsymbol{\chi}_j(t_n)$. Accordingly, the value that we require at the arrival point is equal to the value of the field at the departure point. Thus, the velocity field $\boldsymbol{U}$ is associated with the characteristic curve $\boldsymbol{\chi}_j(t)$ and satisfies the first-order ordinary differential equation

$$\begin{aligned} \frac{d\boldsymbol{\chi}_j(t)}{dt} &= \boldsymbol{U}_h\left(t, \boldsymbol{\chi}_j(t)\right), \qquad t \in [t_n, t_{n+1}], \\ \boldsymbol{\chi}_j(t_{n+1}) &= \boldsymbol{x}_j, \end{aligned} \tag{7}$$

where the departure point $\boldsymbol{\chi}_j(t_n) = \big(X_j(t_n), Y_j(t_n)\big)$ corresponds to the initial position of the particle that will reach the mesh point $\boldsymbol{x}_j = (x_j, y_j)^{\top}$ at time t_{n+1}, as illustrated in Figure 2. For simplicity in the presentation, we will refer to the departure point by $\boldsymbol{\chi}_j^n$. The conventional semi-Lagrangian method requires in a first step a numerical solution of (7). This solution can be formulated as

$$\boldsymbol{\chi}_j^n = \boldsymbol{x}_j - \int_{t_n}^{t_{n+1}} \boldsymbol{U}_h\left(t, \boldsymbol{\chi}_j(t)\right)\, dt. \tag{8}$$

The integral in the right-hand side of (8) is typically computed using numerical integration techniques such as Runge-Kutta methods. However, although these methods are versatile, they are not structure-preserving and they cause rapid system energy growth which render them not suitable for long-term simulations and particle tracking, see for instance [41]. This issue has also been highlighted in previous research [5] in which authors demonstrated that non-geometric integrators such as the well-established explicit Runge-Kutta integrators produce less accurate results with excessive numerical dissipation. In the present study, to address these concerns, a symplectic integrator is proposed as solver of (7) to handle the rapid energy growth in long-term simulations while accurately tracking the corresponding streamlines. It should be noted that symplectic integrators are only applicable to Hamiltonian systems. In two-dimensional incompressible flow, a stream-function ψ can be used to express equation (7) in a Hamiltonian form for the characteristics trajectories $\boldsymbol{\chi} = (X, Y)^{\top}$ as

$$\begin{aligned} \frac{dX}{dt} &= \frac{d\psi}{dY}, \\ \frac{dY}{dt} &= -\frac{d\psi}{dX}, \end{aligned} \tag{9}$$

where the stream-function ψ is considered as the Hamiltonian and the pair (X, Y) is considered as the canonical pair. Hence, the equations (9) are a Hamiltonian system for which it is possible to use symplectic integrators

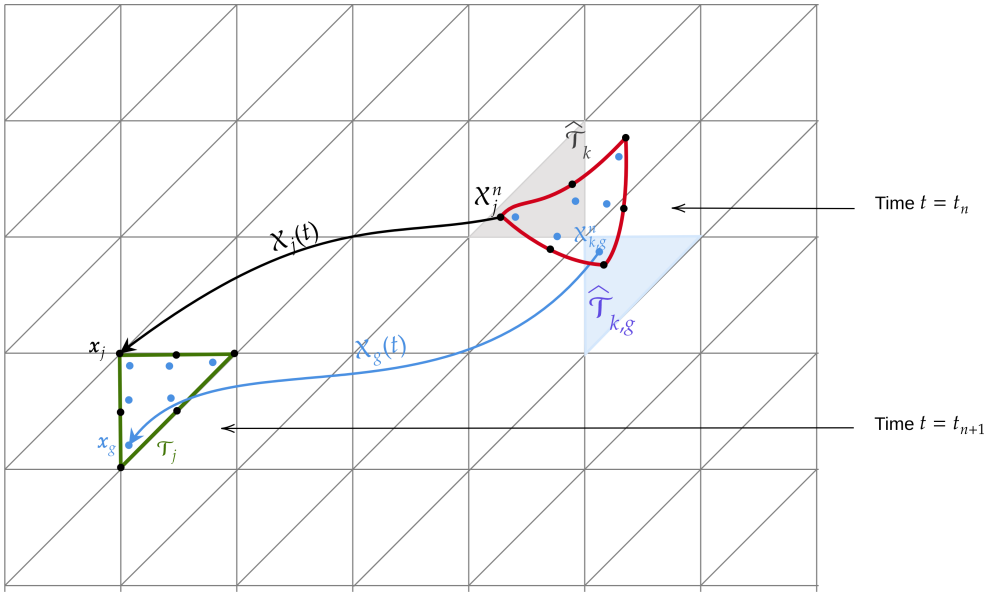

Figure 2: An illustration of the key parameters used for calculating the departure points in the L^2-projection method (coloured in blue) and in the conventional method (coloured in black).

to evaluate the integral in (8). It should be pointed out that a general method for creating volume-preserving difference schemes in three-dimensional divergence-free systems has also been investigated in [23]. This method involves decomposing a vector as a finite sum of two-dimensional Hamiltonian fields and composing the corresponding essentially symplectic schemes into a volume-preserving one. In this study, we consider the one-step method presented in [5] to evaluate the solution of (7) as

$$\boldsymbol{\chi}_{(r+1)} = \boldsymbol{\Phi}_{\Delta t}\left(\boldsymbol{\chi}_{(r)}\right), \tag{10}$$

where the method $\boldsymbol{\Phi}_{\Delta t}$ is assumed to be a second-order method and time symmetric *i.e.* $\boldsymbol{\Phi}_{\Delta t} \circ \boldsymbol{\Phi}_{-\Delta t} = \mathbf{I}$, with $\circ$ and $\mathbf{I}$ represent the function composition symbol and the identity operator, respectively. Note that these requirements are satisfied using the implicit midpoint scheme and the Verlet method. As example of composite symplectic method we consider the *Yoshida* technique given by

$$\boldsymbol{\chi}_{(r+1)} = \boldsymbol{\Psi}_{\Delta t}\left(\boldsymbol{\chi}_{(r)}\right), \qquad \boldsymbol{\Psi}_{\Delta t}\left(\boldsymbol{\chi}_{(r)}\right) = \boldsymbol{\Phi}_{\alpha\Delta t} \circ \boldsymbol{\Phi}_{(1-2\alpha)\Delta t} \circ \boldsymbol{\Phi}_{\alpha\Delta t}(\boldsymbol{\chi}_{(r)}), \tag{11}$$

where $\alpha = \frac{1}{3}\left(2 + 2^{\frac{1}{3}} + 2^{-\frac{1}{3}}\right)$ is chosen to guarantee a fourth-order method, see for instance [6]. In our simulations, we choose $\boldsymbol{\Phi}_{\Delta t}$ to be the following implicit midpoint scheme

$$\boldsymbol{\chi}_{(r+1)} = \boldsymbol{\chi}_{(r)} - \Delta t \boldsymbol{U}_h\left(\frac{\Delta t}{2}, \frac{\boldsymbol{\chi}_{(r)} + \boldsymbol{\chi}_{(r+1)}}{2}\right),$$

which can be solved using the Newton method or Picard iterations and this later are used in our simulations. Here, the solution of the symplectic integrator is the departure point $\boldsymbol{\chi}_j^n$ at time t_n of the trajectory of a particle located at the gridpoint $\boldsymbol{x}_j$ at time t_{n+1}. Hence, the advected solutions $\boldsymbol{U}$ and C are invariant along the characteristic curves associated with the flow fields and are mainly transported by the particle's motion. Therefore, the semi-Lagrangian solution of (5) can be expressed as

$$\boldsymbol{U}_h^{n+1}(\boldsymbol{x}_j) = \boldsymbol{U}_h^n(\boldsymbol{\chi}_j^n), \qquad C_h^{n+1}(\boldsymbol{x}_j) = C_h^n(\boldsymbol{\chi}_j^n). \tag{12}$$

It should be stressed that the computed departure points $\boldsymbol{\chi}_j^n$ do not always coincide with a gridpoint in Ω_h. Therefore, a search-locate method is needed to find the host element $\widehat{\mathcal{T}}_j$ where $\boldsymbol{\chi}_j^n$ belongs. In current work, we

Algorithm 1: Conventional search-locate algorithm

1 Given $\boldsymbol{\chi}_j$;
2 Chose an arbitrary point $\mathbf{p}^0 = (p^0, q^0)$ in the reference element $\mathcal{T}^*$;
3 Select an initial guess for the host element $\mathcal{T}_s$;
4 Define the mapping $\boldsymbol{F}_s$ from $\mathcal{T}^*$ to $\mathcal{T}_s$;
5 Compute $\mathbf{p}^* = (p^*, q^*)$ the solution of equation (3.1);
6 Test **if** $\mathbf{p}^* = (p^*, q^*)$ *in* $\mathcal{T}^*$ **then**
7 The host element $\widehat{\mathcal{T}}_j$ of $\boldsymbol{\chi}_j$ is $\mathcal{T}_s$. (*i.e.* $\widehat{\mathcal{T}}_j \equiv \mathcal{T}_s$);
8 Stop;
9 **else**
10 Apply the selection criteria to select a neighboring elements;
11 Go to step 4;
12 **end**

adopt the method developed in [18, 15, 2] for solving transport problems. Thus, to determine the host element $\widehat{\mathcal{T}}_j$ of the departure point $\boldsymbol{\chi}_j^n$, we chose an initial guess $\mathcal{T}_s$, then we check if the departure point belongs to this element, else, we select one of its neighboring elements according to a selection criterion, and we repeat the process till a convergence is reached. To achieve this step, we consider the one-to-one mapping $\boldsymbol{F}_s$ from an element of reference $\mathcal{T}^*$ to the element $\mathcal{T}_s$, where the reference element $\mathcal{T}^*$ for triangular mesh is defined as

$$\mathcal{T}^* = \{(p, q) \mid 0 \leq p, q \leq 1, 0 \leq 1 - p - q \leq 1\}, \tag{13}$$

or

$$\mathcal{T}^* = \left\{(p, q) \mid \min_i(\mathcal{N}_i(p, q)) \geq 0 \quad \text{and} \quad \max_i(\mathcal{N}_i(p, q)) \leq 1\right\}, \tag{14}$$

where $\mathcal{N}_i (i = 1, 2, 3)$ are the linear shape functions defined on the reference element $\mathcal{T}^*$ by $\mathcal{N}_1(p, q) = p$, $\mathcal{N}_2(p, q) = q$ and $\mathcal{N}_3(p, q) = 1 - p - q$. Therefore, for any $\boldsymbol{x}_p = (x_p, y_p) \in \mathcal{T}_s$, there exists a point $\mathbf{p}^* = (p^*, q^*) \in \mathcal{T}^*$ such that

$$\boldsymbol{x}_p - \mathbf{F}_s(\mathbf{p}^*) = 0. \tag{15}$$

Hence, if there is $\mathbf{p}^* = (p^*, q^*)$ in $\mathcal{T}^*$ satisfying (15), then $\boldsymbol{x}_p$ belongs to the element $\mathcal{T}_s$. Here, we search for a solution $\mathbf{p}^* = (p^*, q^*)$ of (15) in $\mathcal{T}^*$ using the Newton method as follows: Let $\boldsymbol{x}_p = (x_p, y_p) \in \mathcal{T}_s$ and $\mathbf{p}^0 = (p^0, q^0) \in \mathcal{T}^*$, then for $k \geq 0$

$$\mathbf{p}^{k+1} = \mathbf{p}^k - \mathbf{J}_{-\boldsymbol{F}_s}^{-1}\left(\boldsymbol{x}_p - \boldsymbol{F}_s(\mathbf{p}^k)\right) = \mathbf{p}^k + \mathbf{J}_{\boldsymbol{F}_s}^{-1}\left(\boldsymbol{x}_p - \boldsymbol{F}_s(\mathbf{p}^k)\right),$$

where $\mathbf{J}_{\boldsymbol{F}_s}^{-1} = -\mathbf{J}_{-\boldsymbol{F}_s}^{-1}$ is the inverse of the Jacobian matrix $\mathbf{J}_{\boldsymbol{F}_s}$ of the mapping $\boldsymbol{F}_s$. If $\boldsymbol{x}_p \in \mathcal{T}_s$, the Newton method (3.1) will converge to the unique solution $\mathbf{p}^* = (p^*, q^*) \in \mathcal{T}^*$ independently of the initial guess $\mathbf{p}^0 = (p^0, q^0) \in \mathcal{T}^*$. Furthermore, since $\mathcal{T}_s$ is an arbitrary element, then we need to test whether $\mathbf{p}^{k+1}$ belongs to $\mathcal{T}^*$ or not. Thus, using (14) we have $\mathbf{p}^{k+1} \in \mathcal{T}^*$ if

$$\min_i(\mathcal{N}_i(\mathbf{p}^{k+1})) \geq 0, \qquad \text{and} \qquad \max_i(\mathcal{N}_i(\mathbf{p}^{k+1})) \leq 1. \tag{16}$$

If the iterate $\mathbf{p}^{k+1} \notin \mathcal{T}^*$, then $\mathbf{x}_p \notin \mathcal{T}_s$, and as a consequence, there is no $\mathbf{p}^* = (p^*, q^*) \in \mathcal{T}^*$ such that (15) holds. Thus, a neighboring element of $\mathcal{T}_s$ must be selected and reconsider the equation (3.1) for the new guess. The selection criteria used in this work can be expressed as follows:

- Let $l = \text{index}\left(\min_i(\mathcal{N}_i(\mathbf{p}^{k+1}))\right)$.
- Select the element $\mathcal{T}_r$ such that $\Gamma_{sr} = \mathcal{T}_s \cap \mathcal{T}_r$ is the side of $\mathcal{T}_s$ opposite to the vertex $\boldsymbol{x}_l$.

