**POLITECNICO DI MILANO**

School of Industrial and Information Engineering

Master of Science in Energy Engineering

# Adaptive methods and code benchmarking for the simulation of solute transport in heterogeneous porous media

Supervisor: Prof. Giovanni Michele PORTA

Assistant Supervisor: Prof. Alberto GUADAGNINI

External Supervisor: Prof. Philippe ACKERER

Master Thesis of:

Simone SALVI, 852425

Academic year 2016 - 2017

Simone Salvi: *Adaptive methods and code benchmarking for the simulation of solute transport in heterogeneous porous media* | Master Thesis in Energy Engineering, Politecnico di Milano.

Politecnico di Milano:
www.polimi.it

School of Industrial and Information Engineering:
www.ingindinf.polimi.it

# Ringraziamenti

Mi è sempre piaciuto associare l'università ad una palestra di vita, nella quale ambito accademico e personale si intrecciano minuto per minuto. Mantenendo la metafora sportiva, questi sei anni sono stati un duro, ma stimolante allenamento e come sempre lavorare in gruppo offre maggiori soddisfazioni.

Abbandonata questa premessa filosofica, desidero ringraziare:

**Prof. Giovanni Porta, Philippe Ackerer, Carlos David Perez Segarra** perchè mi hanno accompagnato nello svolgimento e nella stesura di questa tesi e hanno avuto un ruolo determinante nel guidarmi all'interno del mondo dei mezzi porosi;

**Bruno, Marco, Spada, Ste, Massi, Fra, Dalmo** perchè compagni di studio (o quasi), di lunghi viaggi in treno, di allegre costinate e si spera ancora di vacanze in Sardegna;

**Luca, Garnik, Marie** perchè un Erasmus non è tale senza dei coinquilini matti;

**Nata, Mirta, Paza, Lozzo, Colo, Claudio** perchè ottimi compagni di bevute;

**Tutta la mia famiglia e l'Ing. Capelli** perchè hanno sempre creduto in me e mi hanno dato la possibilità di fare svariate esperienze all'estero, senza mai controbbatere;

**Ramona** perchè senza di lei non sarebbero stati anni così pieni di emozioni (*always off the beaten path*).

*Milan, April 2018*

*Dùm loquimùr, fùgerit ìnvidia aètas:*
*Carpe Diem,*
*quàm minimùm crèdula pòstero*

# Extended Abstract

According to [1] and [2], transport phenomena in porous media involve the description of fluid flow and solute transport. The former abides by Darcy's law and accomplishes the continuity constraint, while the latter, assuming a chemically inactive solute species, is generally modelled through ADE (Advection Dispersion Equation). Effects of molecular diffusion and mechanical dispersion are taken into account. Flow field is stationary, while concentration varies along time.

These physical processes are of great interest in the energy field, in fact they include important themes of research and applications such as carbon capture and storage or enhancement oil recovery.

Phenomena are illustrated by means of a set of *partial differential equations*, whose outputs of interest are the velocity vector $\mathbf{u}$ and concentration $c$, within the given domain. However, in order to calculate them, it is not possible to rely on analytical integration methods, since they are applicable to few limited and simple cases, such as uniform flow or constant dispersion tensor. This leads to the necessity of having *numerical methods* at one's disposal, which allow computing an approximation $\boldsymbol{u_h}$ and $c_h$ of the exact solution $\mathbf{u}$ and $c$.

In this thesis we have considered three computational codes:

- *Freefem Fix*;
- *Freefem Adaptive*;
- *Fortran Traces.*

The first name coincides with the language code, whereas the second is the actual nomenclature, they are referred with. Each of them shows different features, although they are all implemented in a Finite Element Framework (FEM).

According to [3], *Fix* is based on a fixed grid Galerkin discretization scheme in space and time, while it employs a Streamline Upwind technique, to limit the spurious oscillation, deriving from the convective term in the transport equation.

According to [3] and [4], *Adaptive* relies on an automatic space and time adaptation technique, which is grounded on an anisotropic, a posteriori error estimator. It is an advisable tool, because of the ability to follow the evolution of the concentration front and optimize, as consequence, mesh and time step. A crucial issue is the projection, so far performed by means of linear interpolation, of velocity field from one grid to another. This concept is linked to the matter of local fluid mass conservation and it is particularly critical especially when topology of the grid changes a lot, with respect to the original mesh, on which Darcy's law has been solved (see [5] for further details).

*Traces* is the most sophisticated and stable code, since it uses a Discontinuos Finite Galerkin Method to solve the advective term in the transport equation. Mass balance is discretized by means of a Finite Volume formulation, whereupon exact conservation of mass could be evaluated at element level (we refer to [6] for further details).
The choice of the codes and their peculiarities enables us to carry out a deep and complete analysis of flow and transport problem. Relying on the theoretical background, taken from the scientific literature, aim of this thesis work is the code benchmarking to understand impact of solution strategies and possible critical features, especially associated with space-time adaptation, applied to numerical solution of solute transport in homogeneous and heterogeneous domain.
In particular three macro-sections have been fulfilled, whose objectives are:

1. Detecting numerical effects of grid size variation and issues deriving from the codes strategy, in the discretization of the convective term at different orders of the grid Péclet number ($Pe_h$).

2. Codes benchmarking in the case of strongly heterogeneous field, in order to explore the effect of high level of complexity in the velocity structure and concentration distribution.

3. Implementing a mass conservative scheme for the interpolation of the flow field between two different numerical grids, to be employed in an adaptive solution strategy.

First assessment is realized on a homogeneous bounded domain, with uniform 1D flow. For the transport problem investigated, a 2D analytical solution is available. The aim consists in verifying the codes stability by varying mesh size and dispersivities values, since, in case of transport dominated by convection ($Pe_h > 2$), the discretization of the velocity term induces false diffusion.
The comparison with the analytical solution detects codes verification at different mesh size. On the other hand, dealing with high Péclet, *Fix* demonstrates a low level of accuracy, whereas *Adaptive* and *Traces* provide a better estimation, in fact the former can adapt the mesh size throughout the domain and automatically control the grid Péclet number, while the latter employs an advanced method, Discontinuous Galerkin, which significantly prevents that false diffusion could occur.

Second assessment is based on a strongly heterogeneous control volume, which is characterized by a fixed variance of a random Gaussian permeability field. As already introduced, the choice of a strong heterogeneity implies a complex structure of the velocity field and so of the concentration map. To ensure accuracy of the results, dispersivity values have been set, in order to avoid artificial diffusion.
*Fix* and *Traces* generates identical solutions of local evolution and spatial distribution of concentration. *Adaptive* instead shows remarkable oscillations, which might be determined by an erroneous interpolation of the advective field. Projection of velocity from one grid to another is a feature which only belongs to the adaptive strategy, in fact both *Fix* and *Traces* are built on a fixed space-time discretization scheme. From the analysis performed and the logarithmic-maps of flux balance, it is evident the actual method, labelled as *NCI* (non-conservative interpolation), which

is implemented in a subroutine of *Freefem* and is intrinsically used in the adaptive procedure, suffers from limitations and shortcomings, such as:

- It does not account for the heterogeneity of the system, because the flux is computed through a single permeability value, eventhough the associated edge intersects different permeability regions;
- It does not respect local flux balance;
- It depends on mesh topology.

We have carried out a double vision of code benchmarking, involving both the discretization scheme (Streamline Upwind versus Discontinuos Galerkin) and the space-time procedure (fix versus adaptive strategy). Several scenarios of flow (homogeneous and heterogeneous domain) and transport (advective or diffusive dominated) problem have been investigated. From this analysis we have identified the lack of local mass balance in the actual interpolation scheme as a key weakness of the adaptive method. Therefore the novelty introduced in this work regards the formulation of a mass conservative scheme, labelled as *CI* (conservative interpolation), for the interpolation of the flow field, between two different numerical grids.
We label as transport grid the adaptive mesh, generated along time and as flow grid, the initial mesh, on which Darcy's law is solved. According to the old procedure *NCI*, the flux, associated to an edge of a given triangle in the transport grid, is rebuilt as average interpolation only considering the middle point of the edge itself. On the contrary, new approach *CI* could be summarized in the following steps:

1. Generic triangle of transport grid is located in the flow grid, so its vertices are pinpointed;
2. All intersections between the edge of the given triangle and elements in flow grid are calculated. Subsequently, the edge is split in several segments, each of them characterized by a middle point and a lenght;
3. A partial flux is computed on the segment. The summation of all contributions allow to determine the flux associated to the edge of the element, which belongs to the transport grid.

In this way the flow field is rebuilt throughout the transport mesh. This method is conservative and consistent with the heterogeneity of the system, in fact, calculating all intersection points and splitting the edge in different segments, it is possible to match the whole edge with the corresponding permeability regions. We remind *NCI* uses a single point (so one permeability value) for the computation of the flux. *CI* is general, *i.e.* it does not depend on the topology of the adaptive grid and it is applicable to whatever remeshing is applied. *CI* is a powerful tool and the difference with the previous approach *NCI* is evident, especially in those elements with a coarser size, because they are characterized by different regions of permeability and an average interpolation with a single point can not catch all the pointwise variations in the permeability field.

Future works should be concerned in optimizing the code, in terms of computational cost, in fact, despite the numerical advantage, its application in couple with the adaptive strategy is not feasible. The final objective should be to verify and compare the solutions, which ensue from running a transport simulation.

# Contents

# Abstract

Objective of this thesis work consists in assessing the impact of a space-time adaptation technique, applied to numerical solution of solute transport in both homogeneous and strongly heterogeneous system, where the heterogeneity is expressed in terms of variance of a random Gaussian permeability field.
In order to detect possible critical features and verify the results, associated with the adaptation procedure, we perform a benchmarking of the numerical code, upon comparing the results, given by the adaptive strategy with those yielded by two different codes, which are built on a fixed space-time strategy.
Adaptive method allows to limit spurious oscillations, deriving from the discretization of the convective term in transport equation, especially when the process is advective dominated, therefore at high grid Péclet number. On the contrary, we note that a crucial issue regards the projection of velocity field between two different meshes. Flow problem is initially solved on a regular grid and then interpolated step by step, from one mesh to another, during the transport simulation. This concept is linked to the matter of fluid mass conservation. The actual interpolation scheme, labelled as *NCI* (non-conservative interpolation), does not respect continuity principle, especially when topology of the grid changes a lot, with respect to the original mesh, on which Darcy's law has been solved.
Novelty of this thesis work coincides with the implementation of a new mass conservative method, labelled as *CI* (conservative interpolation). The latter is conservative and consistent with the heterogeneity of the system. It does not depend on the topology of the adaptive grid and it is applicable to whatever remeshing is yielded. While the approach gives very promising results, a development is required to reduce the computational cost. In this way *CI* scheme could be coupled with a full adaptive transport simulation.

**Keywords:** Heterogeneity, Darcy's law, Solute transport, Adaptivity, Interpolation, Mass conservation.

# Sommario

L'obbiettivo di questo lavoro di tesi consiste nel valutare l'impatto che la strategia di adattività spazio-temporale assume nella risoluzione numerica del trasporto di un soluto in sistemi omogenei ed eterogenei. In particolare l'eterogeneità del dominio viene quantificata tramite un campo Gaussiano casuale di permeabilità a varianza costante. Con il fine di rilevare possibili caratteristiche critiche e verificare i risultati ottenuti, proponiamo una strategia comparativa, selezionando due tipi di codici, basati su metodi numerici a variazione temporale costante e griglia fissa. Si riscontra che il procedimento adattivo permette di limitare lo sviluppo di oscillazioni, specialmente in casi, in cui il trasporto sia avvettivamente dominato (alto numero di Péclet) e di conseguenza la discretizzazione del termine convettivo risulti essere un passaggio delicato. Al contrario, si denota come l'aspetto di maggior criticità riguardi la proiezione del campo di velocità tra due griglie. La mappa di flusso viene inizialmente risolta, tramite approssimazione della legge di Darcy su una griglia regolare, dopodichè ad ogni istante temporale la stessa deve essere ricalcolata tramite interpolazione lineare su una nuova griglia. Tale passaggio è legato alla questione della conservazione della massa fluida, di fatti si evince che, l'attuale metodo di interpolazione lineare, chiamato *NCI*, non fornisce una redistribuzione di massa fluida, che permetta di soddisfare un bilancio locale dei flussi sul singolo elemento. Da questa considerazione si sviluppa la novità introdotta in questo lavoro di tesi, ovvero la formulazione di un nuovo metodo di interpolazione del campo avvettivo, chiamato *CI*, che possa soddisfare il principio di continuità. I risultati finali ottenuti giustificano il netto miglioramento sotto questo aspetto e denotano una maggior consistenza tra il campo di velocità generato su griglia adattata e la mappa di permeabilità del dominio di controllo. Bisogna tuttavia ammettere, che il codice implementato, nonostante dia ottime soluzioni numeriche, non è ottimizzato da un punto di vista computazionale di conseguenza sviluppi futuri dovranno dirigersi su questa strada.