In summary, the search-locate algorithm for departure points is carried out using Algorithm 1. Notice that, since linear triangles are used for the search-locate algorithm in our simulations, the one-to-one mapping $\boldsymbol{F}_s$ is a linear form defined by $\boldsymbol{F}_s(p,q) = \boldsymbol{x}_p = \sum_{i=1}^{3} \boldsymbol{x}_{i,s}\mathcal{N}_i(p,q)$, with $(\boldsymbol{x}_{1,s}, \boldsymbol{x}_{2,s}, \boldsymbol{x}_{3,s})$ are the three vertices of the triangle $\mathcal{T}_s$. In this case, the equation (15) can be rewritten as

$$\begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{pmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{pmatrix} \begin{pmatrix} p^* \\ q^* \end{pmatrix} + \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}, \tag{17}$$

and therefore the calculation of the solution $\mathbf{p}^* = (p^*, q^*)$ is straightforward. Once the departure point $\boldsymbol{\chi}_j^n$ and its host element $\widehat{\mathcal{T}}_j$ are determined, solutions in the departure point at time t_n can be approximated from the known values at the vertices of $\widehat{\mathcal{T}}_j$ as

$$\boldsymbol{U}_h^n\left(\boldsymbol{\chi}_j^n\right) = \sum_{i=1}^{N} \boldsymbol{U}_h^n(\widehat{\boldsymbol{x}}_i)\varphi_i\left(\boldsymbol{\chi}_j^n\right), \qquad C_h^n\left(\boldsymbol{\chi}_j^n\right) = \sum_{i=1}^{N} C_h^n(\widehat{\boldsymbol{x}}_i)\varphi_i\left(\boldsymbol{\chi}_j^n\right), \tag{18}$$

where $\{\varphi_i\}_{i=1}^N$ are the local shape functions in the element $\widehat{\mathcal{T}}_j$, N is the number of nodes which define the velocity/concentration mesh type, and $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^N$ are the vertices of the element $\widehat{\mathcal{T}}_j$. Thus, for all mesh nodes $\{\boldsymbol{x}_j\}_j$ the solutions (12) can be reformulated as

$$\boldsymbol{\mathcal{U}}_j^{n+1} = \boldsymbol{U}_h^{n+1}(\boldsymbol{x}_j) = \boldsymbol{U}_h^n\left(\boldsymbol{\chi}_j^n\right) := \widehat{\boldsymbol{\mathcal{U}}}_j^n, \qquad \mathcal{C}_j^{n+1} = C_h^{n+1}(\boldsymbol{x}_j) = C_h^n\left(\boldsymbol{\chi}_j^n\right) := \widehat{\mathcal{C}}_j^n, \tag{19}$$

where $\widehat{\boldsymbol{\mathcal{U}}}_j^n$ and $\widehat{\mathcal{C}}_j^n$ are a short notations of $\boldsymbol{U}_h^n\left(\boldsymbol{\chi}_j^n\right)$ and $C_h^n\left(\boldsymbol{\chi}_j^n\right)$, respectively. Consequently, the conventional semi-Lagrangian finite element solutions of (5) can be expressed as

$$\boldsymbol{U}_h^{n+1}(\boldsymbol{x}) = \sum_{j=1}^{M_v} \widehat{\boldsymbol{\mathcal{U}}}_j^n \phi_j(\mathbf{x}), \qquad C_h^{n+1}(\boldsymbol{x}) = \sum_{j=1}^{M_v} \widehat{\mathcal{C}}_j^n \phi_j(\mathbf{x}). \tag{20}$$

Note that the performance of the conventional semi-Lagrangian finite element approach depends on the size of the computational mesh utilized in the numerical simulations. Furthermore, it has been demonstrated in [18] that the conventional semi-Lagrangian finite element approach fails to properly handle steep gradients exhibited by the convective components if the computational mesh is not fine enough. In order to overcome such problems, the suggested methodology in the present work is to incorporate local enrichments using the L^2-projection. This would enhance the accuracy of the semi-Lagrangian finite element solution of the considered coupled flow-transport problems.

3.2 L^2-projection for local enrichments

In this section, we introduce a novel enrichment approach for the semi-Lagrangian finite element solution of convection-dominated flow problems based on the L^2-projection investigated in [17, 18]. For brevity in the presentation, we formulate the proposed approach only for the concentration solution and the velocity solution can be obtained using the same implementation. Therefore, the concentration solution in (19) can be written as

$$C_h^{n+1}(\boldsymbol{x}_j) = C_h^n\left(\boldsymbol{\chi}_j^n\right). \tag{21}$$

Multiplying both sides of equation (21) by the finite element basis functions ϕ_i and integrating over Ω yields

$$\int_\Omega C_h^{n+1}(\boldsymbol{x})\phi_i(\boldsymbol{x})d\boldsymbol{x} = \int_\Omega C_h^n\left(\boldsymbol{\chi}^n\right)\phi_i(\boldsymbol{x})\, d\boldsymbol{x}, \qquad i = 1, \ldots, M_v. \tag{22}$$

Thus, the equation (22) can be assembled in a global matrix-vector structure as

$$[\mathbf{M}]\left\{\mathbf{C}^{n+1}\right\} = \{\mathbf{r}^n\}, \tag{23}$$

where $[\mathbf{M}]$ is the finite element mass matrix with entries $m_{ij} = \int_\Omega \phi_j \phi_i d\boldsymbol{x}$, $\mathbf{C}^{n+1}$ is the vector formed of the unknown nodal solutions C_j^{n+1} and $\mathbf{r}^n$ is the right-hand side vector with entries r_i^n and defined as

$$r_i^n = \int_\Omega C_h^n\left(\boldsymbol{\chi}^n\right) \phi_i(\boldsymbol{x})\, d\boldsymbol{x}. \tag{24}$$

To evaluate the integrals $\{r_i^n\}$ in equation (24), a quadrature rule is used as

$$r_i^n = \sum_{k=1}^{N_e} \int_{\mathcal{T}_k} C_h^n\left(\boldsymbol{\chi}^n\right) \phi_i(\boldsymbol{x})\, d\boldsymbol{x} \approx \sum_{k=1}^{N_e} \sum_{q=1}^{N_{k,Q}} \omega_{q,k} {C_h}^n(\boldsymbol{\chi}_{q,k}^n)\phi_i(\boldsymbol{x}_{q,k}), \tag{25}$$

where $\boldsymbol{x}_{q,k} = (x_{q,k}, y_{q,k})^\top$ are the quadrature points associated with the element $\mathcal{T}_k$, $\omega_{q,k}$ its corresponding weights, N_e is the total number of elements in computational mesh, $N_{k,Q}$ is the total number of quadrature points in the element $\mathcal{T}_k$, and $\boldsymbol{\chi}_{q,k}^n$ is the departure point reaching the point $\boldsymbol{x}_{q,k}$ at time t_{n+1}. Here, $C_h^n(\boldsymbol{\chi}_{q,k}^n)$ is the concentration solution evaluated at the departure point $\boldsymbol{\chi}_{q,k}^n$ using the equation (18) as

$$\widehat{C}_{q,k}^n := C_h^n(\boldsymbol{\chi}_{q,k}^n) = \sum_{i=1}^{N} C_h^n(\widehat{\boldsymbol{x}}_i)\varphi_i(\boldsymbol{\chi}_{q,k}^n), \qquad q = 1, \cdots, N_{k,Q}, \tag{26}$$

where $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^N$ are the vertices of the element $\widehat{\mathcal{T}}_{q,k}$ hosting $\boldsymbol{\chi}_{q,k}^n$, and $\{\varphi_i\}_{i=1}^N$ are their corresponding local basis functions. Therefore, the entries m_{ij} and r_i^n in (23) are evaluated as

$$m_{ij} \approx \sum_{k=1}^{N_e} \sum_{q=1}^{N_{k,Q}} \omega_{q,k}\phi_j(\boldsymbol{x}_{q,k})\phi_i(\boldsymbol{x}_{q,k}), \qquad r_i^n \approx \sum_{k=1}^{N_e} \sum_{q=1}^{N_{k,Q}} \omega_{q,k}\widehat{C}_{q,k}^n\phi_i(\boldsymbol{x}_{q,k}), \qquad i,j = 1, \ldots, M. \tag{27}$$

Similarly, the approximation of the velocity field can be reformulated as

$$[\mathbf{M}]\left\{\boldsymbol{\mathcal{U}}^{n+1}\right\} = \left\{\mathbf{z}^n\right\}, \tag{28}$$

where $\boldsymbol{\mathcal{U}}^{n+1}$ is the vector of the unknown nodal solutions with entries $\mathbf{U}_j^{n+1} = \left(U_j^{n+1}, V_j^{n+1}\right)$ and $\mathbf{z}^n$ is the right-hand side with inputs z_i^n assembled in same manner as in (25) by

$$z_i^n \approx \sum_{k=1}^{N_e} \sum_{q=1}^{N_{k,Q}} \omega_{q,k}\widehat{\boldsymbol{\mathcal{U}}}_{q,k}^n\phi_i(\boldsymbol{x}_{q,k}), \qquad j = 1, \ldots, M, \tag{29}$$

where $\widehat{\boldsymbol{\mathcal{U}}}_{q,k}^n = \mathbf{U}_h^n(\boldsymbol{\chi}_{q,k}^n)$ is the velocity solution calculated using (26) at the departure point $\boldsymbol{\chi}_{q,k}^n$ as

$$\widehat{\boldsymbol{\mathcal{U}}}_{q,k}^n := \boldsymbol{U}_h^n(\boldsymbol{\chi}_{q,k}^n) = \sum_{i=1}^{N} \boldsymbol{\mathcal{U}}_h^n(\widehat{\boldsymbol{x}}_i)\varphi_i(\boldsymbol{\chi}_{q,k}^n), \qquad q = 1, \cdots, N_{k,Q}, \tag{30}$$

Note that the conventional search-locate algorithm proposed in [2] is considered to be accurate and suitable for the standard semi-Lagrangian finite element methods. However, in the proposed enriched methods, we compute the corresponding departure points for each quadrature point in each element. Although this algorithm converges in a few iterations, using a high number of enrichments makes this search-locate algorithm very demanding. This is mainly because all departure points must be back-traced starting from an initial guess for the host element, as illustrated in the left plot of Figure 3. In this plot, we show the paths followed by the conventional search-locate algorithm back-trace the six quadrature points of an element starting from an arbitrary initial guess $\mathcal{T}_s$. It is clear from this figure that for each point, there is a considerable number of elements that must be tested before allocating the right element. Moreover, when high numbers of enrichments are used, back-tracking departure points using the considered algorithm requires more computational time,

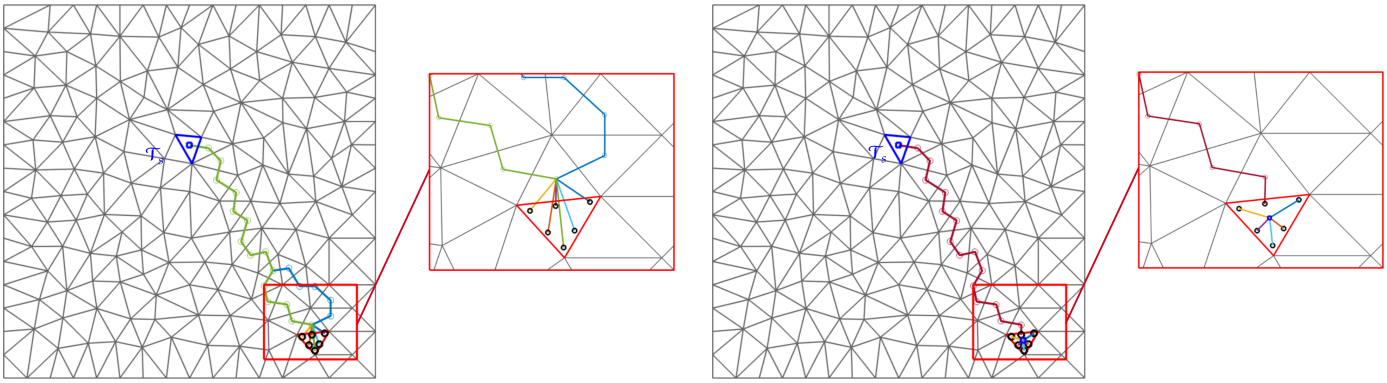

Figure 3: Particle tracking using the conventional search-locate algorithm (left plot) and the modified search-locate algorithm (right plot).

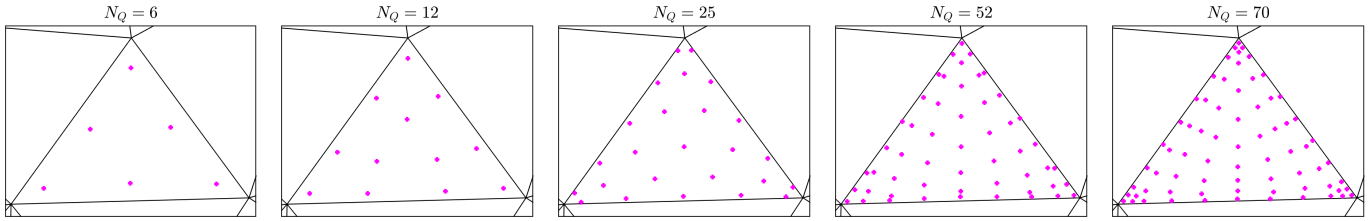

Figure 4: Distribution of Dunavant quadrature points employed for global and local enrichments.

which would affect the efficiency of the developed approach. In the current work, we have increased the efficiency of our method by improving the search-locate steps in Algorithm 1. Since in our enriched method, we search for the departure points of a set of quadrature points of a considered element, these departure points must fall in the same element or in elements that are neighbors. Thus, we can back-trace one of the points using the conventional search-locate algorithm, then use its host element as an initial guess to search for the remaining departure points, as shown in the right plot of Figure 3. For example, for a given number of enrichments $N_{k,Q} = 70$, the modified search-locate algorithm finds the first point after a few iterations, and the other 69 points after one or a maximum of two iterations. The modified search-locate algorithm is detailed in Algorithm 2.

In the current work, the well-established quadrature rules [12] are employed, see Figure 4 for distributions of these quadrature points with $N_{k,Q} = 6$, 12, 25, 52, and 70. Notice that we interpret equations (27) as an enrichment approach based on the distribution of quadrature points in the computational mesh. Hence, as stated in the previous section, the number of quadrature points $N_{k,Q}$ can be adjusted globally using the entire computational mesh or locally at each element. It should be also noted that in the conventional semi-Lagrangian finite element method, no linear systems of algebraic equations are solved, and the numerical solution is interpolated using the quadratic shape functions of the element where the departure points $\boldsymbol{\chi}_i^n$ reside. However, in the proposed semi-Lagrangian finite element method, in contrast to the conventional approach, the departure points $\boldsymbol{\chi}_{q,k}^n$ for all quadrature points belonging to each element $\mathcal{T}_k$ in the considered mesh are evaluated, and a linear system solution is solved to update the numerical solution. Note that other quadrature rules can also be straightforwardly applied in our approach.

3.3 Coupled projection method for solution of the Stokes problem

To solve the Stokes problem (6) we employ the coupled projection method based on rotational pressure correction along with a second-order implicit backward differentiation formula (BDF2) also known as Gear scheme. This enables us to update the velocity field and pressure at each step and complete the implementation of

Algorithm 2: Modified search-locate algorithm

for *each element* $\mathcal{T}_e$ **do**
  Chose an arbitrary point $\mathbf{p}^0 = (p^0, q^0)$ in the reference element $\mathcal{T}^*$;
  Select an initial guess for the host element $\mathcal{T}_s$;
  for *each departure point* $\boldsymbol{\chi}_{q,k}$ **do**
    Find the mapping $\boldsymbol{F}_s$ from $\mathcal{T}^*$ to $\mathcal{T}_s$;
    Compute $\mathbf{p}^* = (p^*, q^*)$ the solution of equation (3.1);
    Test **if** $\mathbf{p}^* = (p^*, q^*)$ *in* $\mathcal{T}^*$ **then**
      The host element $\widehat{\mathcal{T}}_{q,k}$ of $\boldsymbol{\chi}_{q,k}$ is $\mathcal{T}_s$. (*i.e.* $\widehat{\mathcal{T}}_j \equiv \mathcal{T}_s$);
      Stop;
    else
      Apply the selection criteria to select a neighbor elements;
      Go to step 5;
    end
    Select $\widehat{\mathcal{T}}_{q,k}$ as initial guess;
    Go to step 4;
  end
end

the semi-Lagrangian finite element method for solving the incompressible Navier-Stokes equations (1). Note that by incorporating the previous gradient step in the velocity prediction problem, the order of the method is improved compared to the standard projection methods. In addition, the use of rotational pressure correction helps to avoid artificial boundary conditions on the pressure and enhances the convergence rate of the method, see [19, 22] among others. Applied to the Stokes equations (6), the proposed method is carried out using the following steps:

1. Assume that $\widehat{\boldsymbol{U}}^n$, $\widehat{\boldsymbol{U}}^{n-1}$, $\widehat{C}^n$, $\widehat{C}^{n-1}$ and p^n are known.

2. Solve for $C^{n+1} \in \mathbf{V}_h$

$$\frac{3}{2\Delta t} C^{n+1} - \nabla \cdot \left(D \nabla C^{n+1}\right) = \left(\frac{-4\widehat{C}^n + \widehat{C}^{n-1}}{2\Delta t}\right) + S^n. \tag{31}$$

3. Solve for $\widetilde{\boldsymbol{U}}^{n+1} \in \mathbf{V}_h$

$$\frac{3}{2\Delta t}\widetilde{\boldsymbol{U}}^{n+1} - \nu\Delta\widetilde{\boldsymbol{U}}^{n+1} + \gamma\widetilde{\boldsymbol{U}}^{n+1} - f\left(\widetilde{\boldsymbol{U}}^{n+1}\right)^{\top} - \beta(C^{n+1} - C_\infty)\mathbf{e} = -\left(\frac{-4\widehat{\boldsymbol{U}}^n + \widehat{\boldsymbol{U}}^{n-1}}{2\Delta t}\right) - \nabla p^n + \frac{\tau}{H}. \tag{32}$$

4. Compute the solution φ of the Poisson problem using the projection step

$$\Delta\varphi = \frac{3}{2\Delta t}\nabla \cdot \widetilde{\boldsymbol{U}}^{n+1}, \qquad \text{with} \quad \partial_{\mathbf{n}}\varphi = 0 \quad \text{on } \partial\Omega. \tag{33}$$

5. Update the velocity $\boldsymbol{U}^{n+1}$ using the correction step

$$\boldsymbol{U}^{n+1} = \widetilde{\boldsymbol{U}}^{n+1} - \frac{2\Delta t}{3}\nabla\varphi. \tag{34}$$

6. Update the pressure p^{n+1} using the correction step

$$p^{n+1} = p^n + \varphi - \nu\nabla \cdot \widetilde{\boldsymbol{U}}^{n+1}. \tag{35}$$

It should be stressed that in equations (31)-(32), the solutions $\widehat{C}^{n-1}$, $\widehat{C}^{n}$, $\widetilde{\boldsymbol{U}}^{n}$ and $\widetilde{\boldsymbol{U}}^{n-1}$ are required to advance the solution C_h^{n+1} and the iterate velocity $\boldsymbol{U}_h^{(k)}$ in time. In this case, only one initial condition is given at time $t = 0$, and the implicit Euler scheme is used to get the second condition. In addition, boundary conditions for C^{n+1}, $\widetilde{\boldsymbol{U}}^{n+1}$ and $\boldsymbol{U}^{n+1}$ are those given by the problem under study. Here, $\widehat{\boldsymbol{U}}^{n}$, $\widehat{\boldsymbol{U}}^{n-1}$, $\widehat{C}^{n}$ and $\widehat{C}^{n-1}$ are evaluated at the departure points using values of the solutions $\boldsymbol{U}^{n}$, $\boldsymbol{U}^{n-1}$, C^{n} and C^{n-1}, respectively.

4 Multilevel adaptive enrichments

In many applications involving coupled flow-transport problems, steep solution gradients, localized eddies, and boundary shear layers occur in numerical solutions. The enriched semi-Lagrangian finite element method formulated in the previous section accurately captures these features but it requires very fine meshes and a large number of quadrature points, particularly in regions where the solution gradients are extremely high. To avoid uniform enrichments in the entire computational domain, we propose a multilevel adaptive algorithm for local enrichments to increase the accuracy and efficiency of the method. The key idea in this multilevel adaptive technique is to refine the number of quadrature points $N_{k,Q}$ in mesh elements where the solution gradient generates high values and de-refine otherwise according to a given adaptation criterion. In practice, an error indicator and a specified tolerance are needed to adjust the quadrature at each time step. The well-known gradient-based error estimators have been widely used in the literature for h-adaptive finite element methods for solving incompressible Navier-Stokes equations, see [8, 34, 30, 33, 1, 9, 38, 44] among others. However, most of these gradient-based h-adaptive techniques use an initial coarse mesh to compute a primary solution for estimating the gradient errors. Consequently, error accumulation in time is expected due to the coarse mesh used in the approximation, and the computational cost becomes prohibitive due to multiple interpolations between adaptive meshes. In the current work, we consider the normalized gradient of the concentration as an adaptive criterion for the local enrichments of each element in the computational domain as

$$\eta^{n+1}\left(\mathcal{T}_k\right) = \frac{\left\|\nabla C_{\mathcal{T}_k}^{n+1}\right\|}{\max\limits_{j=1}^{N_e}\left\|\nabla C_{\mathcal{T}_j}^{n+1}\right\|}, \tag{36}$$

where $C_{\mathcal{T}_k}^{n+1}$ is the concentration on the element $\mathcal{T}_k$ at time t_{n+1} and $\left\|\nabla C_{\mathcal{T}_k}^{n+1}\right\|$ is the L^2-norm of the solution gradient on $\mathcal{T}_k$ defined by

$$\left\|\nabla C_{\mathcal{T}_k}^{n+1}\right\| = \sqrt{\int_{\mathcal{T}_k} \nabla C_{\mathcal{T}_k}^{n+1} \cdot \nabla C_{\mathcal{T}_k}^{n+1}\, d\Omega}. \tag{37}$$

Note that the considered error indicator (36) can benefit from the semi-Lagrangian method to approximate the gradient $\left\|\nabla C_{\mathcal{T}_k}^{n+1}\right\|$ backwards in time from the known solution at time t_n. Thus, applying the gradient to the concentration solution in (20) on the element $\mathcal{T}_k$ we obtain

$$\nabla C_{\mathcal{T}_k}^{n+1} = \sum_{i=1}^{N} \widehat{C}_i^n \nabla \varphi_i. \tag{38}$$

Therefore, the integral in (37) can be evaluated as

$$\begin{aligned}
\left\|\nabla C_{\mathcal{T}_k}^{n+1}\right\| &= \sqrt{\int_{\mathcal{T}_k} \nabla C_{\mathcal{T}_k}^{n+1} \cdot \nabla C_{\mathcal{T}_k}^{n+1}\, d\Omega}, \\
&= \sqrt{\int_{\mathcal{T}_k} \left(\sum_{i=1}^{N} \widehat{C}_i^n \nabla\varphi_i\right) \cdot \left(\sum_{j=1}^{N} \widehat{C}_j^n \nabla\varphi_j\right)\, d\Omega}, \\
&= \sqrt{\sum_{i=1}^{N} \widehat{C}_i^n \left(\sum_{j=1}^{N} \widehat{C}_j^n \int_{\mathcal{T}_k} \nabla\varphi_i \cdot \nabla\varphi_j\, d\Omega\right)}, \\
&= \sqrt{\left(\widehat{\mathcal{C}}_{\mathcal{T}_k}^n\right)^\top \mathbf{S}_{\mathcal{T}_k} \widehat{\mathcal{C}}_{\mathcal{T}_k}^n}. \qquad (39)
\end{aligned}$$

where $\widehat{\mathcal{C}}_{\mathcal{T}_k}^n = (\widehat{\mathcal{C}}_1^n, \ldots, \widehat{\mathcal{C}}_N^n)^\top$ are the solution values computed at the departure points of vertices of the element $\mathcal{T}_k$ at time t_n, and $\mathbf{S}_{\mathcal{T}_k}$ denotes the elementary stiffness matrix associated with the element $\mathcal{T}_k$. It should be mentioned that normalizing the error indicator ensures that the criteria (36) takes values between 0 and 1. As a result, the multilevel adaptation method we suggest in this study is carried out as follows:

Assuming the tolerances $\{\varepsilon_m\}$ such that $0 = \varepsilon_0 < \varepsilon_1 < \varepsilon_2 < \varepsilon_3 < \varepsilon_4 = 1$ are given. If an element of the computational mesh $\mathcal{T}_k$ meets the following criteria

$$\varepsilon_m \leq \eta^{n+1}(\mathcal{T}_k) \leq \varepsilon_{m+1}, \qquad 0 \leq m \leq 3,$$

then the element $\mathcal{T}_k$ is enriched using the quadrature pairs $(\boldsymbol{x}_{q,k}, w_{q,k})$ with $q = 1, 2 \ldots, N_{k,q_m}$. It should be noted that the values of tolerances $\{\varepsilon_m\}$ and the number of levels vary depending on the problem under study. In summary, the proposed adaptive enriched semi-Lagrangian finite element method for solving the advection problem is described in Algorithm 3. It should be pointed out that a local refinement on triangular elements is also possible but with an additional cost of re-meshing and interpolating solutions between the meshes. Unlike the well-established h-adaptive methods, that require initial coarse meshes to evaluate a primary solution for error estimations, the error indicator used in the proposed algorithm is based on the semi-Lagrangian method which evaluates the error estimation using solutions at the previous time. Thus, if the primary mesh is fine, additional costs are added to the simulation cost in the h-adaptive methods whereas a coarse mesh leads to an incorrect error estimation in these methods. On the other hand, the enrichment points in the proposed algorithm are adjusted as needed without refining the mesh throughout the time integration process. Therefore, the resulting linear systems maintain the same size and structure during the simulations. Needless to mention that in the h-adaptive methods, the mesh keeps changing causing error accumulations and the computation cost becomes prohibitive due to the multiple interpolations between the adaptive meshes.

5 Numerical results

In this section, two examples of transport-dispersion problems are considered to evaluate the performance of the proposed enriched semi-Lagrangian finite element method. To quantify the method accuarcy, the first example is equipped with a known analytical solution such that the relative L^1-error and L^2-error can be evaluated at time t_n as

$$L^1\text{-error} = \frac{\int_\Omega \left|\boldsymbol{U}_h^n - \boldsymbol{U}_{\text{exact}}^n\right| d\Omega}{\int_\Omega \left|\boldsymbol{U}_{\text{exact}}^n\right| d\Omega}, \qquad L^2\text{-error} = \frac{\left(\int_\Omega \left|\boldsymbol{U}_h^n - \boldsymbol{U}_{\text{exact}}^n\right|^2 d\Omega\right)^{\frac{1}{2}}}{\left(\int_\Omega \left|\boldsymbol{U}_{\text{exact}}^n\right|^2 d\Omega\right)^{\frac{1}{2}}}, \qquad (40)$$

where $\boldsymbol{U}_h^n$ and $\boldsymbol{U}_{\text{exact}}^n$ denote respectively, the numerical and analytical solutions at time t_n. In our simulations, the conjugate gradient solver with incomplete Cholesky decomposition is used to solve the associated linear

Algorithm 3: Adaptive enriched semi-Lagrangian finite element algorithm

Require: $\{\varepsilon_m\}_{m=0,1,\ldots,4}$;
while $t_{n+1} \leq T$ **do**
  Assuming that the previous solution C_h^n is known;
  foreach *element* $\mathcal{T}_k$ **do**
    Evaluate the error indicator $\eta^{n+1}(\mathcal{T}_k)$ according to (36);
    foreach $m \in \{0,1,2,3\}$ **do**
      if $\varepsilon_m \leq \eta^{n+1}(\mathcal{T}_k) \leq \varepsilon_{m+1}$ **then**
        $N_{k,Q} = N_{k,q_m}$;
      end
    end
    Produce the quadrature pair $(\boldsymbol{x}_{q,k}, \omega_{q,k})$, $q = 1, \ldots, N_{k,Q}$;
  end
  Calculate the L^2-projection mass matrix $[\mathbf{M}]$ applying (27);
  foreach *element* $\mathcal{T}_k$ **do**
    foreach *quadrature point* $\boldsymbol{x}_{q,k}$, $q = 1, \ldots, N_{k,Q}$ **do**
      Compute the departure point $\boldsymbol{\chi}_{q,k}^n$;
      Identify the element $\widehat{\mathcal{T}}_{q,k}$ hosting $\boldsymbol{\chi}_{q,k}^n$ using Algorithm 2;
      Evaluate $\widehat{C}_{q,k}^n$ using (26);
    end
  end
  Evaluate the right-hand side entry r_i^n using (27);
  Assemble the right-hand side vector $\mathbf{r}^n$;
  Solve the generated linear system (23);
  Update the concentration C_h^{n+1} at time t_{n+1};
end

systems with a stopping criteria set to 10^{-6}, which is small enough to ensure that the total numerical error is dominated by algorithm truncation error. All the computations are performed on an Intel® Core(TM) i7-7500U @ 2.70GHz with 16 GB of RAM.