**Parole chiave:** Eterogeneità, Legge di Darcy, Trasporto di soluto, Adattività, Interpolazione, Conservazione della massa.

# Introduction

Solute transport through porous media is of practical importance and plays a significant role in several phenomena, including saltwater intrusion of coastal aquifers, transport of contaminants in soil, secondary and tertiary oil recovery techniques and many others. In the context of petroleum engineering, the study of subsurface transport in porous media became particularly relevant and it is a key component in the design and optimization of oil and gas production operations or in the planning and management of geological carbon sequestration. Solute transport modelling received particular attention for its relevance to *Carbon Capture and Storage* (CCS) and *Enhanced Oil Recovery* (EOR) processes, which are largely affected by solute transport dynamics.

Following [7], CCS has been validated as part of a portfolio of measures to mitigate climate change. To stabilise atmospheric concentrations of carbon dioxide ($CO_2$), at reasonable levels, drastic cuts in anthropogenic emissions are required in the coming decades. A possible strategy deals with capturing $CO_2$ from flue gases and disposing it underground, in depleted hydrocarbon fields or saline. However, in order for the technology to be deployed at large scale, its viability in terms of injectivity, the containment of the injected $CO_2$ and the long term safety with respect to humans and the environment, needs to be guaranteed.

Instead EOR is also known as tertiary phase of the oil production. It is generally employed in fields, which exhibit heavy oil properties, poor permeability or irregular faultlines. The process entails changing the actual features of the hydrocarbons with the aim to restore formation pressure and enhances oil displacement in the reservoir. Gas injection is a possible solution. It mainly involves natural gas, carbon dioxide or nitrogen, which could improve the oil flow, decreasing its viscosity.

These processes require a deep knowledge of the transport phenomena in porous materials. However, due to the impossibility to determine an analytical solution of the governing equations and predict solute behaviour, we should rely on numerical models to approximate the correct results. A critical challenge to the characterization of solute transport in heterogeneous domain regards the development of numerical methodologies, which could render suitable approximations of the space-time dynamics of concentration fields.

The following thesis work is involved in this context, in particular the study is focused on transport of non-reactive chemical in both homogeneous and heterogeneous porous domains at the continuum scale, as describing through the classical Advection Dispersion Equation (see [2], [1], [8]). The advective term, appearing in the ADE accommodates the details of the velocity field, emerging from the solution of the flow problem, based on Darcy's law and continuity constraint. Structure of the convective

field is linked to the heterogeneity of the system, which has been quantified in terms of random Gaussian permeability field with fixed variance. A crucial aspect is concerned with the ability of the model to retain the relevant features of the input heterogeneous map, because preferential pathways and low velocity region can imprint remarkable effects on the transport solution. A common way to design a mesh is based on fixed space-time discretization scheme, however an adaptive procedure could be a valuable alternative, because of the automatic remeshing upon the concentration evolution (see [9], [3], [10]). A series of previous works provide examples of implementation of adaptive grid in the context of numerical modelling of flow (see [11], [5], [12]), moreover a recent approach incorporates the *anisotropic* features with an *a posteriori* error estimator, to guide space remeshing and computational time step (see [13], [14], [4], [15], [10]). The latter has been used in our study.
Relying on the theoretical background, taken from the scientific literature, aim of this thesis work is to assess the impact of space-time adaptation technique, applied to numerical solution of solute transport in homogeneous and heterogeneous domain. In order to detect possible critical features and verify results, we propose a benchmarking approach, whereupon we introduce two further codes. They are implemented through a fixed space-time discretization strategy. The analysis aims at assessing two aspects:

- Stability of the code by varying mesh size and under the imposition of a transport problem, dominated by advection, therefore high grid Péclet number;
- Behaviour of the code in presence of a strong heterogeneous domain.

The first assessment has been run on a homogeneous system. This let us to assume a uniform flow field and focus only on the discretization of the convective term, which is a delicate issue, especially at high grid Péclet number.
The second matter reveals us the inconsistency of the adaptive solution with the conservation of fluid mass. Advective field needs to be projected from one grid to another at each time step. However the actual interpolation method, labelled as *NCI* (non-conservative interpolation) does not abide by the continuity principle. Some works (see [16] and [17]) deals with the numerical errors induced by a non-free divergence velocity field and the analysis, we perform, confirms these considerations. This limit leads to the key novel development, introduced in this thesis work, namely the formulation of a conservative scheme for the interpolation of the flow field between two numerical meshes, which we label *CI* (conservative interpolation). The algorithm embeds a rigorous process of grid projection, detecting all intersection points between an element, which belongs to the adaptive grid, and the original flow mesh, on which Darcy's problem has been solved. Afterwards a conservative computation of the flux is performed. This scheme represents an original feature of this study.

A short description of the contents of each chapter is provided hereafter.

Chapter 1 introduces the governing equations of transport phenomena in porous media, then codes and associated numerical methods, for the approximation of velocity and concentration fields, are illustrated.

Chapter 2 deals with the homogeneous test case. Initially we assess the effects, involving grid size variation, in order to provide verification of each code, thereafter the analysis is shifted towards the simulation of transport process, dominated by convection.

Chapter 3 is concerned with results of solute transport, taking place in heterogeneous domain. Both flow and concentration field are presented and discussed, with a main focus on the adaptive strategy.

Chapter 4 explains the formulation of the new conservative interpolation algorithm, firstly from a theoretical point of view and then numerical results are depicted.

# Chapter 1

# Problem Definition

This thesis work is focused on the study of flow and transport of a non-reactive solute in heterogeneous porous media at continuum scale, as described through the classical Darcy's law and Advection-Dispersion equation (ADE). These physical processes have been illustrated by means of a set of *partial differential equations* (PDE). In this context, the outputs of interest are the velocity and concentration field, however, in order to calculate them, we can not rely on analytical integration methods, because they are limited to very simple cases. Therefore, following [18], from a theoretical point of view, the analysis of a given PDE is often bound to investigating *existence, uniqueness* and possibly, *regularity* of its solutions, but it lacks practical tools for their actual determination. It leads that it is extremely important to have *numerical methods* at one's disposal, which allow to construct an approximation $z_h$ of an exact solution $z$, and evaluate the error $\parallel z_h - z \parallel$ between exact and numerical solution.
According to [18], main features of a whatever numerical method are *convergence, consistency* and *stability.* We denote with $\mathcal{V}(z(g))$ the space, to which the solution $z$ and the set of data $g$, on which the PDE depends, belong. We denote with $\mathcal{V}_h(z_h(g_h))$ the approximated problem.
A numerical method is *convergent*, if the approximated solution tends to the exact one, as the discretization parameter $h$ tends to zero.

$$\parallel z - z_h \parallel \rightarrow 0 \quad \text{as } h \rightarrow 0 \tag{1.1}$$

A numerical method is said *consistent*, if the exact solution satisfies the algebraic equation, obtained for the approximation of the solution.

$$\mathcal{V}_h(z(g)) \rightarrow 0 \quad \text{as } h \rightarrow 0 \tag{1.2}$$

A numerical method is *stable*, if to small perturbations to the data correspond small perturbations to the solution.

The Chapter is organized as follows:

Section 1.1 is dedicated to the presentation of the governing equation for fluid flow and solute transport in porous media.

Section 1.2 is concerned with the introduction of Finite Element Method and discretization techniques upon flow and transport problem.

Section 1.3 describes the general procedure for mesh adaptation and the associated algorithm, implemented in [19].

Section 1.4 introduces the concept of variable conservation.

Section 1.5 shows the objectives of the thesis work and the novelties introduced.

## 1.1 Transport phenomena in porous media

A porous medium is a portion of space, occupied by heterogeneous or multiphase matter. It contains pores or voids, which are enclosed by a solid matrix and generally filled by gaseous or liquid phase (at least for the sake of interest of reservoir engineering). Void spaces are interconnected each other, through continuous paths. As far as flow through porous media is concerned, unconnected pores may be considered as part of the solid matrix. This concept anticipates the definition of *porosity* $\varphi$, the fraction of volume of porous medium, occupied by void space. More strictly speaking, the *effective porosity* is the variable to take into account, since it excludes regions of void space, entirely enclosed by solid material, where fluids can not flow. Furthermore, the fraction of void space of the soil occupied by water is defined as *saturation* $S_w$. Another important property of the geometry of the porous medium is the *permeability* $k\,[m^2]$, which measures the ability to transmit a fluid through it. Following [1], permeable formations, such as sandstone, tend to have many large well-connected pores, whereupon they could transmit fluid readily. On the contrary, shales or siltstones tend to be finer grained with less interconnected pores. The difference in grain shape is depicted in Figure 1.1, the bottom-left sample (Mount Gambier) shows high-porosity and high-permeability with well connected pore space, while the bottom-right (Portland) is well-cemented with shell fragments. In the present thesis work, let us consider a 2D domain in $x - y$ directions, since the depth can be generally neglected, in fact transport phenomena of interest take place along the planar direction, also depending on the geometry of the aquifer/reservoir. The continuum approach is adopted, therefore the porous medium is interpreted as a fictitious continuum, indeed at any point, kinematic, dynamics variables and parameters are continuous functions of the spatial coordinates.

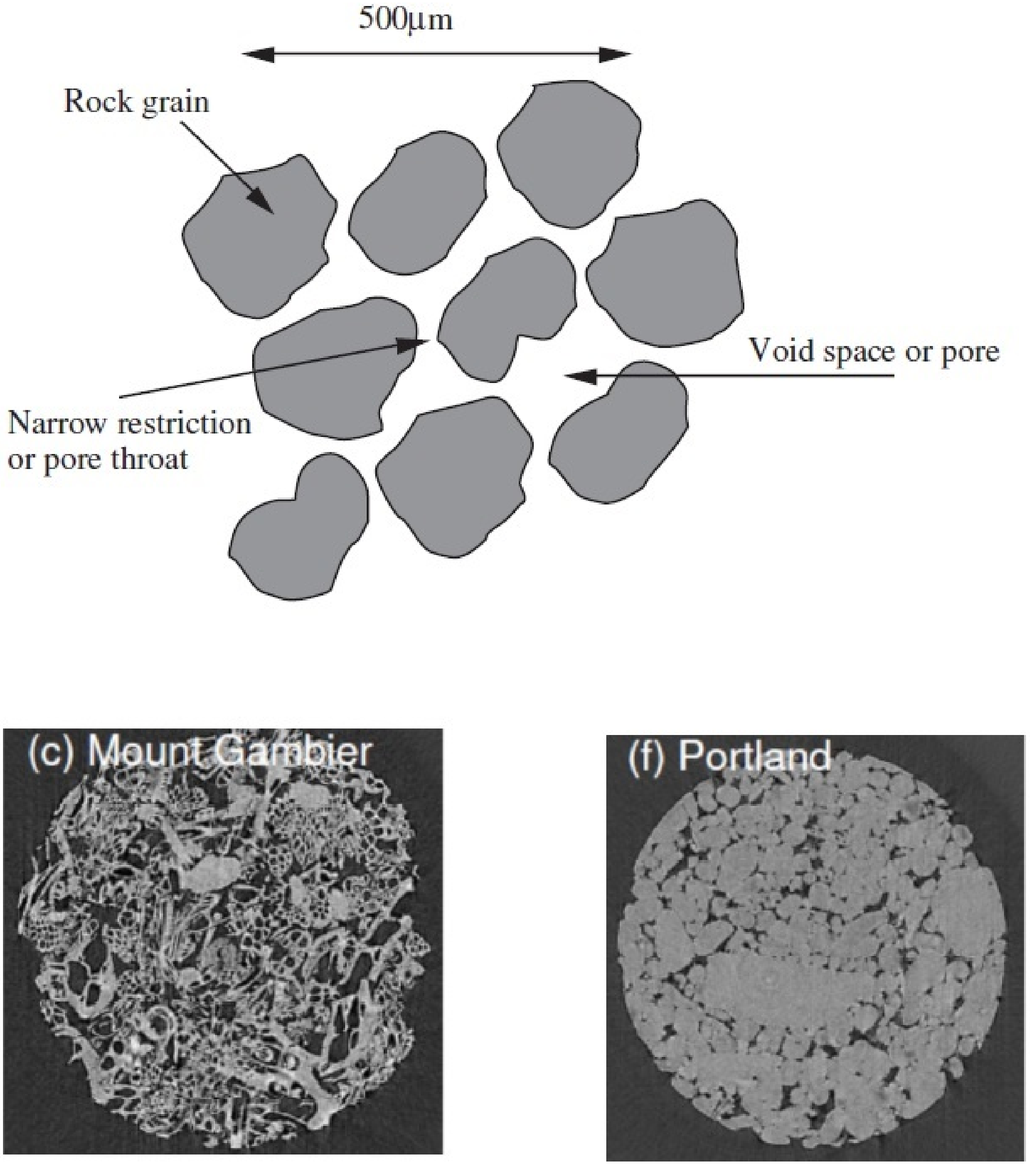


**Figure 1.1:** Top, a schematic two dimensional cross-section through a porous rock; bottom (left and right) a two dimensional cross-section of a three-dimensional image of a limestone, showing individual grains, from [1]

### 1.1.1 Fluid Flow

Let us consider a chemically inactive, viscous, Newtonian, of a constant temperature fluid. Derivation of the equations governing the behaviour of the fluid is here explained.
Two different view points could be taken into account:

- *Eulerian*: a volume element is fixed in a space frame of reference;
- *Lagrangian*: the surface of the volume element is co-moving with the fluid, in a fluid frame of reference.

*Eulerian* approach is considered. The infinitesimal volume element $d\Omega$ is shown in Figure 1.2.

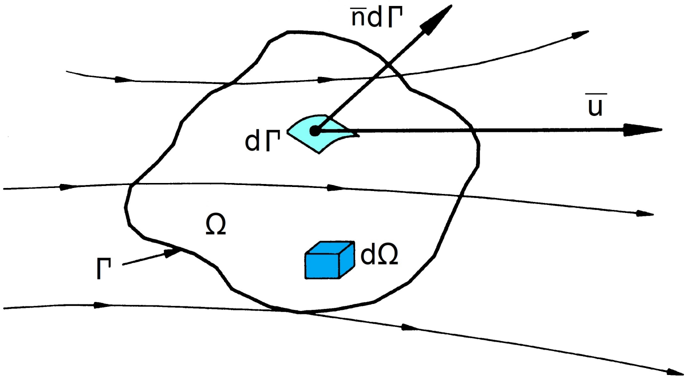


**Figure 1.2:** Finite control volume fixed in space, from [20]

There are two fundamental concepts to use for describing the fluid flow: conservation of mass and Navier-Stokes equation. Firstly, following [20], a model of moving fluid is considered with velocity $\mathbf{u}= (u_1, u_2)^T$ $(1, 2$ denotes Cartesian coordinates $x, y)$. The mass of the infinitesimal element $d\Omega$ is $dM$, which could be calculated as:

$$dM = \rho d\Omega \tag{1.3}$$

We assume there are no sources or sinks of mass within $d\Omega$, then the rate of change is simply related to the mass flow across the boundaries (right-hand side of the Equation (1.4)). Examining Figure 1.2, by convention $\mathbf{n}d\Gamma$ points outwards, hence denoting positive an outflow. Since the control mass $M$ is conserved, the total time rate of change of the fluid mass is zero. From an analytical point of view, recalling Reynolds transport theorem:

$$\frac{dM}{dt} = \int_\Omega \frac{\partial \rho}{\partial t} d\Omega + \int_\Gamma \rho\mathbf{u} \cdot \mathbf{n}\, d\Gamma = 0 \tag{1.4}$$

According to Gauss-Divergence theorem:

$$\int_\Gamma \rho\mathbf{u} \cdot \mathbf{n}\, d\Gamma = \int_\Omega \nabla \cdot (\rho\mathbf{u}) d\Omega \tag{1.5}$$

Coupling Equation (1.4) and (1.5), local conservation in an Eulerian frame reference is derived:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1.6}$$

According to [1], we assume relatively incompressible fluids in a small domain (the pore scale), where changes in pressure, hence density, are small compared to the overall pressure, the density can be approximated as constant and so Equation (1.6) can be rewritten as $\nabla \cdot \mathbf{u} = 0$.

The second concept to introduce is the conservation of momentum, analytically described through Navier-Stokes, in Equation (1.7):

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla P + \mu \nabla^2 \mathbf{u} \tag{1.7}$$

where $P$ indicates pressure $[Pa]$ and $\mu$ the viscosity $[Pa\, s]$.

According to [1] and [21], the flow field changes slowly over time, and it is reasonable to neglect any explicit time dependence. This notion could be quantified by Reynold's number $Re$, ratio of inertial to viscous forces:

$$Re = \frac{\rho u_C L_C}{\mu} \tag{1.8}$$

where $u_C$ and $L_C$ coincide with characteristic velocity and length. Roughly speaking for a sand field and water fluid, typical values of the quantities, involved in Equation (1.8), are $\rho = 10^3\, kg/m^3$, $\mu = 10^{-3}\, Pa\, s$,
$L_C = [10^{-5} \div 10^{-4}]\, m$, $u_C = 10\, m/day$, as consequence, orders of magnitude of $Re$ belongs to the range $[10^{-1} \div 10^{-3}]$. Assumption of laminar regime is so reasonable (see [1]).

It is possible to average the Navier-Stokes equation and derive a linear relation between volumetric flow rate and pressure gradient, known as Darcy's law. According to [2], it reads:

$$\mathbf{Q} = -\frac{k}{\mu} A (\nabla P - \rho \mathbf{g}) \tag{1.9}$$

where $\boldsymbol{Q}$ is the volumetric flow rate $[m^3/s]$, $k$ is the permeability $[m^2]$, $A$ is the cross sectional area $[m^2]$, $P$ is the pressure $[Pa]$, $\rho$ is the density $[kg/m^3]$, $g$ is gravitational acceleration $[m/s^2]$. The porous medium is taken to be saturated with the fluid of interest, in the sense that fluid-fluid interfaces do not form and a single fluid prevails in the pore space.

In case of isotropic and homogeneous media, permeability is reduced to a scalar quantity. In our case, we assume to deal with an heterogeneous system, which is characterized in terms of spatial distribution of permeability (see Section 3.1). A scalar map $k = k(x, y)$ is considered. It is function of space coordinates and it is obtained as random Gaussian field with a fixed variance of natural logarithm, such as in Figure 1.3 (for further details see [9] and [19]).

Avoiding the gravitational effects, the following form is obtained:

$$\frac{\mathbf{Q}}{A} = \mathbf{q} = -\frac{k}{\mu} \frac{\partial P}{\partial \mathbf{x}} = \frac{k}{\mu} \left[ \frac{\partial P}{\partial x} + \frac{\partial P}{\partial y} \right] \tag{1.10}$$

It is important to remind the flow occurs only in the pore space, therefore the effective area has to take into account the porosity $\varphi$. This is particularly significant in order to calculate the actual velocity, because only a fraction of the total formation volume is available.

$$\mathbf{u} = \frac{\mathbf{q}}{\varphi} = -\frac{k}{\mu\varphi}(\nabla P + \rho\mathbf{g}) \tag{1.11}$$

It is also possible to write an alternative formulation of Darcy's law, by introducing two new terms, piezometric head $P_H$ and hydraulic conductivity $K_H$. The former is calculated as the sum of pressure head $P/(\rho\,\mathbf{g})$ and the elevation head $z_H$, which corresponds to the vertical coordinate. The latter is a variable, which accounts for thermo-physical properties of the fluid (by means of $\rho$,$\mu$) and geometry features of the medium (by means of $k$)

$$K_H = \frac{k\rho g}{\mu} \tag{1.12}$$

$$\mathbf{Q} = -A\,K_H \nabla P_H \tag{1.13}$$

Although Equation (1.13) and (1.9) are equivalent in describing Darcy's law, the latter is taken as reference.
It is now possible to derive the final system, which globally describes the fluid flow. Darcy's law and continuity equation are included.

$$\begin{cases} \varphi\mathbf{u} = -k/\mu(\nabla P + \rho\mathbf{g}) \quad \mathbf{x} \in \Omega \\ \nabla \cdot \mathbf{u} = 0 \quad \mathbf{x} \in \Omega \end{cases} \tag{1.14}$$

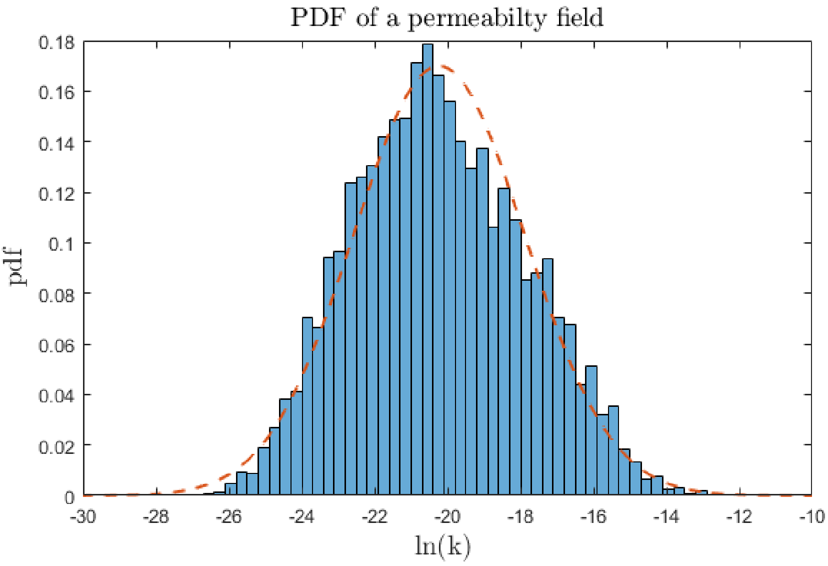


**Figure 1.3:** Probability density function of a random Gaussian field against log-permeability scale, from [19]

### 1.1.2 Transport

We assume a porous medium satutated with water. Transport of species dissolved in a single phase is governed by three physical processes:

**Advection** : it causes translation of solute mass by moving it with the flow velocity;

**Mechanical dispersion** : it derives by the effects of the presence of a velocity distribution at the pore scale on the dispersion of the solute;

**Diffusion** : it describes the spread of particles through random motion in the presence of a concentration gradient.

Following the definitions reported in [22], diffusion is a term reserved for the mixing in the molecular level, where motion is random, while mechanical dispersion occurs due to different velocity channels within the porous medium. This second effect is governed by the variations in the flow field and it usually dominates with huge spreading of a dissolved plume of solute.