5.1 Transport problem with anisotropic dispersion

The main objective of this example is to illustrate the performance of the proposed adaptive enriched semi-Lagrangian finite element method for solving advection-diffusion problems with anisotropic dispersion. Thus, we consider the advection-diffusion equation

$$\frac{\partial C}{\partial t} + \mathbf{U} \cdot \nabla C - \nabla \cdot (\mathbf{D} \nabla C) = 0, \tag{41}$$

to be solved in a squared domain $\Omega = [-1,1] \times [-1,1]$ with the dispersion tensor $\mathbf{D}$ defined by (2)-(3). The boundary and initial conditions are obtained from the analytical solution

$$C(t,x,y) = \frac{C_0}{2\pi\sqrt{|det(\sigma)|}} e^{-\frac{(\mathbf{x}-\mathbf{U}t)^\top \sigma^{-1} (\mathbf{x}-\mathbf{U}t)}{2}}, \tag{42}$$

where $\sigma = \sigma_0^2 \mathbf{I} + 2Dt$, σ^{-1} is the inverse matrix of σ, and $det(\sigma)$ is the determinant of σ. In our simulations, we set $C_0 = 1$, $\sigma_0 = 0.1$, $x_0 = -0.7$, $y_0 = -0.7$, $D_m = 10^{-6}$, $\Delta t = 0.05$ and the velocity field $\mathbf{U} = (0.3, 0.3)^\top$. The goal of this test is to demonstrate the ability of the proposed local enrichment approach to accurately capture sharp gradients in the numerical solution. To achieve this, we employ a multilevel adaptive technique

Table 1: Results obtained for the transport problem with anisotropic dispersion using the adaptive and fixed enrichment semi-Lagrangian methods at time $t = 4.24$ on different structured meshes. CPU times are given in seconds.

$\alpha_L = 10^{-2}$ and $\alpha_T = 10^{-4}$

		Fixed enrichments						Adaptive enrichments					
h	N_Q	L^1-error	L^2-error	Mass	Max C	Min C	CPU	L^1-error	L^2-error	Mass	Max C	Min C	CPU
	12	6.5371E-03	1.2359E-02	0.9889	0.7629	-0.0027	0.93	6.5664E-03	1.2360E-02	0.9890	0.7629	-0.0027	0.76
1	25	2.9787E-03	5.3997E-03	0.9925	0.7561	-0.0016	1.55	3.0253E-03	5.4004E-03	0.9925	0.7561	-0.0016	0.99
32	52	1.9295E-03	2.8780E-03	0.9949	0.7424	-0.0012	2.39	2.0176E-03	2.8781E-03	0.9950	0.7424	-0.0012	1.32
	70	1.3853E-03	2.0924E-03	0.9952	0.7406	-0.0007	3.30	1.4018E-03	2.0932E-03	0.9952	0.7406	-0.0007	1.45
	12	9.2647E-04	1.4526E-03	0.9943	0.7367	-0.0000	3.98	9.5422E-04	1.4538E-03	0.9945	0.7367	-0.0000	1.94
1	25	6.9040E-04	9.8587E-04	0.9960	0.7364	-0.0000	5.35	6.9741E-04	9.8590E-04	0.9961	0.7364	-0.0000	2.21
64	52	4.7452E-04	7.1342E-04	0.9971	0.7326	-0.0000	8.45	4.7668E-04	7.1338E-04	0.9971	0.7326	-0.0000	2.78
	70	3.2390E-04	4.9577E-04	0.9974	0.7314	-0.0000	12.36	3.3345E-04	4.9676E-04	0.9975	0.7314	-0.0000	3.61
	12	2.2370E-04	3.5527E-04	0.9981	0.7318	-0.0000	13.54	2.2370E-04	3.5521E-04	0.9981	0.7318	-0.0000	6.59
1	25	1.3285E-04	2.4077E-04	0.9986	0.7313	-0.0000	23.04	1.3320E-04	2.4062E-04	0.9986	0.7313	-0.0000	7.38
128	52	8.8335E-05	1.2622E-04	0.9991	0.7310	-0.0000	37.30	8.8347E-05	1.2622E-04	0.9991	0.7310	-0.0000	11.77
	70	7.5142E-05	1.0391E-04	0.9994	0.7308	-0.0000	48.81	7.5177E-05	1.0391E-04	0.9994	0.7308	-0.0000	14.16

$\alpha_L = 10^{-4}$ and $\alpha_T = 10^{-2}$

		Fixed enrichments						Adaptive enrichments					
h	N_Q	L^1-error	L^2-error	Mass	Max C	Min C	CPU	L^1-error	L^2-error	Mass	Max C	Min C	CPU
	12	1.9673E-02	2.9218E-02	0.9920	0.8208	-0.0174	0.92	1.9696E-02	2.9220E-02	0.9921	0.8207	-0.0173	0.74
1	25	9.7965E-03	1.5743E-02	0.9934	0.6659	-0.0145	1.53	9.9069E-03	1.5742E-02	0.9934	0.6659	-0.0145	0.98
32	52	7.0162E-03	1.1752E-02	0.9939	0.6844	-0.0025	2.39	7.0260E-03	1.1751E-02	0.9940	0.6844	-0.0025	1.33
	70	4.1301E-03	7.4532E-03	0.9948	0.6873	-0.0012	3.27	4.1492E-03	7.4467E-03	0.9947	0.6874	-0.0012	1.40
	12	2.5157E-03	4.4763E-03	0.9954	0.7158	-0.0012	3.95	2.5189E-03	4.4754E-03	0.9957	0.7158	-0.0012	1.90
1	25	1.0127E-03	1.8298E-03	0.9959	0.7277	-0.0000	5.35	1.0188E-03	1.8297E-03	0.9960	0.7277	-0.0000	2.19
64	52	8.2256E-04	1.4897E-03	0.9962	0.7304	-0.0000	8.44	8.5257E-04	1.4903E-03	0.9964	0.7304	-0.0000	2.75
	70	6.3141E-04	1.0690E-03	0.9967	0.7309	-0.0000	12.35	6.4221E-04	1.0693E-03	0.9968	0.7309	-0.0000	3.61
	12	4.0424E-04	6.2247E-04	0.9959	0.7333	-0.0000	13.52	4.0481E-04	6.2248E-04	0.9959	0.7333	-0.0000	6.60
1	25	2.4152E-04	3.9869E-04	0.9961	0.7321	-0.0000	23.34	2.4286E-04	3.9890E-04	0.9962	0.7321	-0.0000	7.36
128	52	1.6115E-04	2.9422E-04	0.9964	0.7318	-0.0000	37.00	1.6923E-04	2.9436E-04	0.9964	0.7318	-0.0000	11.16
	70	9.0224E-05	1.4515E-04	0.9986	0.7310	-0.0000	49.09	9.1035E-05	1.4516E-04	0.9986	0.7310	-0.0000	14.29

$\alpha_L = 10^{-2}$ and $\alpha_T = 10^{-2}$

		Fixed enrichments						Adaptive enrichments					
h	N_Q	L^1-error	L^2-error	Mass	Max C	Min C	CPU	L^1-error	L^2-error	Mass	Max C	Min C	CPU
	12	2.2325E-03	3.0739E-03	0.9597	0.3559	-0.0000	1.10	2.2483E-03	3.0753E-03	0.9599	0.3559	-0.0000	0.82
1	25	7.9610E-04	1.2513E-03	0.9564	0.3500	-0.0000	1.64	8.0697E-04	1.2502E-03	0.9568	0.3500	-0.0000	0.99
32	52	5.9721E-04	7.3471E-04	0.9598	0.3495	-0.0000	2.59	6.0286E-04	7.3478E-04	0.9600	0.3495	-0.0000	1.36
	70	5.1832E-04	6.1566E-04	0.9649	0.3490	-0.0000	3.53	5.1945E-04	6.1478E-04	0.9652	0.3490	-0.0000	1.42
	12	3.5394E-04	5.1631E-04	0.9631	0.3521	-0.0000	4.14	3.5763E-04	5.1702E-04	0.9632	0.3521	-0.0000	1.92
1	25	1.7910E-04	2.4655E-04	0.9645	0.3500	-0.0000	5.49	1.7752E-04	2.4640E-04	0.9645	0.3500	-0.0000	2.21
64	52	9.4326E-05	1.4447E-04	0.9719	0.3488	-0.0000	9.05	9.4394E-05	1.4446E-04	0.9720	0.3488	-0.0000	2.78
	70	8.8904E-05	1.0820E-04	0.9735	0.3486	-0.0000	12.90	8.9554E-05	1.0828E-04	0.9735	0.3486	-0.0000	3.64
	12	6.9783E-05	1.0080E-04	0.9677	0.3518	0.0000	14.36	6.9793E-05	1.0080E-04	0.9677	0.3518	-0.0000	6.62
1	25	5.4835E-05	7.6791E-05	0.9759	0.3492	0.0000	24.43	5.4733E-05	7.6786E-05	0.9759	0.3492	-0.0000	7.47
128	52	2.9456E-05	4.4734E-05	0.9794	0.3483	0.0000	37.82	2.9439E-05	4.4733E-05	0.9794	0.3483	-0.0000	11.20
	70	1.9836E-05	2.8831E-05	0.9813	0.3484	-0.0000	49.97	1.9757E-05	2.8834E-05	0.9813	0.3484	-0.0000	14.35

Table 2: Results obtained for the transport problem with anisotropic dispersion using the conventional semi-Lagrangian method at time $t = 4.24$ on different structured meshes. CPU times are given in seconds.

$\alpha_L = 10^{-2}$ and $\alpha_T = 10^{-4}$

h	L^1-error	L^2-error	Mass	Max exact	Max C	Min C	CPU
$\frac{1}{32}$	1.81006E-02	3.31305E-02	1.0297	0.7307	0.5986	-0.0015	0.49
$\frac{1}{65}$	3.50488E-03	6.59988E-03	0.9887	0.7307	0.6990	-0.0001	1.78
$\frac{1}{128}$	6.13681E-04	1.13976E-03	0.9928	0.7307	0.7274	-0.0000	8.13

$\alpha_L = 10^{-4}$ and $\alpha_T = 10^{-2}$

h	L^1-error	L^2-error	Mass	Max exact	Max C	Min C	CPU
$\frac{1}{32}$	5.81798E-02	8.67067E-02	1.0258	0.7305	0.4381	-0.0223	0.58
$\frac{1}{65}$	2.18551E-02	3.68212E-02	0.9926	0.7305	0.6117	-0.0228	1.79
$\frac{1}{128}$	5.55469E-03	9.70051E-03	0.9901	0.7305	0.7061	-0.0019	7.44

$\alpha_L = 10^{-2}$ and $\alpha_T = 10^{-2}$

h	L^1-error	L^2-error	Mass	Max exact	Max C	Min C	CPU
$\frac{1}{32}$	1.24522E-02	1.67340E-02	0.9727	0.3477	0.3075	-0.0014	0.60
$\frac{1}{65}$	2.53555E-03	3.38925E-03	0.9552	0.3477	0.3404	-0.0000	1.74
$\frac{1}{128}$	5.46042E-04	7.25754E-04	0.9599	0.3477	0.3478	0.0000	7.22

using $\varepsilon_1 = 0.07$, $\varepsilon_2 = 0.2$, and $\varepsilon_3 = 0.3$. The multilevel enrichments are performed using $N_{k,Q} = 70$, ($N_{k,Q} = 52, N_{k,Q} = 70$), and ($N_{k,Q} = 12, N_{k,Q} = 52, N_{k,Q} = 70$) for single, two, and three-level enrichments, respectively. Notice that the number of quadrature points in elements without enrichment is fixed at $N_{k,Q} = 6$.

For comparison purposes, the relative L^1-error, L^2-error, maximum (Max), minimum (Min), relative mass (Mass), and computational cost (CPU) are calaculated for the considered methods using diffeent meshes, numbers of enrichments and values of longitudinal and transverse dispersion coefficients. Table 1 presents the results obtained by the proposed enriched semi-Lagrangian finite element method using both fixed and adaptive enrichments on different structured meshes and numbers of quadrature points with $(\alpha_L, \alpha_T) = (10^{-2}, 10^{-4})$, $(\alpha_L, \alpha_T) = (10^{-4}, 10^{-2})$, and $(\alpha_L, \alpha_T) = (10^{-2}, 10^{-2})$. Results obtained by the conventional semi-Lagrangian finite element method using the same parameters are presented in Table 2. Note that the selected values of longitudinal and transverse dispersion coefficients cover a wide spectrum of anisotropy in the considered advection-diffusion problem. It should also be noted that the minimum of the analytical solution is 0 while the exact maximum values are included in Table 1 and Table 2. Considering the values of L^1-error and L^2-error in Table 1, it can be observed that the fixed and adaptive approaches yield similar results with insignificant differences for all used meshes and numbers of quadrature points. However, results presented in Table 2 demonstrate that the conventional semi-Lagrangian finite element method is far less accurate than the proposed enriched methods. For example, the conventional semi-Lagrangian method shows negative minimum

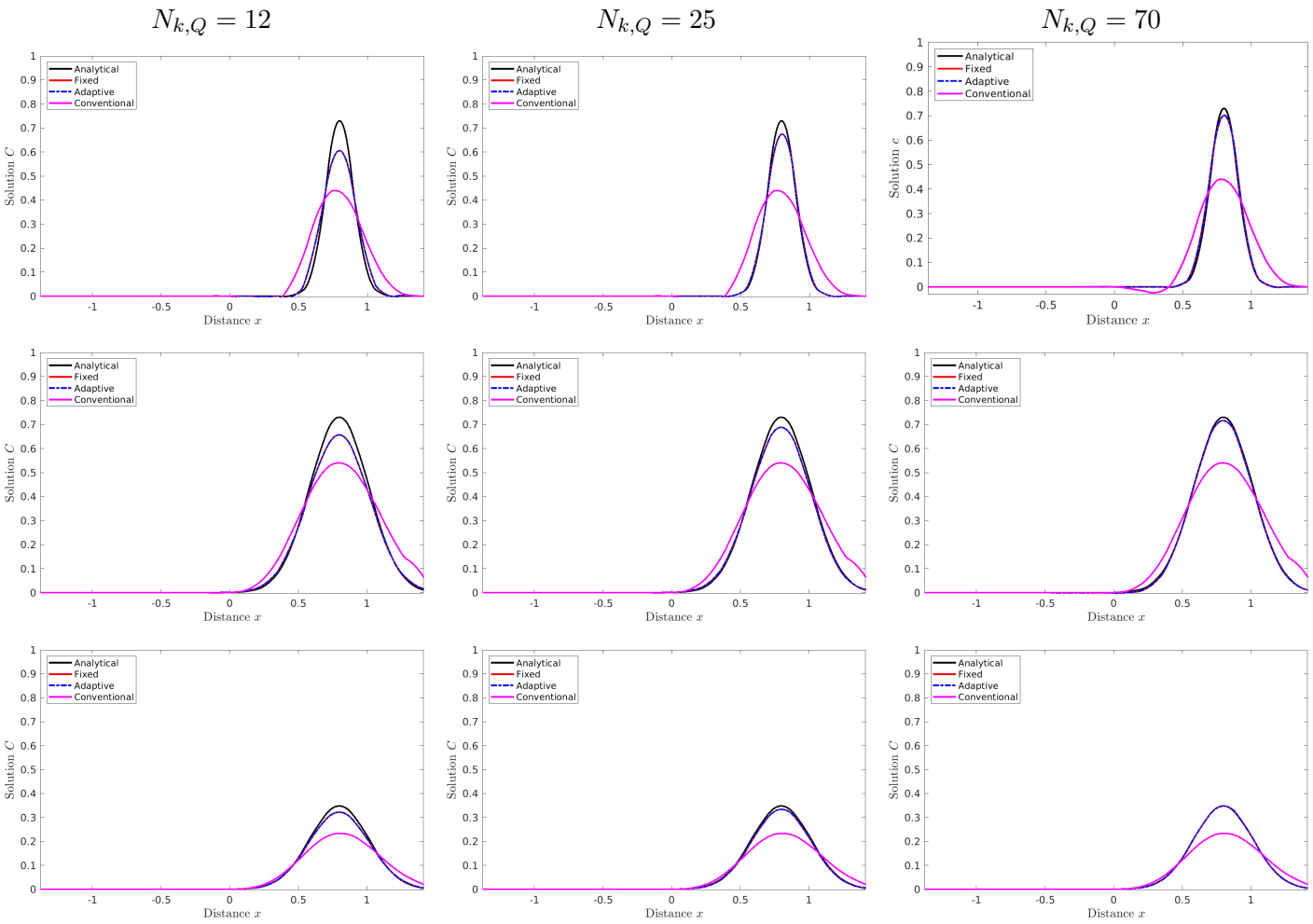

Figure 5: Cross-sections at the main diagonal $x = y$ obtained for the transport problem with anisotropic dispersion at time $t = 4.24$ on a mesh with $h = \frac{1}{32}$ using $(\alpha_L = 10^{-4}, \alpha_T = 10^{-2})$ (first row), $(\alpha_L = 10^{-2}, \alpha_T = 10^{-4})$ (second row) and $(\alpha_L = 10^{-2}, \alpha_T = 10^{-2})$ (third row) using $N_{k,Q} = 12$ (first column), $N_{k,Q} = 25$ (second column) and $N_{k,Q} = 70$ (third column).

values which are avoided by increasing the number of enrichments in the enriched results. Note that a clear improvement in the results obtained using all the considered approaches can be seen in Table 2 and Table 1 when refining the computational mesh or increasing the number of quadrature points in the simulations. For instance, refining the mesh or the quadrature improves the accuracy of the relative mass in Table 1. In terms of the computational costs, it is evident that the fixed enrichment method is more demanding than its adaptive counterpart when comparing CPU time requirements. For the considered dispersion conditions, using the adaptive enriched method, the CPU time required is 63% less than when using the fixed enriched method. Moreover, for a fine mesh with $h = \frac{1}{128}$ and $N_{k,Q} = 70$, the local enrichments reduce the computational costs by 71% compared to the fixed enriched method. It is worth mentioning that, although the computational costs of the conventional semi-Lagrangian method in Table 2 are much lower than those of the proposed enriched method, this reduction in cost is achieved at the expense of the accuracy and stability of the results obtained. To highlight these properties, Figure 5 depicts cross-sections at the main diagonal $x = y$ of the results computed using the considered methods. These results are evaluated on a structured mesh with $h = \frac{1}{32}$ using various numbers of enrichments $N_{k,Q} = 12$, 25, and 70, for $(\alpha_L, \alpha_T) = (10^{-2}, 10^{-4})$, $(\alpha_L, \alpha_T) = (10^{-4}, 10^{-2})$, and $(\alpha_L, \alpha_T) = (10^{-2}, 10^{-2})$. Figure 5 also shows that when the number of enrichments $N_{k,Q}$ is either globally or locally increased in the computational domain, the accuracy of numerical results obtained using the enriched semi-Lagrangian finite element method with both fixed and adaptive enrichments highly improves. On the other

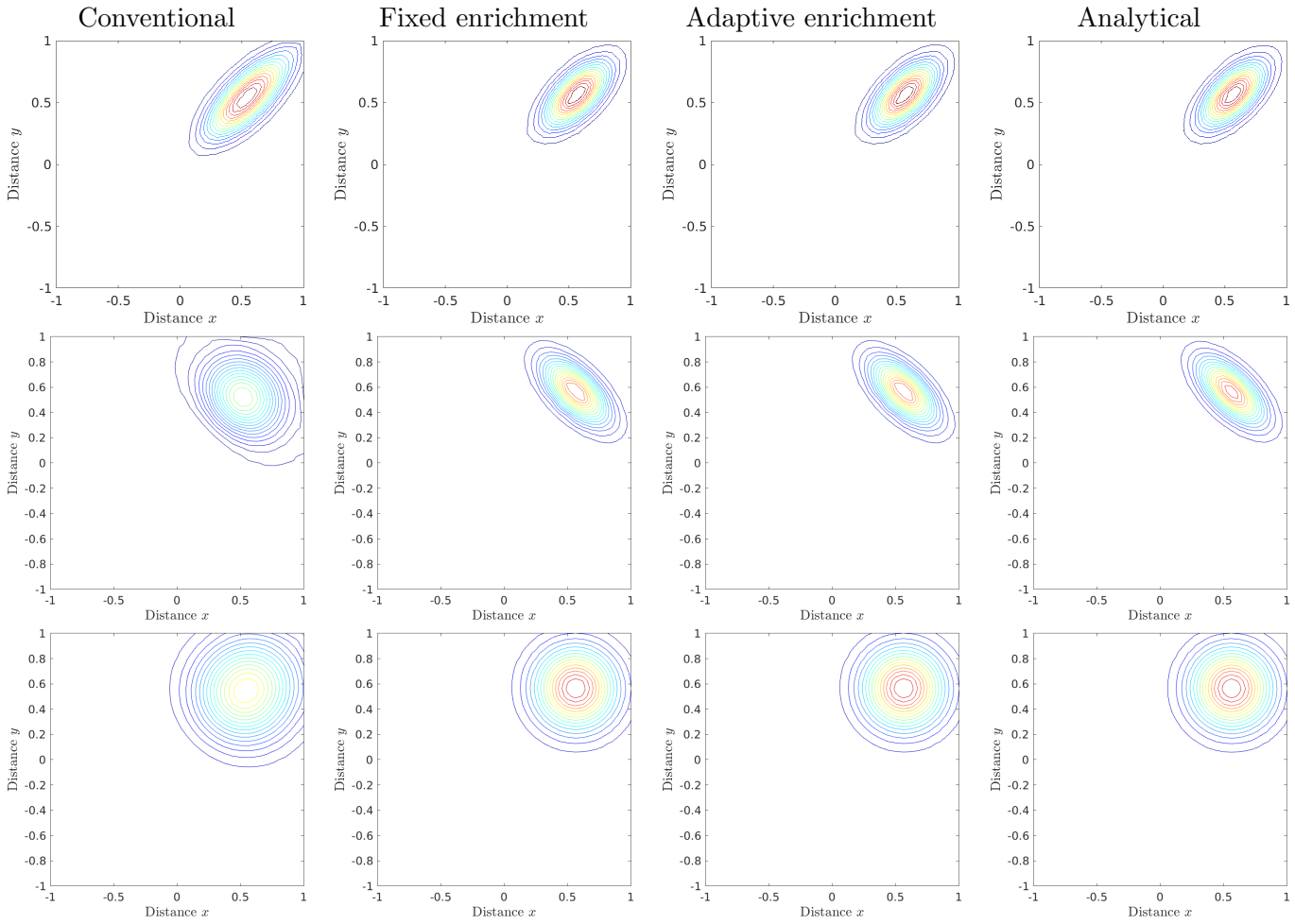

Figure 6: Contour lines obtained for the transport problem with anisotropic dispersion at time $t = 4.24$ using $(\alpha_L = 10^{-4}, \alpha_T = 10^{-2})$ (first row), $(\alpha_L = 10^{-2}, \alpha_T = 10^{-4})$ (second row) and $(\alpha_L = 10^{-2}, \alpha_T = 10^{-2})$ (third row).

hand, the results computed using the conventional semi-Lagrangian finite element method exhibit excessive numerical diffusion as shown in the same figure.

Next, to compare the results obtained using the adaptive and fixed enriched semi-Lagrangian methods to those computed using the conventional semi-Lagrangian method, we display in Figure 6 15 equi-distributed contour lines of the computed solutions at time $t = 4.24$ for the considered dispersion cases $(\alpha_L = 10^{-4}, \alpha_T = 10^{-2})$, $(\alpha_L = 10^{-2}, \alpha_T = 10^{-4})$, and $(\alpha_L = 10^{-2}, \alpha_T = 10^{-2})$. For comparison, contour lines of the analytical solutions are also included in Figure 6. Recall that in the fixed enriched semi-Lagrangian method, each element of the computational mesh is enriched with 70 enrichment points. As expected, the conventional semi-Lagrangian finite element method fails to resolve this transport problem with anisotropic dispersion, while the fixed and adaptive enriched method produce similar results with relatively small differences compared to the analytical solutions. It should be pointed out that the performance of the proposed enriched finite element semi-Lagrangian method is very attractive since the computed solutions remain stable and accurate even when coarse meshes are used without requiring small time steps in the simulations or nonlinear solvers of algebraic equations.

Our final concern with this example is to examine the multilevel adaptive criterion used in the simulations. To this end, Figure 7 exhibits the distribution of quadrature points using single-, two- and three-level adaptive enrichments at four distinct instants namely, $t = 0.46$, 1.74, 2.96 and 4.24 for the considered longitudinal and transverse dispersion coefficients. The aim is to evaluate the effectiveness of the adaptive enriched technique

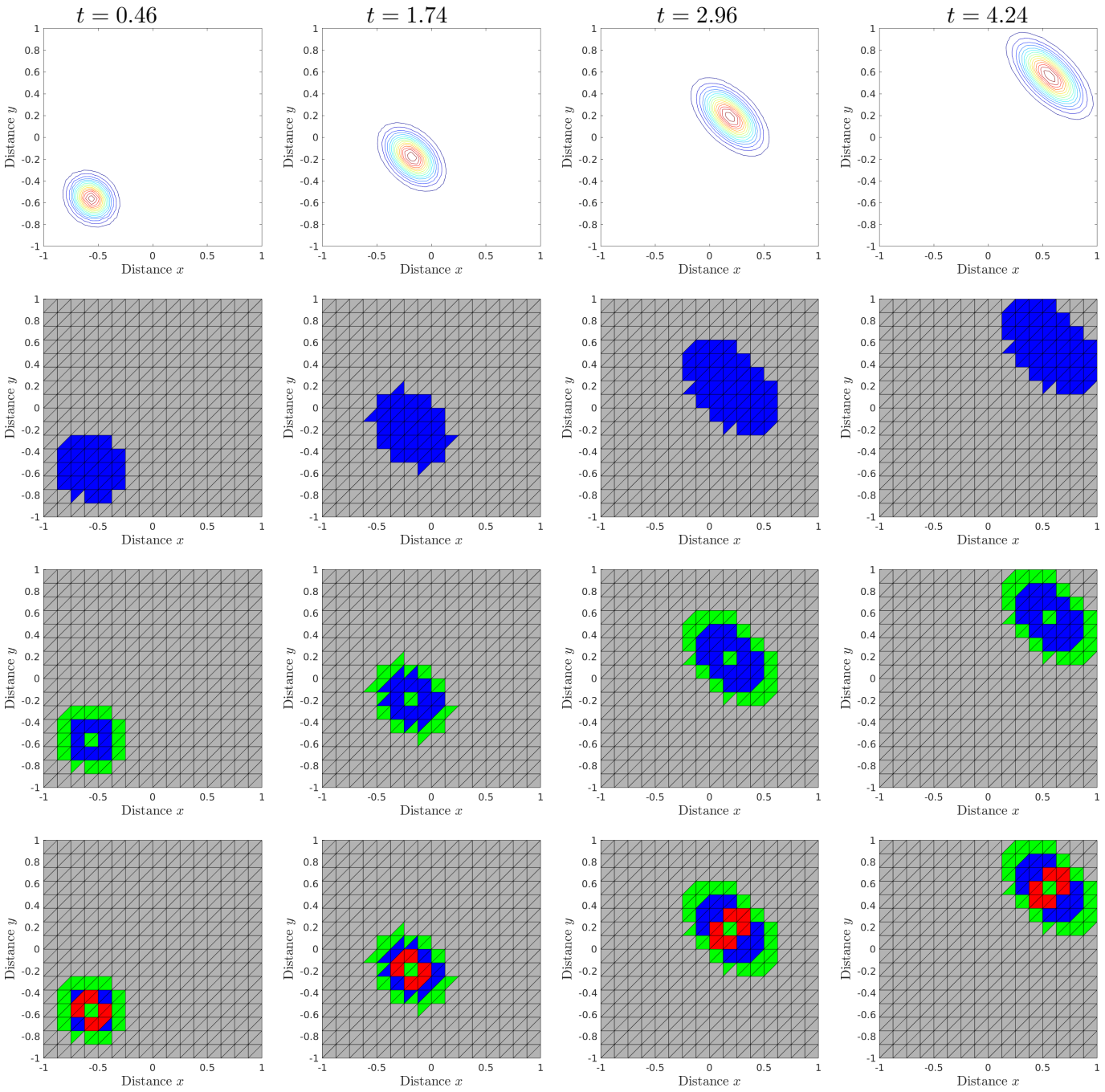

Figure 7: Contour-lines of solution computed using three-level adaptive enrichments (first row), distribution of quadrature points in single-level adaptive enrichments (second row), two-level adaptive enrichments (third row) and three-level adaptive enrichments (fourth row) obtained for the transport problem with anisotropic dispersion at time $t = 0.46$ (first column), $t = 1.74$ (second column), $t = 2.96$ (third column) and $t = 4.24$ (fourth column).