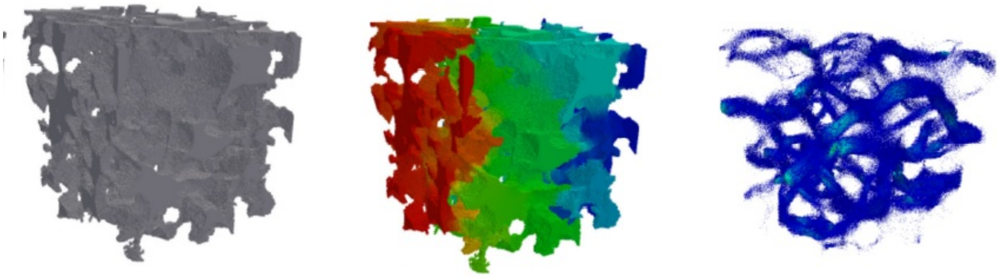


**Figure 1.4:** From left to right, pore space, pressure field (red colour represents high values, while blue low ones) and flow field, from [1]

According to [2], diffusion and dispersion could be quantified and summed up in a parameter defined in tensor form **D**= $\{D_{ij}\}$. It is labelled as the effective dispersion coefficient, which accounts for both molecular and random nature of the flow field. Consider an heterogeneous and anisotropic medium, the tensorial notation ensues:

$$D_{ij} = (\alpha_T \parallel \mathbf{u} \parallel_2 + D_m)\delta_{ij} + (\alpha_L - \alpha_T)\frac{u_i\, u_j}{\parallel \mathbf{u} \parallel_2} \quad \text{with } i,j = 1,2 \tag{1.15}$$

Here $\alpha_L$ $[m]$ and $\alpha_T$ $[m]$ are longitudinal and transverse dispersivity. $\delta_{ij}$ is Kronecker's delta, $D_m$ $[m^2/s]$ is molecular diffusion and $\parallel \mathbf{u} \parallel_2$ denotes the standard Euclidean norm of velocity vector. Neglecting reactive terms, the conservation equation of a solute in a porous media could be derived.
We label:

- $c$ as the concentration of solute per unit pore volume, hence $\varphi\, c$ is the mass per unit volume of soil;

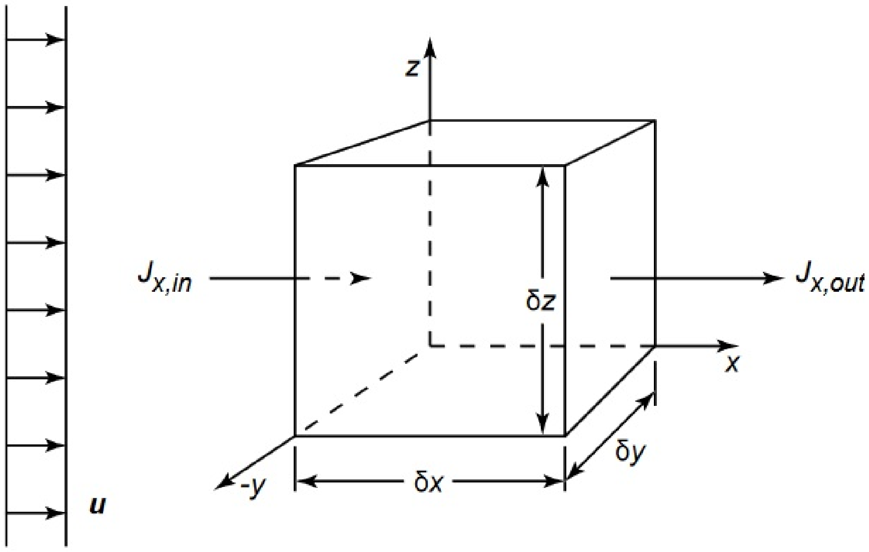


**Figure 1.5:** A schematic control volume, used to derive conservation equation, from [1]

- $\mathbf{J} = \{J_i\}$ is the total flux per unit cross section area, along $i^{th}$ direction.

$$\varphi \mathbf{J} = \varphi \mathbf{u}\, c - \varphi \mathbf{D} \nabla c \tag{1.16}$$

According to [1], convection and dispersion-diffusion are clearly additive and independent each other, in fact advection does not bias the probability a molecule will take a diffusive step.
Referring to Figure 1.5, we consider an arbitrary volume $\Omega$ of a porous medium, bounded by a surface $\Gamma$, then mass balance holds if:

$$\varphi \int_\Omega \frac{\partial c}{\partial t}\, d\Omega + \varphi \int_\Gamma \mathbf{J} \cdot \mathbf{n}\, d\Gamma = 0 \tag{1.17}$$

Green theorem is adopted to convert the surface integral into a volume one:

$$\varphi \int_\Omega \frac{\partial c}{\partial t}\, d\Omega + \varphi \int_\Omega \nabla \cdot \mathbf{J}\, d\Omega = 0 \tag{1.18}$$

In order to perform an overall volume balance, we assume an incompressible fluid, Advection-Dispersion equation (ADE) reads:

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c - \nabla \cdot (\mathbf{D} \nabla c) = 0 \tag{1.19}$$

Manipulating the terms in Equation (1.19), it is also possible to write an equivalent version, known as *conservative form*:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u} c - \mathbf{D} \nabla c) = 0 \tag{1.20}$$

At this point, we provided a theoretical description of the transport phenomena in porous media, however, as discussed at the beginning of the chapter, the actual determination of the variables of interest requires the adoption of numerical methods.

Therefore, the next step corresponds to the shift from the actual partial differential problem to its approximated form and, in order to generate an estimation of the solutions for flow and concentration field, three different codes have been taken into account, which are labelled as:

- *Freefem Fix*: it relies on a fixed space and time discretization scheme (see Section 1.2);

- *Freefem Adaptive*: it is based on an adaptive procedure for the transport problem resolution (see Section 1.3 and Appendix A.2);

- *Fortran Traces*: it is a powerful tool, which uses an advanced level of discretization to guarantee stability of the solution (for further details we refer to Appendix A.1).

The first name corresponds to the language programme, while the second is the actual nomenclature, with they will be referred to.
In the following sections, we argue on the theoretical background, underlying the numerical methods, which have been implemented in the given codes. Each of them is built on a Finite Element Framework, therefore we initially introduce a general overview on the topic, afterwards the focus is shifted on the discretization scheme of flow and transport equation.

## 1.2 Finite Element Method

Finite element method (FEM) has become a very widespread tool, thanks to its flexibility, which allows to accommodate complex domain and heterogeneity in the problem parameters. All the given codes, *Fix*, *Adaptive* and *Traces* rely on this framework.
The common definition of Finite Element, according to [23], reads as follows:

**Definition 1.** A finite element is defined by a triplet $(\Omega,\mathcal{V},\ Z)$, where:

- the domain $\Omega$ is a bounded, closed subset of $\mathbb{R}^n$ with non empty interior and piecewise smooth boundary;

- the space $\mathcal{V} = \mathcal{V}(\Omega)$ is a finite dimensional function space on $\Omega$ of dimension $n$;

- the set of degrees of freedom $Z= \{z_1, z_2, ..., z_n\}$ (*i.e* the values that must be assigned to define uniquely the functions themselves).

This leads to three different steps, which are the drivers for generating a solution: domain discretization, problem approximation and functional space definition. Independently on the type of code, these phases need to be accomplished, in order to achieve an approximated solution $z_h$ of the real variable $z$.

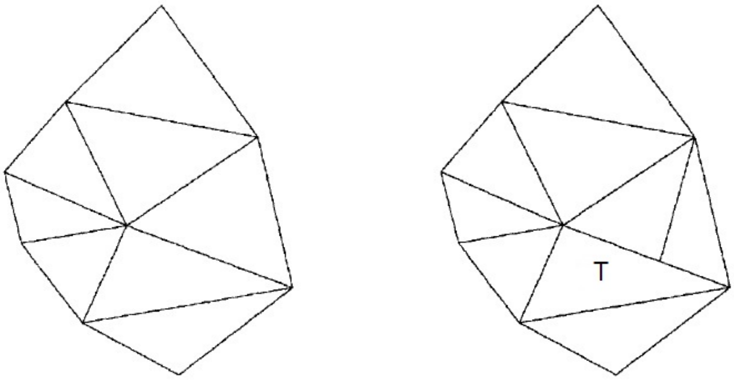


**Figure 1.6:** Conformal (left) and non-conformal (right) mesh, from [24]

### Domain Discretization

We consider a domain $\Omega \in \mathbb{R}^n$ ($n = 2$ or $n = 3$) on which a PDE is defined. In order to solve a partial derivative system in a finite element framework, it is needed to provide an approximation of $\Omega$, labelled as $\Omega_h$. The domain $\Omega_h$ is divided in triangles or rectangles ($n = 2$) or tetrahedra, pentahedra or hexahedra for ($n = 3$). The covering-up of the domain $\Omega_h$, by means of such elements is called *mesh* or *grid*, and it is indicated with the symbol $\mathcal{T}_h$, where $h$ stands for the mesh size. In case of polygonal control volume, $\Omega_h = \cup_{T \in \mathcal{T}_h} T$ and $\Omega_h$ coincides with $\Omega$, therefore, for sake of simplicity, the subscript $h$ will be removed from the notation.
In this thesis work, a polygonal 2D domain has been considered and triangles have been used for discretization, as consequence $h$ represents the triangle diameter. A *conforming* triangulation is needed to be realized, because an efficient mesh reduces the difficulties, related to the computation of solution and more reliable results will be obtained. According to [24], a *conformal* $\mathcal{T}_h$ of $\Omega$ needs to satisfy the following criteria:

- $\Omega = \Omega_h = \text{int } (\cup_{T \in \mathcal{T}_h} T)$;
- the interior of every element $T$ in $\mathcal{T}_h$ is non empty;
- the intersection of the interior of two elements is empty, in order to avoid element overlapping;
- the intersection of two elements in $\mathcal{T}_h$ is either the empty set, a vertex or an edge.

Figure 1.6 depicts an example of conformal and non-conformal mesh, due to the non-matching vertex, located on the edge of the element $T$. Main parameters of the mesh are here enlisted:

- $\Omega$: discrete bounded domain;
- $\Gamma$: boundary of the given domain;
- $h$: mesh size *i.e* triangle diameter;

- $T$: generic triangle in $\mathcal{T}_h$;
- $N_t$: number of triangles element in $\mathcal{T}_h$;
- $V_i$: $i^{th}$ vertex $(i = 1, 2, 3)$ of generic triangle $T$;
- $N_v$: number of vertices in $\mathcal{T}_h$;
- $E_i$: $i^{th}$ edge $(i = 1, 2, 3)$ of a generic triangle $T$;
- $N_e$: number of edges in $\mathcal{T}_h$.

Furthermore, note that meshes are used to discretize a certain domain, to obtain the solution of the problem under investigation. They can be interpreted as spatial support, where each differential equation is solved in a precise location, labelled as *node*. As consequence, it is then possible to classify a triangulation according to the structure, in this sense a *structured* mesh is the simplest kind of grid and it is built, placing nodes in the space. This often leads to an optimization of the implementation of the numerical procedure. The disadvantage of the structured grids is that they can be used only for geometrically simple domains. On the other hand, an *unstructured* mesh allows a more flexible distribution of the nodes in space and it can be applied for complex geometries.