in capturing the transport and dispersion of the Gaussian pulse on a structured mesh with $h = \frac{1}{32}$. Note that different colors are used in Figure 7 to illustrate the levels of enrichment for each element in the computational mesh. This figure also includes contour lines of the numerical solution computed using the three-level enrichment technique. However, contour lines of the numerical solution obtained by the single- and two-level enrichment approaches are not shown because they are identical. Figure 7 demonstrates that the location of quadrature points for all enrichment levels follows the gradients of the solution correctly, and no distortion is recorded in the

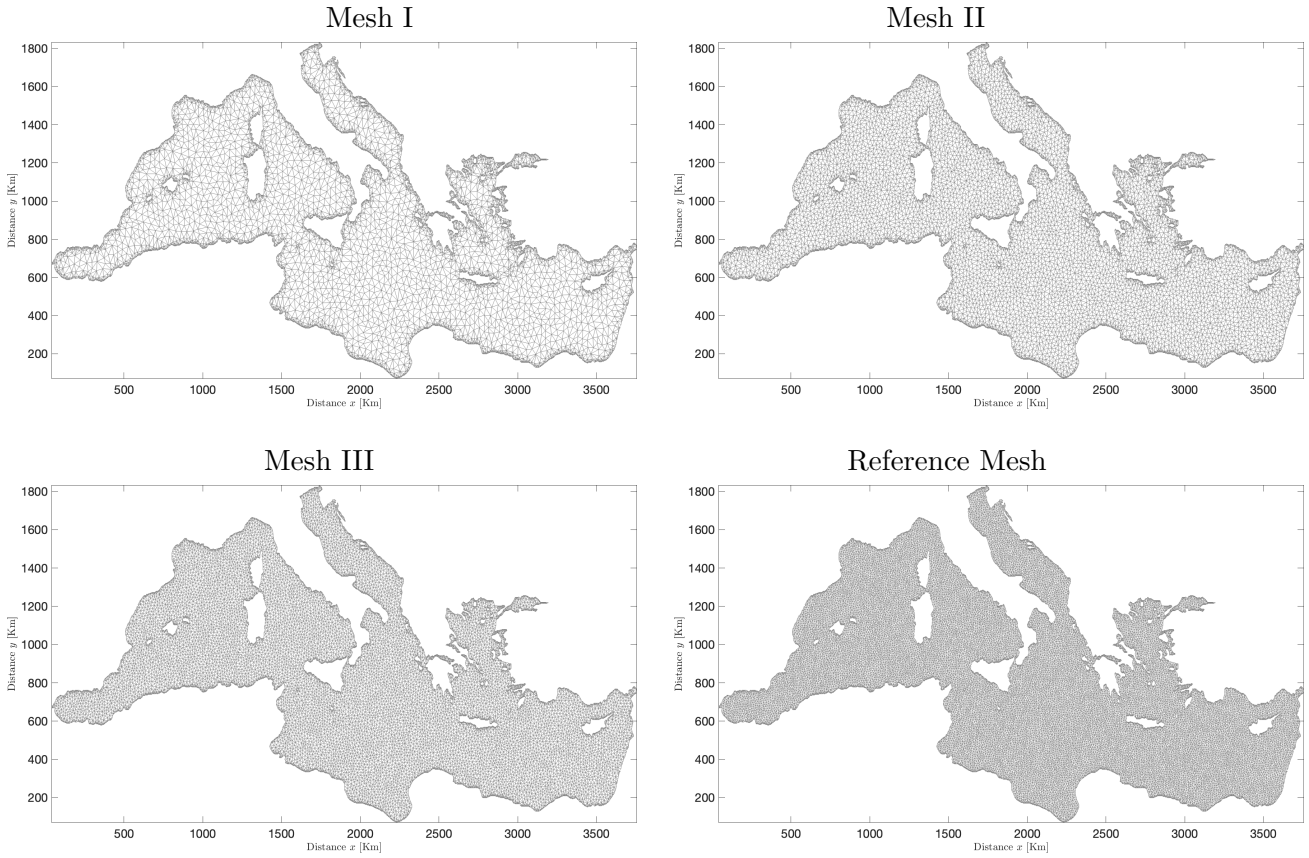

Figure 8: Computational meshes used for the transport and dispersion in the Mediterranean sea.

distribution of these enrichment points. Thus, the proposed adaptive enrichments efficiently resolve transport problems with anisotropic dispersion due to the local refinement of the enrichment points where needed, while the mesh is held fixed during the time integration procedure. This is not the case for the well-established h-adaptive finite element methods as in these methods the computational mesh is not fixed and it needs to be adapted at each time step.

5.2 Transport and dispersion in the Mediterranean sea

This case study solves the equations (1) in the computational domain displayed in Figure 9 using four different unstructured meshes shown in Figure 8. In our simulations, while all coastlines are subjected to non-slip boundary conditions, a well-developed velocity profile with a maximum of $u_\infty = 0.54; m/s$ is imposed at the Gibraltar entrance of the sea. This profile corresponds to the annual mean of the Atlantic input flux and it is comparable to the main semidiurnal component M_2, see for instance [3, 36, 24]. On the other hand, the advection-dispersion equation in (1) is equipped with a pollution release rate of $S = 1$ tones/hour at the Gibraltar entrance. On the remaining boundaries, free-boundary conditions are used for the concentration solution. In our simulations for this problem, we consider the enriched semi-Lagrangian finite element methods with fixed and three-level adaptive enrichments. In the adaptive enrichments, we use the tolerances $\varepsilon_1 = 0.08$, $\varepsilon_2 = 0.15$, and $\varepsilon_3 = 0.32$, and the initial number of quadrature points is set to $N_{k,Q} = 6$ in each element without enrichments. This number is refined according to Algorithm 3 using ($N_{k,Q} = 12$, $N_{k,Q} = 25$, $N_{k,Q} = 52$). For all results reported in this section, the Coriolis parameter $f = 8.55 \times 10^{-5}$ $/s$, the bottom friction coefficient $\gamma = 0.012$ $s/m^{1/3}$, the kinematic viscosity $\nu = 1.18$ m^2/s, and the wind stress $\tau = 1.5$ N/m^2. We consider both continuous and instantaneous releases of pollutants for this problem, using a fixed time step $\Delta t = 45$ min. We also compare the results obtained using the proposed multilevel adaptive enriched semi-Lagrangian finite element method to those obtained using the conventional semi-Lagrangian finite element method. The main

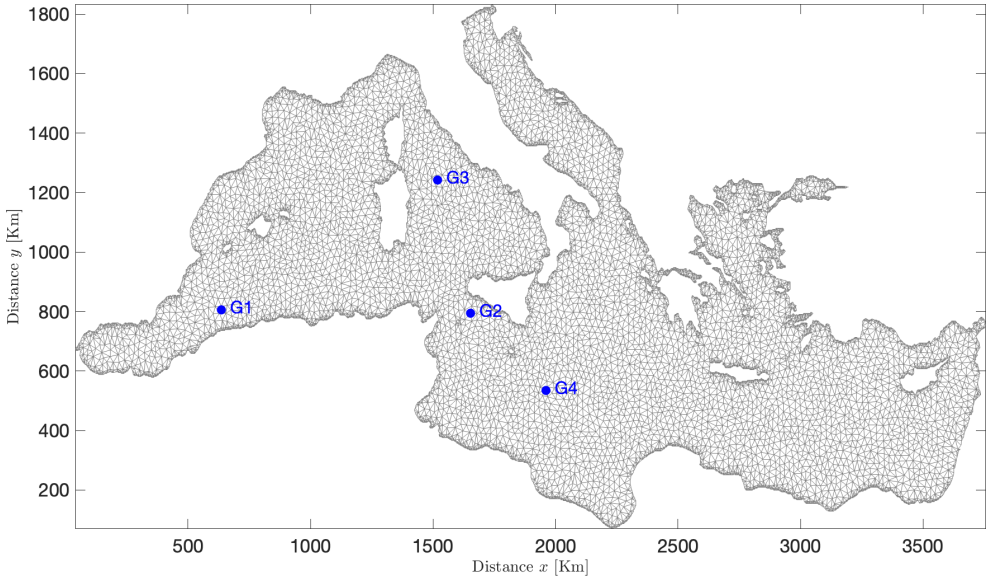

Figure 9: Location of the considered four Gauges G1, G2, G3, and G4 located in the Mediterranean sea and used for monitoring the concentration.

objective of this example is to validate the capability of the proposed numerical algorithm to accurately handle complex geometries and to develop a robust method for studying transport and dispersion of pollutants in the Mediterranean sea.

Table 3: Mesh statistics, values of the kinetic energy and the total averaged concentration computed using the adaptive enrichment method at time $t = 13$ days, and computational times for the considered meshes for the transport and dispersion in the Mediterranean sea using continuous and instantaneous releases. CPU times are given in seconds.

				Continuous release			Instantaneous release		
	# ele	# P_2 nodes	# P_1 nodes	Energy	Averg C	CPU	Energy	Averg C	CPU
Mesh I	14217	33759	9757	3.8725E+04	1.38841	200	3.7667E+04	1.17725E-01	198
Mesh II	18827	42979	12062	4.4322E+04	1.52454	559	4.1579E+04	1.47203E-01	547
Mesh III	27376	60078	16337	4.4877E+04	1.55016	797	4.1879E+04	1.51944E-01	785
Reference	47325	100015	26330	4.5303E+04	1.57056	2941	4.2089E+04	1.54988E-01	2804

We first perform a mesh convergence study for this problem using four different unstructured meshes of triangular finite elements as displayed in Figure 8. The corresponding statistics of these meshes in terms of numbers of elements, P_2 nodes and P_1 nodes are listed in Table 3. We also monitor the concentration at four gauges G1, G2, G3, and G4 located in the Mediterranean sea at $(611.6; km, 816.2; km)$, $(1628.8; km, 805.1; km)$, $(1494; km, 1251.8; km)$, and $(1936.7; km, 545.2; km)$, respectively, see Figure 9 for an illustration. The time evolution of the concentration at the considered gauges computed using the adaptive enriched semi-Lagrangian finite element method on the considered meshes at time $t = 13$ days is presented in Figure 10. It should be noted that a reference solution calculated using the fixed enriched semi-Lagrangian method on the Reference mesh using $N_{k,Q} = 70$ is also included in this comparaison. It is clear that results obtained for both continuous and instantaneous releases using the coarse Mesh I are less accurate than those calculated using the Reference mesh. By refining the density of elements in the computational domain, results obtained on Mesh III and Reference mesh are nearly similar. This confirms the mesh convergence in the proposed multilevel adaptive enriched semi-Lagrangian finite element method for this example. To quantify this mesh convergence study, we also summarize in Table 3 the computational times, the averaged concentration volume and the total kinetic

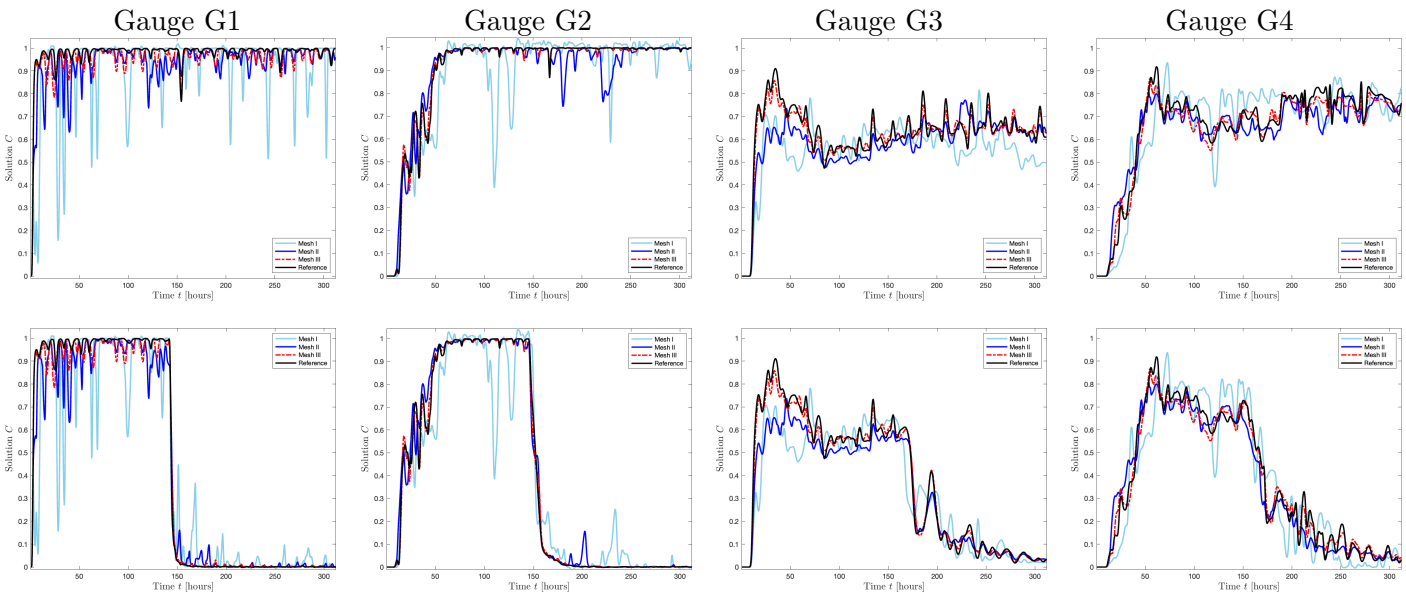

Figure 10: Time evolution of the concentration obtained using the adaptive enrichment method at the gauges $G1$, $G2$, $G3$ and $G4$ using different meshes for the transport and dispersion in the Mediterranean sea using continuous release (first row) and instantaneous release (second row).

energy obtained at time $t = 13$ days using the considered meshes for both continuous and instantaneous releases. Here, the total kinetic energy and the averaged concentration volume are defined as

$$\frac{1}{2}\int_{\Omega}\left(U^2(t,\boldsymbol{x})+V^2(t,\boldsymbol{x})\right)\,d\boldsymbol{x} \qquad \text{and} \qquad \frac{1}{|\Omega|}\int_{\Omega}C(t,\boldsymbol{x})\,d\boldsymbol{x},$$

respectively. It can be seen from Table 3 that there are slight differences between the results obtained for the total kinetic energy and the averaged concentration volume on Mesh II, Mesh III, and the Reference mesh. For example, the discrepancies in values of the total kinetic energy and the averaged concentration volume using continuous release on Mesh II and the Reference mesh are less than 2.16% and 1.30%, respectively. These values become less than 0.94% and 1.93% on Mesh III and Reference mesh. On the other hand, the computational times needed in the proposed multilevel adaptive enriched semi-Lagrangian finite element method on Mesh III are 73% less than the one required on Reference mesh whereas, the use of Mesh II reduces this value to 81%. Therefore, Mesh III is believed to be appropriate to obtain numerical results free of grid effects. Hence, the results presented hereafter are based on Mesh III.