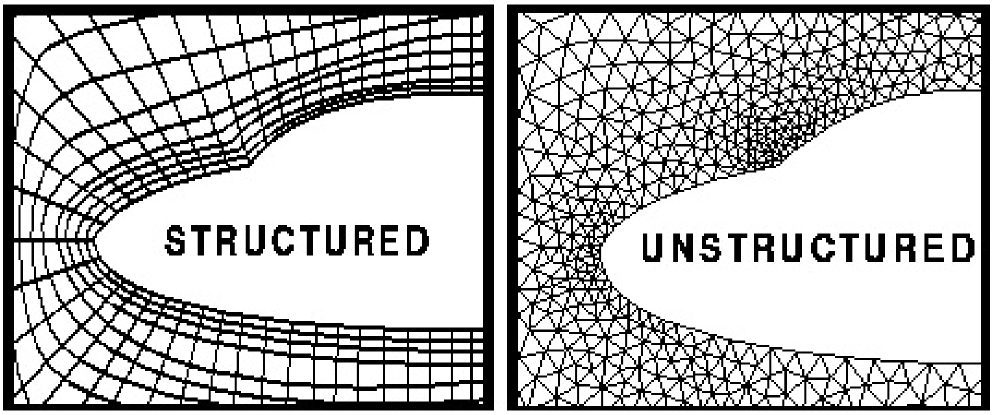


**Figure 1.7:** Structured and unstructured mesh, from [24]

### Problem Discretization

The general idea of Finite Element Method is to numerically approximate a generic solution $z$ of PDE with a function $z_h$, using combinations of basis functions, according to the following expression:

$$z \approx z_h = \sum_{i}^{N} \varpi_i(\mathbf{x}) z_i(t) \tag{1.21}$$

where $\mathbf{x}$ is the vector of coordinates, $t$ the time, $\varpi_i(\mathbf{x})$ is an interpolation (or basis) function defined over the whole domain $\Omega$, $N$ the total number of nodes (or degrees

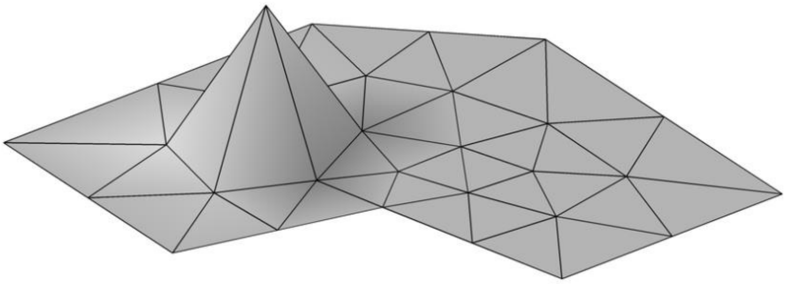


**Figure 1.8:** Basis function in 2D triangular mesh, from [25]

of freedom) and $z_i = z(x_i)$ is the $i^{th}$ degree of freedom, which corresponds to the nodal evaluation of $z$. Assuming Lagrangian basis function (see Section 1.2) $\varpi_i(\mathbf{x})$ takes the value of 1 in the respective node and 0 elsewhere, as illustrated in Figure 1.8.

### Functional Space

Following [18], [23], some basic, but fundamental concepts are recalled, in order to figure out problem interpretation and discretization in Section 1.2.1 and 1.2.2. We denote with $L^2(\Omega)$, the space of functions, which satisfy the following criterium:

$$L^2(\Omega) = \left[ z : \Omega \mapsto \mathbb{R} : \int_\Omega |z(\mathbf{x})|^2 \, d\Omega < +\infty \right] \tag{1.22}$$

where $z(\mathbf{x}) = z(x, y)$ is a generic function, which belongs to $L^2(\Omega)$. It is fundamental, because it identifies the set of square integrable functions.
We denote with $H^1$ the space:

$$H^1(\Omega) = \left[ z \in L^2(\Omega) : \int_\Omega \left( \left( \frac{\partial z}{\partial x} \right)^2 + \left( \frac{\partial z}{\partial y} \right)^2 \right) d\Omega < +\infty \right] \tag{1.23}$$

All functions $z$ embedded in $H^1$ are of class $C^0$ (*i.e* continuous) on a bounded domain $\Omega$, whose derivatives are square-integrable as well. For essential boundary conditions, space $H_0^1(\Omega)$ (subspace of $H^1$) might also be defined, in order to abide by the condition $z|_\Gamma = 0$.
We denote with $H(div)$ the space, consisting of vector fields, which requires the $C^0$ continuity of the normal component.
A functional space is defined, based on the continuity and differentiability requirements of the problem at hand, thereafter, once an appropriate space has been chosen, the goal is to create an approximation, that depends on the mesh size $h$. To this end a partition $\mathcal{T}_h$ has been introduced.
We consider for example the case of functions $z$ embedded in space $H^1(\Omega)$, since they are continuous on the bounded domain $\Omega$, the following family of spaces $\mathcal{V}_h^r$, for the approximated functions $z_h$, could be constructed:

$$\mathcal{V}_h^r = \{ z_h \in C^0(\Omega) : \; z_h|_T \in \mathcal{P}_r \quad \forall T \in \mathcal{T}_h \} \tag{1.24}$$

Having denoted by $\mathcal{P}_r(\Omega)$, the space of polynomials with degree lower than or equal $r$. Space $\mathcal{V}_h^r$ coincides with a subspace of $H^1(\Omega)$, as it is constituted by differentiable functions. The fact that, functions embedded in $\mathcal{V}_h^r$ are locally (element-wise) polynomials makes the computation easier, furthermore, it is convenient to choose a *Lagrangian* basis $\{\varpi_i\}$ for $\mathcal{V}_h^r$, because the coefficient of the expansion of a generic $z_h \in \mathcal{V}_h^r$ , in the basis itself, is the value taken by $z_h$ at the node. To summarize, the *Lagrangian element* is defined for $r = 0, 1$ (for what concerns this thesis work):

$$\Omega = \cup_{T\in\mathcal{T}_h} T \quad T \text{ triangle} \tag{1.25}$$

$$\mathcal{V}_h^r = \mathcal{P}_r(\Omega) \subset H^1(\Omega) \tag{1.26}$$

$$z_i = z(x_i) \tag{1.27}$$

Degrees of freedom in Equation (1.27) coincide with the function evaluated at the vertices ($r = 1$) or baricenter ($r = 0$).

We consider now the space $H(div)$, the main family of subspace is known as *Lowest order Raviart-Thomas* ($RT0$), whose element, for $r = 1$, is defined in the following way:

$$\Omega = \cup_{T\in\mathcal{T}_h} T \quad T \text{ triangle} \tag{1.28}$$

$$\mathcal{V}_h^r = [\mathcal{P}_0(\Omega)]^2 + \mathbf{x}\mathcal{P}_0(\Omega) \quad \mathcal{V}_h^r \subset H(div) \tag{1.29}$$

$$z_i = \int_{Ei} z \cdot \mathbf{n}\, dE_i \tag{1.30}$$

Raviart-Thomas finite element belongs to a space of vector fields (Equation (1.29)), for which each normal component needs to be continuous, and to ensure it, degrees of freedom are moments of the normal component, or rather fluxes across edges $E_i$ of the mesh (Equation (1.30)).

The theoretical overview just presented is essential to understand the discretization scheme, related to flow and transport problem. We remind that FEM represents a common background to all codes.

In the following sections, we argue on the strategies, adopted by *Fix*, *Adaptive* and *Traces* for the discretization of flow and transport, respectively Equation (1.14) and (1.19). For further details, we refer to the scientific literature suggested and to Appendix A.

### 1.2.1 Darcy's equation

Differential System (1.31) represents the *strong* formulation of flow equation. The common way to solve it is to introduce a modified version of the problem, labelled as *weak* form, which reduces the order of derivation, by means of integral formulation, moreover it automatically enforces natural boundary conditions. The procedure, we report below, describes in detail the discretization process, involving in *Fix* and *Adaptive*, nonetheless at the end of the section some considerations, regarding *Traces* are pointed out.

Starting from the general strong formulation:

$$\begin{cases} \mathbf{u} = -k/(\mu\varphi)(\nabla P + \rho \mathbf{g}) \quad \mathbf{x} \in \Omega \\ \nabla \cdot \mathbf{u} = 0 \quad \mathbf{x} \in \Omega \end{cases} \tag{1.31}$$

We introduce the following functional spaces $\mathcal{V} \subset H(div)$, $\mathcal{W} \subset H_0(div)$ and $\mathcal{X} \subset L^2$, (see [18], [3] for further details). We consider two kind of test functions, $\mathbf{w}$ for Darcy law and $\xi$ for continuity equation, thereafter multiplying them for both equations in System (1.31) and integrating on $\Omega$, we find:

$$\int_\Omega \left( \frac{\mu\varphi}{k} \mathbf{u} \cdot \mathbf{w} + \mathbf{w} \cdot \nabla P + \nabla \cdot \mathbf{u}\xi \right) d\Omega = - \int_\Omega (\rho \mathbf{g} \cdot \mathbf{w}) d\Omega \tag{1.32}$$

Applying Green theorem on the pressure term, it is obtained:

$$\begin{aligned} \int_\Omega \mathbf{w} \cdot \nabla P \, d\Omega &= \int_\Omega (\nabla \cdot (\mathbf{w}P) - P\nabla \cdot \mathbf{w}) \; d\Omega \\ &= - \int_\Omega P\nabla \cdot \mathbf{w} \, d\Omega + \int_\Gamma \mathbf{w} \cdot \mathbf{n} P \, d\Gamma \end{aligned} \tag{1.33}$$

Rearranging Equation (1.33) into (1.32), the final weak formulation ensues. It demands to find $\mathbf{u} \in \mathcal{V}$ and $P \in \mathcal{X}$ such that, for any $\mathbf{w} \in \mathcal{W}$ and $\xi \in \mathcal{X}$:

$$\int_\Omega \left( \frac{\mu\varphi}{k} \mathbf{u} \cdot \mathbf{w} - P\nabla \cdot \mathbf{w} + \nabla \cdot \mathbf{u}\xi \right) d\Omega = - \int_\Omega (\rho \mathbf{g} \cdot \mathbf{w}) d\Omega \tag{1.34}$$

As discussed in previous section, the original PDE (1.31) needs to be discretized, generating the so called *Galerkin problem*, however, instead of the strong form, the weak Equation (1.34) is solved, in order to seek approximation $\mathbf{u}_h \in \mathcal{V}_h$ and $P_h \in \mathcal{X}_h$. The finite dimensional spaces $\mathcal{V}_h$, $\mathcal{X}_h$, which have just been introduced, are simply subspaces of the previous ones, therefore $\mathcal{V}_h \subset \mathcal{V}$ and $\mathcal{X}_h \subset \mathcal{X}$. The same occurs for test functions, which have to be taken into discrete form, or rather $\mathbf{w}_h \in \mathcal{W}_h$ and $\xi_h \in \mathcal{X}_h$. Furthermore, according with the notation, defined in Section 1.2, $\mathcal{V}_h$, $\mathcal{X}_h$ coincide respectively with a Raviart-Thomas space ($RT0$) and a zero order Lagrangian space $\mathcal{P}_0$.

It is important to underline that Finite Element Method, in which two spaces are used to approximate two different variables receive the general denomination of *Mixed Methods*. Following [18], [26], this technique has been adopted because it is particularly appropriate for the description of phenomena with two natural independent variables, such as flow problem (1.14) with pressure and velocity.

Thus $RT0 - \mathcal{P}_0$ discretization of (1.34) reads: find $\mathbf{u}_h \in \mathcal{V}_h$ and $P_h \in \mathcal{X}_h$ such that, for any $\mathbf{w}_h \in \mathcal{W}_h$ and $\xi_h \in \mathcal{X}_h$,

$$\sum_{T \in \mathcal{T}_h} \int_T \left( \frac{\mu\varphi}{k} \mathbf{u}_h \cdot \mathbf{w}_h - P_h \nabla \cdot \mathbf{w}_h + \nabla \cdot \mathbf{u}_h \xi_h \right) dT = - \sum_{T \in \mathcal{T}_h} \int_T (\rho \mathbf{g} \cdot \mathbf{w}_h) dT \tag{1.35}$$

Since velocity field is steady, the problem is solved once, for prescribed set of data. We remind the above description regards the discretization scheme for the flow problem, adopted by *Fix* and *Adaptive*. On the other hand, *Traces* is still built on a Mixed Finite Element, therefore the approximate velocity belongs to functional space $RT0$, while the pressure to the zero order Lagrangian $\mathcal{P}_0$. The difference consists in the governing equation, taken into account, in fact *Traces* approximates Equation 1.13, calculating first the piezometric head and then the actual flux. We always refer to Appendix A.1 for further details.