Next we display in Figure 11 the velocity fields, concentration snapshots and distribution of quadrature points using three-level adaptive enrichments for the continuous release at time $t = 1$, 2, 4, 6, 8, 11 and 13 days. Those results obtained for the case of instantaneous release are displayed in Figure 12. It is clear that the proposed multilevel adaptive enriched semi-Lagrangian finite element method successfully captures the complex concentration and flow structures, comparing the decrease and increase of the strengths of flow vortices with time in these results. Inside these vortices, there is a more complex vortex pattern which has been accurately resolved. Moreover, these vortices and the high gradients in transported concentration are well captured by the proposed adaptive approach and the quadrature points are distributed in regions with the specified tolerances according to the desired three-level enrichment algorithm in the computational domain. Note that three distinct colors are used to represent the three-level adaptive distribution of quadrature points namely green, blue, and red for the first-, second-, and third-level enrichments, respectively. A gray color is used for elements with smooth solutions and low concentration gradients. As expected, denser quadrature points are generated for the three-level adaptive enrichments. It can be shown from the same figures that under the considered flow and dispersion conditions, the pollution is mainly transported towards the African coast. Obviously, the concentration follows the stream induced by the mean flow entering the Mediterranean

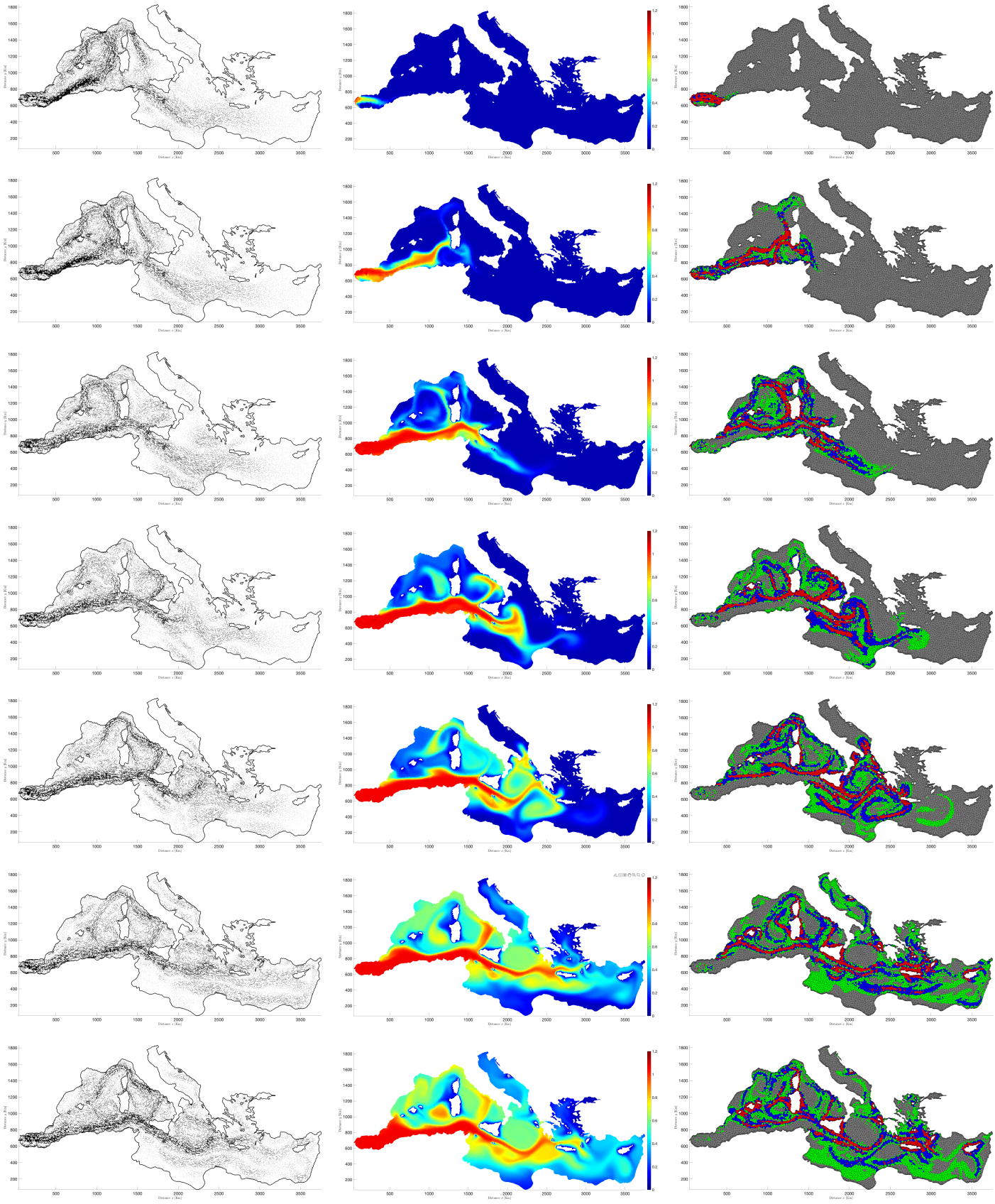

Figure 11: Velocity fields (first column), solution snapshots (second column) and distributions of quadrature points (third column) for the transport and dispersion in the Mediterranean sea at time $t = 1$ day (first row), $t = 2$ days (second row), $t = 4$ days (third row), $t = 6$ days (fourth row), $t = 8$ days (fifth row), $t = 11$ days (sixth row) and $t = 13$ days (seventh row) using continuous release.

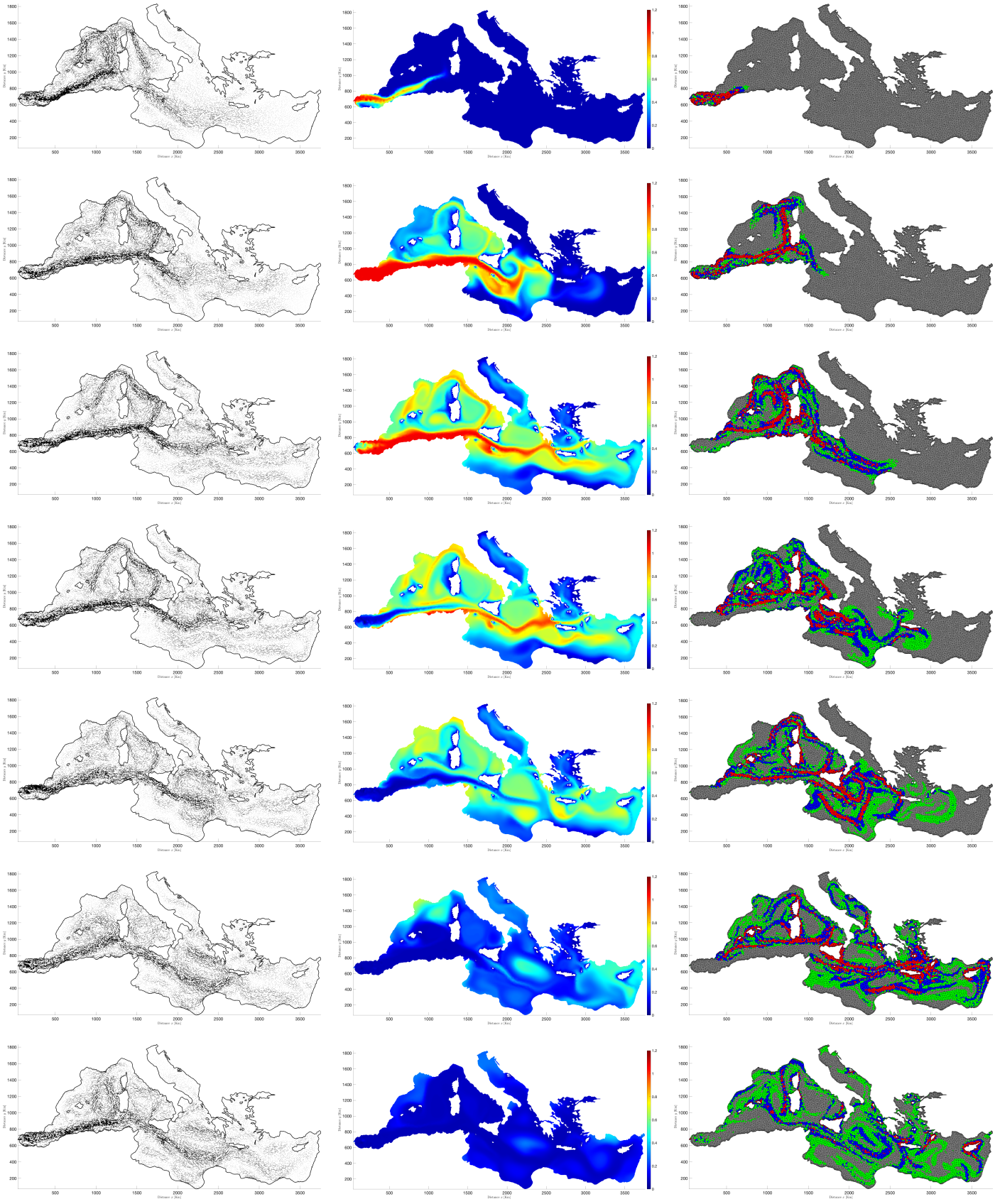

Figure 12: Same as Figure 11 but using instantaneous release.

Table 4: Results for the kinetic energy, the total averaged concentration and computational times obtained using the considered methods at time $t = 13$ days for the transport and dispersion in the Mediterranean sea using continuous and instantaneous releases. CPU times are given in seconds.

	Continuous release			Instantaneous release		
	Energy	Averg C	CPU	Energy	Averg C	CPU
Conventional	3.26888E+04	1.06962	223	3.02316E+04	1.03272E-01	211
Fixed	4.48921E+04	1.56411	2043	4.189712+04	1.53236E-01	1963
Adaptive	4.48771E+04	1.55016	797	4.18790E+04	1.51944E-01	785
Reference	4.53032E+04	1.57059	2941	4.20897E+04	1.54988E-01	2804

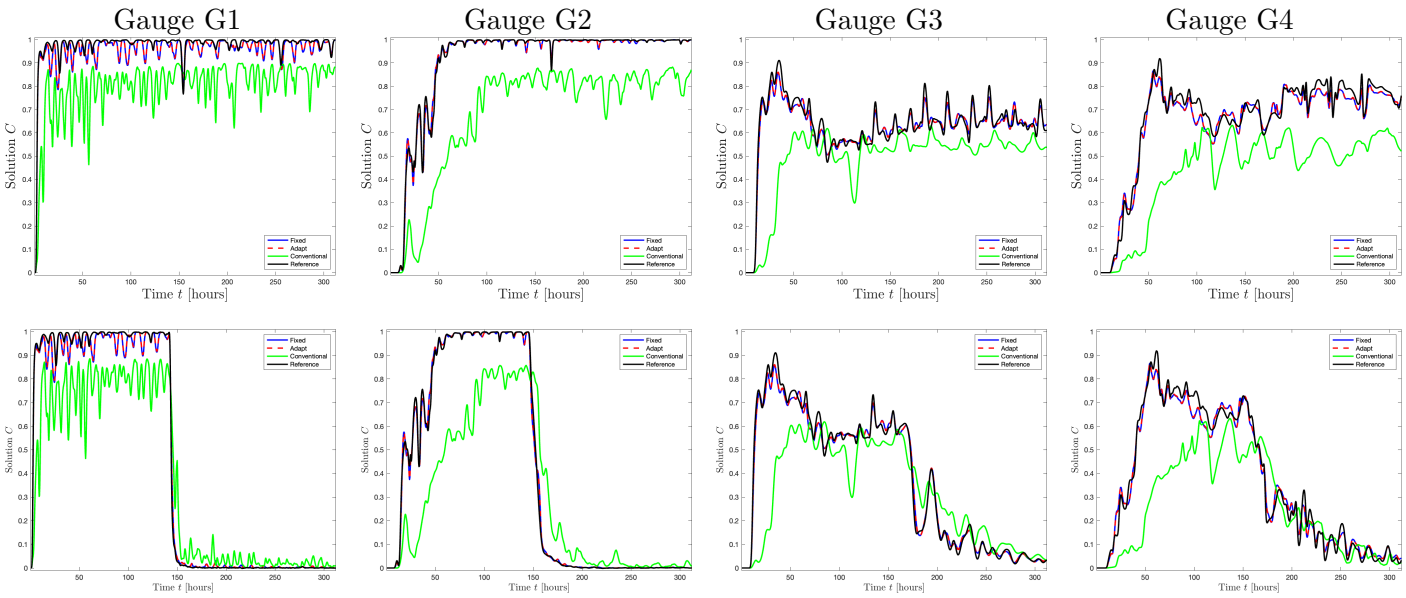

Figure 13: Time evolution of the concentration computed using the considered methods at the gauges $G1$, $G2$, $G3$ and $G4$ for the transport and dispersion in the Mediterranean sea using continuous release (first row) and instantaneous release (second row)

sea through the Gibraltar strait. During its dispersion, the concentration alerts the flow structure developing recirculation zones with different frequencies and magnitudes in many areas in the Mediterranean sea. In summary, the incompressible flow field is resolved relatively well, the concentration transported is captured accurately. All these properties have been obtained using time steps bigger than those necessary for Eulerian-based finite element approach in incompressible convection-dominated flows.