### 1.2.2 Advection-Dispersion Equation

The procedure, we report below, describes in detail the discretization process, involving in *Fix* and *Adaptive*, nonetheless at the end of the section some considerations regarding *Traces* are pointed out.
Conservative formulation (see [1], [2], [18]) of transport reads:

$$\frac{\partial c}{\partial t} - \nabla \cdot (\mathbf{D}\nabla c - \mathbf{u}c) = 0 \tag{1.36}$$

Assume $c \in \mathcal{Z}$, $\mathbf{u} \in \mathcal{V}$ with $\mathcal{Z} \subset H^1(\Omega)$ and $\mathcal{V} \subset H(div)(\Omega)$, in order to find the weak formulation of the above equation, a test function $\zeta \in \mathcal{Z}$ is considered. Following [3], the weak formulation reads:

$$\int_\Omega \left[\frac{\partial c}{\partial t}\zeta - \zeta\nabla \cdot (\mathbf{D}\nabla c - \mathbf{u}c)\right] = 0 \tag{1.37}$$

Applying Green theorem, it is obtained:

$$\begin{aligned} &\int_\Omega \left[\zeta\nabla \cdot (\mathbf{D}\nabla c - \mathbf{u}c)\right] d\Omega \\ = \int_\Gamma \zeta(\mathbf{D}\nabla c - \mathbf{u}c)\cdot \mathbf{n}\, d\Gamma - &\int_\Omega (\mathbf{D}\nabla c - \mathbf{u}c)\cdot \nabla\zeta \, d\Omega \end{aligned} \tag{1.38}$$

Rearranging Equation (1.38) into (1.37), the final weak formulation ensues:

$$\int_\Omega \left[\frac{\partial c}{\partial t}\zeta + (\mathbf{D}\nabla c - \mathbf{u}c)\cdot\nabla\zeta\right] d\Omega = 0 \tag{1.39}$$

Further details, regarding the solution of the weak problem, are given in Appendix B.1.
Thereafter, Galerkin discrete approximation, which is implemented in *Fix* and *Adaptive* is immediately derived:
find $c_h = c_h(t) \in \mathcal{Z}_h \subset \mathcal{Z}$ such that for any $\zeta_h \in \mathcal{Z}_h$:

$$\sum_{T\in\mathcal{T}_h}\left[\int_T \left(\frac{\partial c_h}{\partial t}\zeta_h + (\mathbf{D}_h\nabla c_h - \mathbf{u}_h\, c_h)\cdot\nabla\zeta_h\right) dT\right] = 0 \tag{1.40}$$

*Traces*, instead develops a mass balance, through a finite volume formulation, which allows to ensure exact conservation at local element-wise level. Convective part of Equation (1.40) is solved through Discontinuous Finite Element, the rest by Mixed Hybrid Finite Element (see Appendix A.1).

Discretization of the convective term in Equation (1.40) demands a particular treatment. In the following part, we outline the numerical issue and strategy adopted by each code.

#### Stabilization of the convective term

In multidimensional problems, both convection and diffusion carry information. Velocity field determines direction and speed of the convective transport, whereas the net diffusive flux depends on the definition of $\mathbf{D}$, in particular on longitudinal and transverse dispersivity $\alpha_L$ and $\alpha_T$. In case of transport advective-dominated (see Section 2.3 and Appendix B.2 for further details), the erroneous discretization of the convective term induces numerical crosswind diffusion, which needs to be stabilized. Each of the considered codes manages it in a different way:

- *Fix* uses the so called Streamline Upwind Technique, which is presented below;
- *Adaptive* can count on the remeshing itself. Mesh size refinement (see Section 1.3) allows to obtain accurate results in the discretization of the convective term.
- *Traces* employs another stabilization method, known as Discontinuous Galerkin. It is much more sophisticated than the Streamline Upwind technique and it ensures exact local mass balance at element level, without oscillations and reducing the artificial diffusion (we refer to Appendix A.1 for further details).

We focus on the description of Streamline Upwind technique. It introduces anisotropic balance dissipation, acting along the streamlines, but not transversely. Referring to [27], [18], [28], the upwind artificial-viscosity could be generalized for the 2D case, simply by adding to Equation (1.40) a term like:

$$b_h(c_h, \zeta_h) = \sum_{T \in \mathcal{T}_h} \gamma_T \int_T (\mathbf{u}_h \cdot \nabla c_h)(\mathbf{u}_h \cdot \nabla \zeta_h) dT \tag{1.41}$$

The resulting discrete problem is therefore a modification of the initial Galerkin (1.40). Basically, the integrand of Equation (1.41) represents a weak form of the second convective derivative:

$$(\mathbf{u} \cdot \nabla)^2 c = \mathbf{u} \cdot \nabla(\mathbf{u} \cdot \nabla c) \tag{1.42}$$

therefore the term $b_h(c_h, \zeta_h)$ incorporates a *streamline diffusion* into the Galerkin scheme. The stabilization parameter $\gamma_T$ is defined, according to [3] as

$$\gamma_T = \frac{h_T}{3|\mathbf{u}_h|_T} \tag{1.43}$$

where $h_T$ is the triangle diameter, while $|\mathbf{u}_h|_T$ is the velocity module. Both quantities are referred to element $T$.

Finally time discretization is performed via $\vartheta$-method, setting $\vartheta = 2/3$ to guarantee the unconditionally absolute stability.
According to *Fix* approach, we denote:

$$R(c_h, t) = \left( \frac{\partial c_h}{\partial t} \zeta_h + \nabla \zeta_h \cdot (\mathbf{D}_h \nabla c_h - \mathbf{u}_h c_h) \right) + \gamma_T (\mathbf{u}_h \cdot \nabla c_h)(\mathbf{u}_h \cdot \nabla \zeta_h) \tag{1.44}$$

A compact notation of the discrete problem, including both time and space discretization, could be written in the following way:

$$\frac{1}{\Delta t}(c_h^{n+1} - c_h^n) + (1 - \vartheta) R(c_h^n, t^n) + \vartheta R(c_h^{n+1}, t^{n+1}) = 0 \tag{1.45}$$

## 1.3 Space Time Adaptation

Both *Fix* and *Traces* are characterized by fixed strategy, in terms of space-time discretization. This means that numerical mesh and computational time step need to be defined a priori via a suitable convergence analysis (see Chapter 3). On the other hand, *Adaptive* embeds another scheme, which is here clarified.

In this Section, space-time adaptation procedure is rapidly illustrated, in order to give a general view of the theoretical background. The whole scheme is reported in Appendix A.2 and we refer to the suggested literature for further details (see [3], [9], [11], [29], [30], [5], [13], [14], [31], [32], [10], [15]).

The method is said adaptive, because the process depends on previous results at all stages and it has to be interpreted as an enhancement of the classic finite element algorithm (*Fix*) described in Section 1.2.

An automatically adaptive grid is a very advisable tool in modelling real scale applications, in fact, often, the most relevant processes, driving the evolution of a variable, take place in different regions throughout the domain, therefore a homogeneous distribution could lead to useless computational cost. The actual remeshing and computation time definition are grounded on an *anisotropic a posteriori* error estimator.

About the choice of an *anisotropic* estimator, the main objective is to include geometrical information, regarding orientation and size, about a generic triangle $T$ in mesh $\mathcal{T}_h$. This may be optimized for the discretization of ADE, in particular to concentration evolution within the domain. Furthermore, according to [14], using rather elongated elements may help in reducing the number of degrees of freedom necessary for a given solution accuracy.

About the choice of *a posteriori* estimator, it is important to remind that an analytical solution is not available for the Problem (1.20) at hand, therefore neither the error could be calculated. However, a posteriori estimator only relies on the discrete function and it can qualitatively detect which elements in the mesh give a larger error contribution. This could be perfectly couple with a refinement, just selecting those triangles with a significant error and subdividing them, reducing the mesh size $h$.

The algorithm, which has been implemented in [19] combines the discretization scheme of ADE with the information, provided by the error estimator $\eta_{ht}$

$$\eta_{ht} = \eta_h + \eta_t \tag{1.46}$$

which includes a compound $\eta_h$ for spatial discretization and one, $\eta_t$ for time.

The objective is to automatically adapt the mesh $\mathcal{T}_h$ and time domain $[0, t_{end}]$, moreover, in order guarantee the error below a certain value, following [10], two accuracy targets need to be specified:

$$\tau = \tau_{\Delta t} + \tau_h \tag{1.47}$$

$\tau_h$ drives the mesh adaptation, while $\tau_{\Delta t}$ the time advancement. The main goal is to find the mesh with the least number of elements and associated with the fixed accuracy (*i.e.* such that $\eta_h \simeq \tau_h$), while guaranteeing an equidistribution of the error in space. Once the optimal mesh has been derived, some constraints have to be imposed, in order to avoid a cluster of elements with small surface area. The following input parameters are set:

$p_{min}$ : minimum area value allowed for $T$;

$N_{min}$, $N_{max}$ : maximum and minimum number of elements in $\mathcal{T}_h$;

$h_{min}$, $h_{max}$ : maximum and minimum length of triangle edges.

Same considerations occur to control the time adaptation, therefore we fix:

$\Delta t_{min}$ : minimum allowed time step for the adaptation procedure.

$\Delta t_{max}$ : maximum allowed time step for the adaptation procedure.

### 1.3.1 Algorithm

Referring to Figure 1.9, which depicts the whole algorithm, the strategy to combine the discretization technique (Section 1.2.1 and 1.2.2), with space-time adaptation is here detailed. The variable of interest is the solute concentration $c$. To start, preliminary *input data* need to be assigned, therefore:

1. Velocity field from flow problem $\longrightarrow$ $\mathbf{u} = (u_1, u_2)^T$;

2. Initial concentration $\longrightarrow$ $c_0 = c(\mathbf{x}, t = 0)$;

3. Space adaptivity parameters $\longrightarrow$ $p_{min}$, $\tau_h$, $h_{max}$, $h_{min}$, $N_{max}$, $N_{min}$;

4. Time adaptivity parameters $\longrightarrow$ $\Delta t_{max}$, $\Delta t_{min}$, $\tau_{\Delta t}$.

We consider time step $t^{n+1}$ and we assume to know concentration map, grid distribution and time interval length at $t^n$. First of all, Equation (1.20) is discretized on interval $I_n = [t^n, t^{n+1}]$ and grid $\mathcal{T}_h^n$, via Galerkin approximation (1.40), thus yielding the solution $c_h^*(t^n)$. The latter is employed to generate the adaptive mesh $\mathcal{T}_h^n$, following the anisotropic adaptive procedure (see Appendix A.2), whereupon all the quantities associated with $\mathcal{T}_h^n$ are projected on the new grid $\mathcal{T}_h^{n+1}$.
According to [33], at each adaptation step, a complete remeshing is performed by the *Freefem* mesh generator BAMG, which, as far as possible, preserves the previous position of mesh nodes, during adaptation. Nevertheless, to successfully complete the generation of the adapted mesh and to make the whole simulation process more stable and smoother, some controls on the topology and cardinality are mandatory. This leads to bound on the minimum triangle area and maximum element number, such as discussed in previous section. In this way, the actual $c_h(t^n)$ is defined. Notice that, to contain the computational cost, characterizing the whole time window, we do not resort to an iterative algorithm to get the adapted mesh $\mathcal{T}_h(t^{n+1})$, by demanding a stagnation of the number of mesh elements. On the contrary, the mesh identified by the optimal metric is directly assumed as the mesh to be associated with time $t^{n+1}$. Finally moving from the approximate solutions, the next time step is predicted (see Appendix A.2). Concerning the projection step, a standard $L^2$-projection is adopted, because it exhibits good conservation properties, when applied, for instance, to unsteady solute transport equation (see Section 1.4).