Comparisons between results obtained using the conventional method, fixed enrichments, and adaptive enrichments have also been carried out for this test example. The averaged concentration volume, computational times, and total kinetic energy obtained at time $t = 13$ days using Mesh III for both continuous and instantaneous release are presented in Table 4. Reference solutions computed using fixed enrichments with $N_{k,Q} = 70$ on the Reference mesh are also included in this table. It can be seen that both fixed and adaptive enrichment methods produce similar results which are closer to the reference solutions than those computed using the conventional method. In terms of computational cost, it is evident that the adaptive enriched method is less expensive compared to the fixed enriched method which requires high CPU times. For the considered flow and transport conditions, the CPU time needed by the adaptive enriched method is roughly 60% less than the CPU time required by the fixed enriched method. Needless to mention that, when compared to the enriched techniques on the same mesh, the conventional approach has a lower computational cost, but its overall accuracy

and stability are much inferior to those achieved by the enrichment methods.

Finally, Figure 13 illustrates the time evolution of the monitored concentration at the gauges G1, G2, G3, and G4 shown in Figure 9 using fixed enrichments with $N_{k,Q} = 52$, three-level adaptive enrichments, and the conventional method. Reference solutions obtained using fixed enrichments with $N_{k,Q} = 52$ on Reference mesh are also included in these plots. For both types of release, the concentration exhibits fluctuations in its time evolution at the considered gauges but remains bounded between 0 and 1 as expected. From the same plots, it can be seen that the results obtained using the conventional approach suffer from numerical diffusion. As in the previous results, there are negligible differences between the results obtained using the fixed and adaptive enriched methods. For the considered flow, transport and dispersion conditions, it can be clearly demonstrated that the proposed multilevel adaptive enriched semi-Lagrangian finite element method successfully captures the complicated concentration and flow structures in the Mediterranean sea. It should be stressed that the obtained computational results should be compared to observational data of real sea-surface pollution dispersion in the Mediterranean sea. However, there is currently no data available to make this comparison and as a result for the time being, we can simply run simulations and verify that the findings are credible and consistent.

6 Concluding remarks

In the present study, a novel adaptive enriched semi-Lagrangian finite element method is proposed for the modelling and simulation and modeling of transport and dispersion of pollutants in the Mediterranean sea. The mathematical model consists of a class of barotropic ocean equations with friction terms, bathymetric forces, Coriolis and wind stresses coupled to an advection-diffusion equation with anisotropic dispersion tensor and source terms. The proposed method combines the modified method of characteristics, finite element discretization, a coupled projection scheme based on a rotational pressure correction, and an adaptive L^2-projection using quadrature rules. The implementation of multilevel adaptive enrichments further improves the accuracy and efficiency of the numerical solution without the need for mesh refinement throughout the time integration process. Computational results obtained for verification examples supported the conclusion that the proposed method can effectively be used to resolve flow and transport features for transport and dispersion of pollutants in the Mediterranean sea. In particular, an example with known analytical solutions for transport problems with anisotropic dispersion have been used to demonstrate the good performance of the developed method. In the considered numerical simulations, the proposed methodology successfully recovered the flow and dispersion characteristics with significantly fewer degrees of freedom compared to the conventional finite element method. This results in a considerable decrease in computing requirements without compromising the accuracy of the solution. Future work will focus on developing highly accurate error estimates such as a posteriori error estimates, to further improve the adaptive enrichments. In addition, following the arguments used for analysis of convergence and stability of the conventional semi-Lagrangian finite element method for the incompressible Navier-Stokes equations in [14] and for the coupled Darcy-transport problems in [37], it is also possible to establish theoretical analysis of convergence and stability for the adaptive enriched semi-Lagrangian finite element method proposed in this study. Results on this analysis will be reported in the near future. Overall, the proposed adaptive enriched semi-Lagrangian finite element method provides an effective and efficient approach for modeling and simulating pollution transport in the Mediterranean sea.

Acknowledgments and funding

We would like to thank the Moroccan Ministry of Higher Education, Scientific Research and Innovation and the OCP Foundation who funded this work through the APRD research program.

Data availability

The data underlying the results can be obtained from the corresponding author on a reasonable request.

Conflict of interest

The authors have no competing interests to declare that are relevant to the content of this paper.

References

[1] B. R. Ahrabi, W. K. Anderson, and J. C. Newman III. An adjoint-based *hp*-adaptive stabilized finite-element method with shock capturing for turbulent flows. *Comput. Methods Appl. Mech. Engrg.*, 318:1030–1065, 2017.

[2] A. Allievi and R. Bermejo. A generalized particle search–locate algorithm for arbitrary grids. *Journal of Computational Physics*, 132(2):157–166, 1997.

[3] J. Almazán, H. Bryden, T. Kinder, and G. Parrilla. *Seminario Sobre la Oceanografía Física del Estrecho de Gibraltar*. SECEG, Madrid, 1988.

[4] N. Berrojalbiz, J. Dachs, S. D. Vento, M. J. Ojeda, M. C. Valle, J. Castro-Jiménez, G. Mariani, J. Wollgast, and G. Hanke. Persistent organic pollutants in Mediterranean seawater and processes affecting their accumulation in plankton. *Environmental Science & Technology*, 45, 05 2011.

[5] W. Boscheri. A space-time semi-Lagrangian advection scheme on staggered voronoi meshes applied to free surface flows. *Computers & Fluids*, 202:104503, 2020.

[6] C. J. Budd and A. Iserles. Geometric integration: Numerical solution of differential equations on manifolds. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 357(1754):945–956, 1999.

[7] H. Chen, K. Li, and S. Wang. A dimension split method for the incompressible Navier-Stokes equations in three dimensions. *Int. J. Numer. Methods. Fluids.*, 73(5):409–35, 2013.

[8] P. De Sampaio, P. R. M. Lyra, K. Morgan, and N. P. Weatherill. Petrov-Galerkin solutions of the incompressible Navier-Stokes equations in primitive variables with adaptive remeshing. *Comput. Methods Appl. Mech. Engrg.*, 106(1-2):143–178, 1993.

[9] P. De Sampaio, P. R. M. Lyra, K. Morgan, and N. P. Weatherill. Petrov-Galerkin solutions of the incompressible Navier-Stokes equations in primitive variables with adaptive remeshing. *Comput. Methods Appl. Mech. Engrg.*, 106(1-2):143–178, 1993.

[10] S. Deparis, M. Deville, F. Menghini, L. Pegolotti, and A. Quarteroni. Application of the Rosenbrock methods to the solution of unsteady 3D incompressible Navier-Stokes equations. *Computers & Fluids*, 179:112–122, 2019.

[11] S. Deparis and A. Løvgren. Stabilized reduced basis approximation of incompressible three-dimensional Navier-Stokes equations in parametrized deformed domains. *Journal of Scientific Computing*, 50:198–212, 2012.

[12] D. Dunavant. High degree efficient symmetrical gaussian quadrature rules for the triangle. *Int. J. Numer. Meth. Engng.*, 21(6):1129–1148, 1985.

[13] M. El-Amrani, A. Ouardghi, and M. Seaid. Enriched Galerkin-characteristics finite element method for Incompressible Navier–stokes equations. *SIAM Journal on Scientific Computing*, 43(2):A1336–A1361, 2021.

[14] M. El-Amrani and M. Seaid. Convergence and stability of finite element modified method of characteristics for the incompressible Navier-Stokes equations. *Journal of Numerical Mathematics*, 15(2):101–135, 2007.

[15] M. El-Amrani and M. Seaid. Numerical simulation of natural and mixed convection flows by Galerkin-characteristic method. *Int. J. Numer. Meth. Fluids.*, 53(12):1819–1845, 2007.

[16] M. El-Amrani and M. Seaid. An essentially non-oscillatory semi-Lagrangian method for tidal flow simulations. *Int. J. Numer. Meth. Engng.*, 81(7):805–834, 2010.

[17] M. El-Amrani and M. Seaid. An L^2-projection for the Galerkin-characteristic solution of incompressible flows. *SIAM Journal on Scientific Computing*, 33(6):3110–3131, 2011.

[18] M. El-Amrani and M. Seaid. A finite element semi-Lagrangian method with L^2 interpolation. *Int. J. Numer. Meth. Engng.*, 90(12):1485–1507, 2012.

[19] M. El Haddad, Y. Belhamadia, J. Deteix, and D. Yakoubi. A projection scheme for phase change problems with convection. *Computers & Mathematics with Applications*, 108:109–122, 2022.

[20] A. El-Kacimi and O. Laghrouche. Numerical modelling of elastic wave scattering in frequency domain by the partition of unity finite element method. *Int. J. Numer. Meth. Engng.*, 77:1646–1669, 2009.

[21] G. Gabrielides. Pollution of the mediterranean sea. *Water Science & Technology Water Supply*, 32, 1995.

[22] J.-L. Guermond and A. Salgado. A splitting method for incompressible flows with variable density based on a pressure poisson equation. *Journal of Computational Physics*, 228(8):2834–2846, 2009.

[23] F. Kang and S. Zai-jiu. Volume-preserving algorithmsfor source-free dynamical systems. *Numerische Mathematik*, 71(4):451–463, 1995.

[24] P. T. L. Houpert, S. S. X. Durrieu de Madron, C. E. F. D'Ortenzio, and H. Lavigne. Seasonal cycle of the mixed layer, the seasonal thermocline and the upper-ocean heat storage rate in the Mediterranean sea derived from observations. *Progress in Oceanography*, 132:333–352, 03 2015.

[25] C. Liu. Numerical solution of three-dimensional Navier-Stokes equations by a velocity-vorticity method. *Int. J. Numer. Methods. Fluids.*, 35:533–557, 2001.

[26] D. Lo, K. Murugesan, and D. Young. Numerical solution of three-dimensional velocity-vorticity Navier-Stokes equations by finite difference method. *Int. J. Numer. Methods. Fluids.*, 47:1469–1487, 2005.

[27] P. L. Lourd. *Oil pollution in the Mediterranean Sea*, volume 6. Royal Swedish Academy of Sciences, 1977.

[28] A. Malagó, F. Bouraoui, B. Grizzetti, and A. D. Roo. Modelling nutrient fluxes into the Mediterranean sea. *Journal of Hydrology: Regional Studies*, 22:100592, 2019.

[29] MedECC. Climate and Environmental Change in the Mediterranean Basin - Current Situation and Risks for the Future. First Mediterranean Assessment Report, Nov. 2020.

[30] C. Min and F. Gibou. A second order accurate projection method for the incompressible Navier-Stokes equations on non-graded adaptive grids. *J. Comp. Physics*, 219(2):912–929, 2006.

[31] A. Ouardghi, M. El-Amrani, and M. Seaid. An enriched Galerkin-characteristics finite element method for convection-dominated and transport problems. *Applied Numerical Mathematics*, 167:119–142, 2021.

[32] A. Ouardghi, M. El-Amrani, and M. Seaid. An adaptive enriched semi-lagrangian finite element method for coupled flow-transport problems. *Computers & Fluids*, 240:105474, 2022.

[33] S. K. Patro, R. P. Selvam, and H. Bosch. Adaptive h-finite element modeling of wind flow around bridges. *Engineering Structures*, 48:569–577, 2013.

[34] S. Popinet. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *J. Comp. Physics*, 190(2):572–600, 2003.

[35] Z. Qian and J. Zhang. Implicit preconditioned high-order compact scheme for the simulation of the three-dimensional incompressible Navier-Stokes equations with pseudo-compressibility method. *Int. J. Numer. Methods. Fluids.*, 69:1165–1185, 2012.

[36] A. Robinson, W. Leslie, A. Theocharis, and A. Lascaratos. Mediterranean sea circulation. *Ocean currents*, 1:19, 2001.

[37] L. Salhi, M. El-Amrani, and M. Seaid. A priori error estimates for a semi-lagrangian unified finite element method for coupled darcy-transport problems. *Numerical Methods for Partial Differential Equations*, 39(4):3441–3472, 2023.

[38] K. Selim, A. Logg, and M. G. Larson. An adaptive finite element splitting method for the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 209:54–65, 2012.

[39] M. Shadi Mohamed, M. Seaid, J. Trevelyan, and O. Laghrouche. A partition of unity fem for time-dependent diffusion problems using multiple enrichment functions. *Int. J. Numer. Meth. Engng.*, 93(3):245–265, 2013.

[40] C. Subich, K. Lamb, and M. Stastna. Simulation of the Navier-Stokes equations in three dimensions with a spectral collocation method. *Int. J. Numer. Methods. Fluids.*, 73:103–129, 2013.

[41] N. Tanaka and T. Kitayama. Symplectic semi-Lagrangian schemes for computational fluid dynamics. *International Journal of Computational Fluid Dynamics*, 18(4):303–308, 2004.

[42] F. Trincardi, F.and Francocci, C. Pellegrini, M. d'Alcalà, and M. Sprovieri. The mediterranean sea in the anthropocene. In *Oceanography of the Mediterranean Sea*, pages 501–553. Elsevier, 2023.

[43] R. Waldman and H. Giordani. Ocean barotropic vorticity balances: theory and application to numerical models. *Journal of Advances in Modeling Earth Systems*, 15(4):e2022MS003276, 2023.

[44] F. Wang, M. Ling, W. Han, and F. Jing. Adaptive discontinuous Galerkin methods for solving an incompressible Stokes flow problem with slip boundary condition of frictional type. *J. Comp. Applied Math.*, 371:112700, 2020.

[45] M. Wang and T. Sheu. On a compact mixed-order finite element for solving the three-dimensional incompressible Navier-Stokes equations. *Int. J. Numer. Methods. Fluids.*, 25:513–522, 1997.

[46] S. Wille. The three-dimensional prolonged adaptive unstructured finite element multigrid method for the Navier-Stokes equations. *Int. J. Numer. Methods. Fluids.*, 25:371–392, 1997.