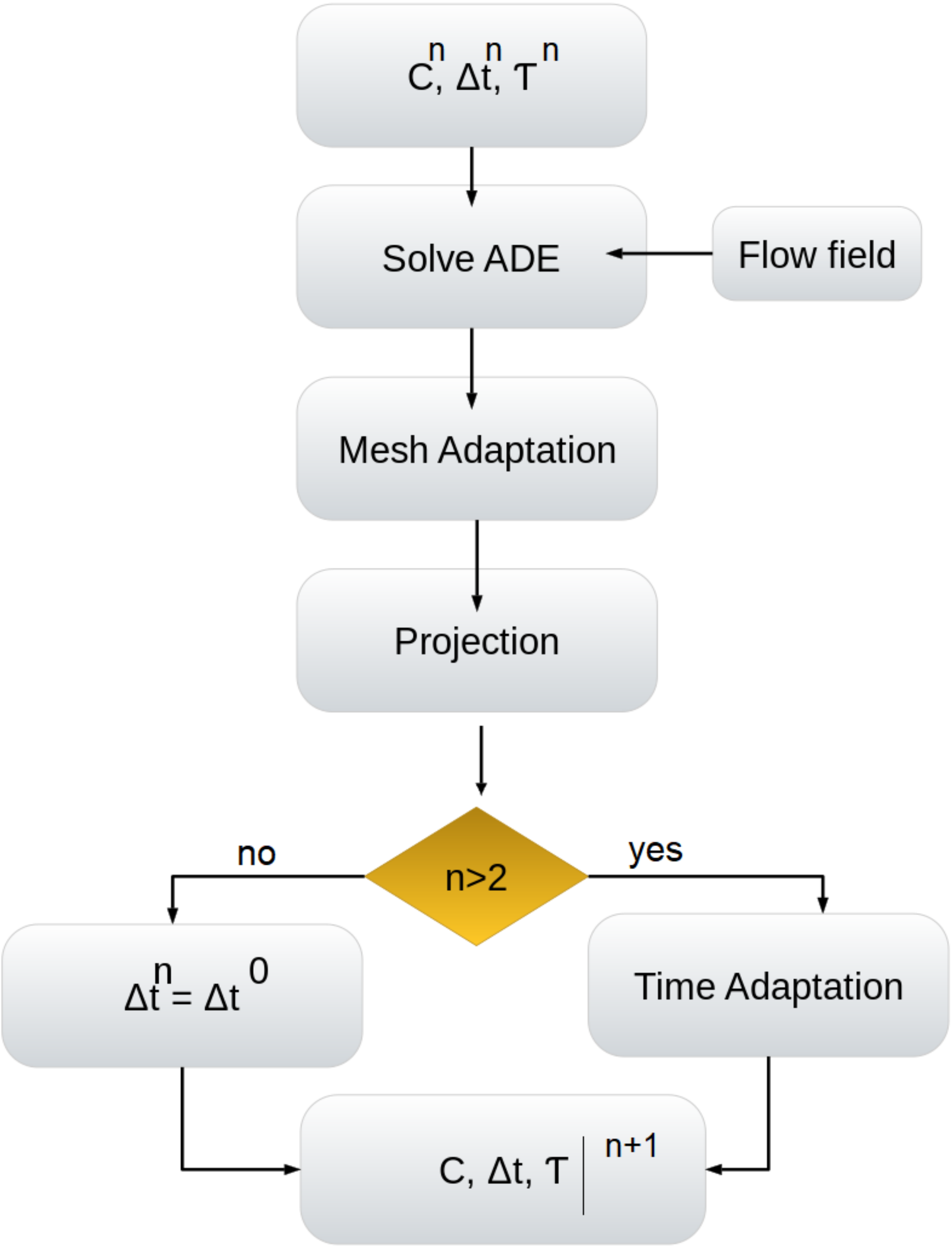


**Figure 1.9:** Sketch of the solution-adaptation algorithm

## 1.4 Variable Conservation

As explained in [30], in a mesh adaptation procedure a crucial issue is the projection of the solution computed on $\mathcal{T}_h^n$ on the new grid $\mathcal{T}_h^{n+1}$. As depicted in Figure 1.10, the highlighted triangle in mesh $\mathcal{T}_h^{n+1}$ intersects six triangles of $\mathcal{T}_h^n$, rendering the translation of the information a difficult task. This step has to be handled carefully, especially in an unsteady framework as a non-conservative projection may heavily compromise the accuracy of the computed solution during the propagation of the process.

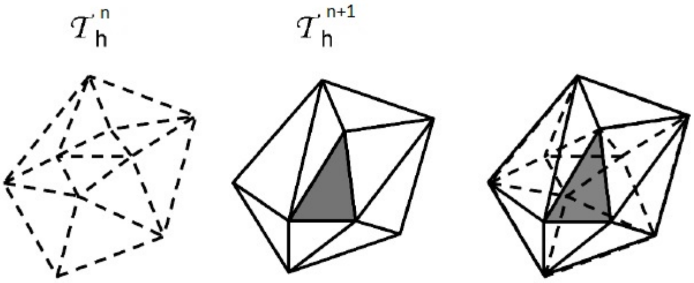


**Figure 1.10:** Graphical explanation of interpolation error among two different mesh, from [30]

Two variables need to be projected from one mesh to another, concentration and velocity field. About the former, at global scale, we need to verify the conservation of the solute mass on different meshes, but same time instant, in fact the new adapted grid is built, based on concentration gradient at previous time step. This means that the mass content, evaluated either on $\mathcal{T}_h^n$ or $\mathcal{T}_h^{n+1}$, must be the same.
The following recipe is resorted:

$$\sum_{T\in\mathcal{T}_h^{n+1}} \int_T (c_h^{n+1} - c_h^n) \cdot \zeta_h^{n+1}\, dT = 0 \tag{1.48}$$

where $c_h^n$ stands for the concentration, evaluated at time $t^n$, whereas $\zeta_h^{n+1}$ denotes the generic finite element test function, associated with the new mesh $\mathcal{T}_h^{n+1}$. Numerical solution of Equation (1.48) has been implemented via a dedicated algorithm. This procedure ensures, in general, a global conservation of the variable with respect to the $L^2$-norm, in particular it allows a reasonable global conservation of mass during the projection step.
On the contrary, for what concerns velocity field, Equation (1.48) offers acceptable results from a global point of view, but we also need to ensure a local conservation, in order to avoid fictitious fluid mass redistribution.
It is reminded that the velocity profile is stationary, therefore it has been computed once, before running transport simulation. However, time to time, passing from one mesh to another, the adaptation procedure implies to recalculate it, and this is usually done by means of linear interpolation, which unfortunately does not satisfy

the continuity constraint, especially when the transport grid changes a lot with respect to the initial flow grid. Difficulties arise from the imposition of a strong heterogeneous field, which induces a complex spatial arrangement of the convective map. It exhibits preferential channels and low velocity regions, which demand a precise scheme of interpolation, in order to be reproduced in the correct way.
Moreover, due to the dependence on the advective field, results of concentration have been found to be very sensitive to the quality of the velocity approximation (see Chapter 3), particularly regarding the satisfaction of the divergence free-condition. Such errors determine in the existence within the flow field of sources and sinks, which numerically provide a generation of fluid mass.
To distinguish the notation and simplify the reading of this section, we denote as *Flow grid*, the regular mesh, where Darcy's problem has been solved, whereas *Transport grid* represents the adapted mesh. According to Figure 1.11, we state to use the following symbols:
Flow grid:

- $\mathcal{T}_h^0$: flow grid;
- $T$: generic triangle in $\mathcal{T}_h^0$;
- $E_i$: $i^{th}$ edge of triangle $T$ in $\mathcal{T}_h^0$;
- $R_i$: intersection point;
- $S_i = (\overline{R_i - R_{i+1}})$: segment, composed by two consecutive intersection points;
- $M_i = (x_M, y_M)$: middle point of $S_i$;
- $q_{i,j}$: flux associated to segment $S_i$, which composes the edge $L_j$;
- $Q_i$: stationary flux, computed through Darcy's law, associated with $E_i$.

Transport adaptive grid:

- $\mathcal{T}_h^n$: transport adaptive grid;
- $K$: generic triangle in $\mathcal{T}_h^n$;
- $L_i$: $i^{th}$ edge of triangle $K$ in $\mathcal{T}_h^n$;
- $N_i = (x_N, y_N)$: middle point of $L_i$;
- $q_j$: unknown flux, associated with $L_j$.

In order to deeply understand the problem, let us consider a generic triangle $T$ in $\mathcal{T}_h^0$ and denote with $Q_i$ the flux associated with $E_i$ (*i.e* the $i^{th}$ edge of triangle $T$). Following [34], to compute the flux, an integral quadrature formula with one Gauss point, the middle point of the edge, has been used, while the orientation of the normal is justified by the sign of vertices number.
In the original flow grid, the following expression holds:

$$\int_T \nabla \cdot \mathbf{u} \, dT = \sum_{i=1}^{3} \int_{Ei} \mathbf{u} \cdot \mathbf{n} \, dE_i \simeq \sum_{i=1}^{3} Q_i \approx 0 \tag{1.49}$$

Numerically speaking, the summation can not be equal to zero, but in the order of magnitude of $[10^{-14} \div 10^{-15}]$, which is acceptable from a conservative point of view. However, once grid adaptation is performed, it is not possible to determine a priori new vertices position, therefore 2D interpolation of the flow field is a compulsory step.

As discussed in Section 1.1.1, heterogeneity is quantified in terms of random Gaussian permeability field with constant variance. In case of strongly heterogeneous domain, advective profile is characterized by either low and high velocity regions, which are associated to the original flow grid. On the other hand, transport mesh, due to the adaptation process, is not conformal anymore in describing the initial Darcy's map, especially when the grid topology is completely altered. In accordance with this context, we can understand that interpolation is the only alternative to project velocity from one grid to another. Thinking to solve at each time step the flow problem would be impossible for the inconsistency between the generated transport grid and the structure of the heterogeneous field.

The actual scheme is implemented by means of linear interpolation in a subroutine of *Freefem*. We label it *NCI* (non-conservative interpolation) and to figure out how it works, we refer to Figure 1.11.

We denote as $\mathcal{T}_h^0$ the original flow grid, where triangle $T$ lay (vertices $B_1, B_2, B_3$). We also consider the highlighted triangle $K$ of vertices $V_1, V_2, V_3$, which has been taken from the transport mesh $\mathcal{T}_h^n$.

We denote with $N_i = (x_N, y_N)$ the middle point of edge $L_i = (\overline{V_j - V_i})$ and with $q_i$ the associated unknown flux.

Following [35] and [34], we consider the bold triangle $T$, which contains middle point $N_3$ of edge $L_3$.

Fluxes $Q_i$ are applied at the middle point of the associated edge $E_i$ and they are known, since they have been calculated as direct numerical solution from the discretization of Darcy's problem. The surface of triangle $T$ could be interpreted as a piecewise interpolation scheme, therefore by inserting the values of the three known medium points of $E_i$, it is possible to build a linear system of equations, where the value $Q_i$ can be visualized as the altitude:

$$\begin{cases} Q_1 = ax_1 + by_1 + c \\ Q_2 = ax_2 + by_2 + c \\ Q_3 = ax_3 + by_3 + c \end{cases} \tag{1.50}$$

Coefficients $a, b, c$ define a generic plane, hence the value $Q(x, y)$ at any arbitrary point $(x, y)$, within the triangle, can be found, such as in the case of $q_3$ (Figure 1.11). Summing up, what *NCI* scheme performs at each time step, to determine the unknown flux $q_i$, is:

1. To locate the medium point $N_i \in \mathcal{T}_h^n$ into the original flow grid $\mathcal{T}_h^0$;

2. To compute $q_i$ by means of 2D linear interpolation, based on the stationary fluxes $Q_i$, computed on the original grid $\mathcal{T}_h^0$;

3. Associate the value $q_i$ with the whole edge $L_i$.

In Chapters 3 and 4, it is demonstrated this strategy induces significant errors in the fluid fluxes balance, because it is an average flow reconstruction, based on a single point. As consequence, it does not allow to generate a free divergence velocity field along time.
Roughly speaking, to have a perception of the matter, advection transport Equation (1.51) is taken into account:

$$\begin{cases} \partial_t c + \nabla \cdot (\mathbf{u}\, c) = 0 \\ \partial_t c + \mathbf{u} \cdot \nabla c = 0 \\ \partial_t c + c\, \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla c = 0 \end{cases} \tag{1.51}$$

From a theoretical point of view, all equations (1.51) are consistent and equivalent, because of null divergence (continuity principle), nevertheless, from a numerical point of view, it holds only in the case $\nabla \cdot \mathbf{u} \approx 0$.
This is not always satisfied during adaptation procedure and, in order to figure out the order of magnitude, we perform a sensitivity analysis, assuming the following data, which have to be interpreted as average values:

$$\begin{aligned} c \approx [10^{-4} \div 10^{0}] \rightarrow \bar{c} \approx 10^{-2} \\ \overline{\nabla \cdot \mathbf{u}} \approx 10^{-15} \rightarrow \bar{c}\,\overline{\nabla \cdot \mathbf{u}} \approx 10^{-17} \quad \text{(Darcy)} \\ \overline{\nabla \cdot \mathbf{u}} \approx 10^{-5} \rightarrow \bar{c}\,\overline{\nabla \cdot \mathbf{u}} \approx 10^{-7} \quad (\mathit{NCI}) \end{aligned} \tag{1.52}$$

The difference is quite impressive, almost 10 orders of magnitude, which could really affect the accuracy of the ensuing concentration profile, since the divergence is not negligible anymore.
This concept establishes the main novelty, developed in this thesis work, in fact in the following chapters, several simulations have been run and analysis performed, in order to rigorously detect any issue in the fluid conservation.

## 1.5 Target of the thesis

The choice of the codes and their peculiarities let to carry out a deep and complete analysis of flow and transport problem, touching different aspects. Relying on the theoretical background, taken from the scientific literature, aim of this thesis work is the code benchmarking to understand impact of solution strategies and possible critical features, especially associated with space-time adaptation, applied to numerical solution of solute transport in homogeneous and heterogeneous domain. In particular three objective are investigated:

1. Detecting numerical issues deriving from the codes strategy, by changing mesh size and due to discretization of the convective term at different orders of the grid Péclet number ($Pe_h$). A homogeneous domain is taken into account.

2. Codes benchmarking in the case of strongly heterogeneous field, in order to explore the effect of high level of complexity in the velocity structure and concentration distribution.

3. Implementing a mass conservative scheme, labelled as *CI* (conservative interpolation), for the interpolation of the flow field, between two different numerical grids, to be employed in an adaptive solution strategy.

The thesis organizations reflects those goals:

Chapter 2 deals with the homogeneous test case. Initially we assess the effects, involving grid size variation, in order to provide verification of each code, thereafter the analysis is shifted towards the simulation of transport process, dominated by convection.

Chapter 3 is concerned with results of solute transport, taking place in heterogeneous domain. Both flow and concentration field are presented and discussed, with a main focus on the adaptive strategy.

Chapter 4 explains the formulation of the new conservative interpolation algorithm *CI*, firstly from a theoretical point of view and then numerical results are depicted.

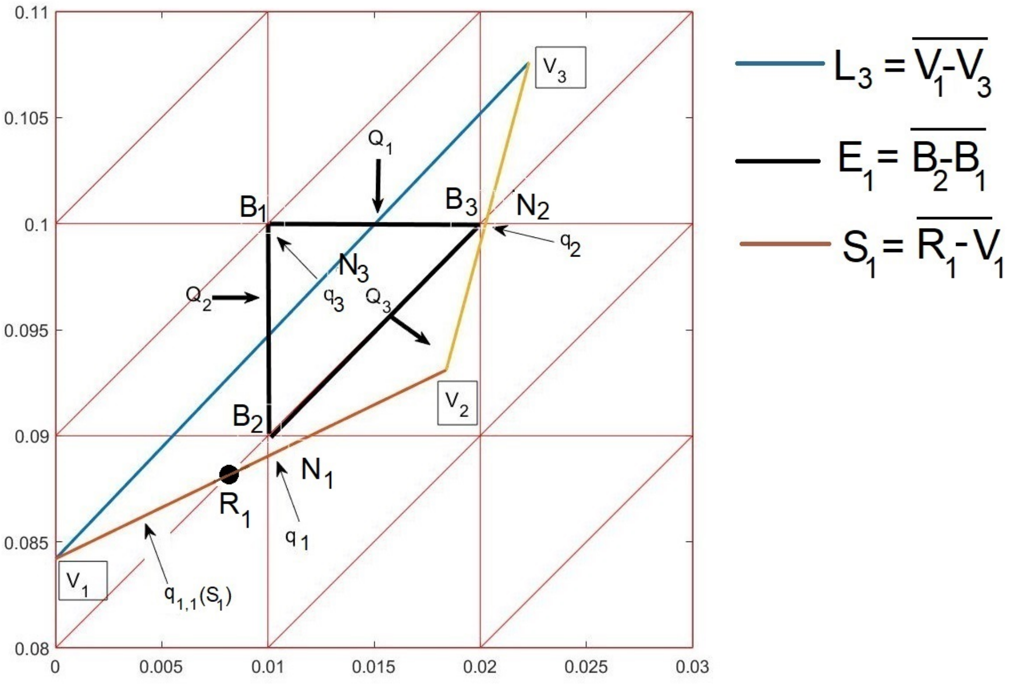


**Figure 1.11:** *NCI* flux projection

# Chapter 2

# Code Verification

This chapter deals with a simple homogeneous test case, in order to verify accuracy of solutions, generated by *Fix, Adaptive* and *Traces.* Numerical results are compared with respect to an analytical case, evaluating the effects upon grid size variation and dispersivity modification. The former could help in understanding the convergence properties, while the latter is concerned with the so called *False Diffusion*, a numerical issue, arising from the discretization of the convective term. The Chapter is organized, starting from the presentation of the problem taken into account, afterwards grid size and dispersivity effects have been analysed and then numerical results showed.

## 2.1 Study case

### Domain

Let $\Omega \subset \mathbb{R}^2$ be a semi-infinite, homogeneous bounded domain
$x \times y$=$[0,200] \times [0,50] m^2$ and $(0,50)s$ be the interval of interest. Boundary $\Gamma$ is decomposed into four borders:

- $\Gamma_{west}$: western border coincides with the inflow part.
- $\Gamma_{south}$ and $\Gamma_{north}$: souther and northern border are impermeable solid walls.
- $\Gamma_{east}$: eastern border coincides with the outflow part.

### Flow Problem

A constant horizonthal velocity is set, in order to approximate the problem to a steady 1D case. Data have been chosen arbitrarily and they are reported beyond:

- Permeability: $k$ =1.0204$\cdot 10^{-9}$ $m^2$
- Pressure at the outlet: $P_o$ =1 $Pa$
- Velocity: $u$ =2 $m/s$
- Porosity: $\varphi$ =0.35

- Darcy's flux: $q_0 = u \cdot \varphi$ =0.7 $m/s$

Darcy's flux $q_0$ has been imposed on the western wall, while pressure value $P_0$ has been fixed at the eastern boundary.

### Transport Problem

Governing equations in the differential System (2.1) describe the transport phenomena, taken into account.

$$\begin{cases} \partial_t c = \nabla \cdot (\mathbf{D}\nabla c - u\, c) \quad \mathbf{x} \in \Omega \times [0, 50] \\ c_D = c(\mathbf{x}_D, t) = 1 \quad \mathbf{x}_D : (x = 0,\, y \leq 20) \\ c(\infty, t) = 0 \quad \forall t > 0 \\ -\mathbf{D}\nabla c \cdot \mathbf{n} = 0 \quad \mathbf{x} \in (\partial\Omega - \Gamma_{west}) \end{cases} \tag{2.1}$$

Velocity has been noted down as scalar quantities.
Neumann boundary condition has been imposed to all the borders, but for the inflow part. A Dirichlet boundary condition has been applied to the western border, while the assumption of semi-infinite domain is analytically translated by the third equation in the System (2.1).
A conservative approach is used to approximate the ADE (first equation of the system (2.1)). Discretization scheme for *Traces* is reported in Appendix A.1, while Section 1.2.2 is concerned with *Fix* and *Adaptive*. Data regarding transport problem are the following:

- $\alpha_L$ =1 $m$ (only for Section 2.2, then it is made vary to reproduce different scenarios of transport process, Section 2.3);
- $\alpha_T$ =0.1 $m$ (only for Section 2.2, then it is made vary to reproduce different scenarios of transport process, Section 2.3);
- $D_m$ =0 $m^2/s$ (in this way molecular diffusion is neglected, in favor of mechanical dispersion);
- $\Delta t$ =0.1 $s$ (for *Adaptive*, $\Delta t_{min}$= 0.5 $s$ due to computational cost).

The given problem could be categorized as a continuous injection case in 2D domain, whose analytical solution is reported in [36].

### Output of interest

In order to provide an efficient code verification, both local and global estimators are adopted to test the code behaviour. The former is evaluated in two different points, $P_1(50, 10)$ and $P_2(50, 20)$. The latter takes into account the whole control volume. A limiting value of 5% upon both errors is arbitrary established as the upper bound of accuracy.
We label:

- BTC as Breakthrough curve;
- $c_h$ as the approximated concentration, generated by each of the three codes;
- $c$ as the concentration value, obtained from Equation (**??**);
- $T$ as generic triangle element;
- $N_t$ as the number of triangle elements;
- $err_r$ as the relative error in $L^2$-norm

$$err_r = \sqrt{\frac{1}{\int_\Omega c^2 \, d\Omega} \sum \int_T (c_h - c)^2 \, dT} \quad \text{with } \log_{10}(c) > \varepsilon \tag{2.2}$$

$\varepsilon = -4$ for grid size variation case (Section 2.2) and $\varepsilon = -2.5$ for dispersivity one (Section 2.3). The constraint on the concentration is automatically translated on a reduction of the number of elements, therefore the summation inside Equation (2.2) does not count for whole mesh cardinality. The choice has been done, because under a certain order of magnitude, there is no distinction for sake of accuracy. Furthermore, the choice of a $L^2$-norm estimator allows to provide a fair comparison between fix and adaptive strategy, because the error is associated to the element size.

## 2.2 Grid size variation

The objective of this section is to quantify effects upon grid size variation. Three different levels of mesh refinement are taken into account. To distinguish them, we label *coarse, medium, fine* grids. For what concerns *Fix* and *Traces*, number of elements is chosen, afterwards mesh size is calculated. Handling with *Adaptive*, only the maximum number of elements ($N_{max}$) is selected, because a single value of mesh size $h$ does not assume any meaning, since we should consider a scalar map.

### 2.2.1 Parameters

The choice of parameters has done considering both fix and adaptive approaches, but only for the *coarse grid.*
We initially consider *Fix* and *Traces.*
Calibration of parameters involves time step $\Delta t$ and mesh size $h$. The latter has been chosen arbitrarily, in order to give an appropriate description of the phenomenon. Instead, $\Delta t$ has been defined as the maximum value, which generates stable local results.
The choice is subjected to three kind of constraints:

- Discretization of time derivative;
- Maximum principle (Equation (2.4) from [37]);
- Courant criterion (Equation (2.5)).

Recalling Equation (2.3), the approximation of the time derivative as incremental difference requires a suitable choice for the denominator, or rather as smallest as possible

$$\frac{\partial c}{\partial t} = \lim_{\Delta t \to 0} \frac{c(t^{n+1}) - c(t^n)}{\Delta t} \tag{2.3}$$

However, according to [37], a critical aspect for Finite Element coincides with the violations of the maximum principle, namely a lower bound in the time step range needs to be accomplished (Equation (2.4)), otherwise oscillations could incur, affecting the accuracy of the result. We remind all codes are built on a Finite Element framework, however Equation (2.4) only affects *Fix* and *Traces*, since adaptation method automatically updates the time step

$$\Delta t_{min} = \sqrt{2}\frac{h^2}{6D} \tag{2.4}$$

Furthermore, in the discretization of convective term, *Traces* uses an explicit scheme, which is also subjected to an upper bound:

$$\Delta t_{max} = \frac{h}{|u|} \tag{2.5}$$

Local breakthrough curve and relative error are computed for the *coarse grid* case, in order to find the correct $\Delta t$, which could fulfil all the constraints.

**Traces and Fix**

| Mesh | $N_x$ | $N_y$ | $N_t$ | $\Delta t$ | $\alpha_L$ | $\alpha_T$ | $h$ |
|---|---|---|---|---|---|---|---|
| Coarse | 200 | 50 | $2{\cdot}10^4$ | 0.2 | 1 | 0.1 | 1.40 |
| Medium | 400 | 100 | $8{\cdot}10^4$ | 0.2 | 1 | 0.1 | 0.70 |
| Fine | 800 | 200 | $32{\cdot}10^4$ | 0.2 | 1 | 0.1 | 0.35 |

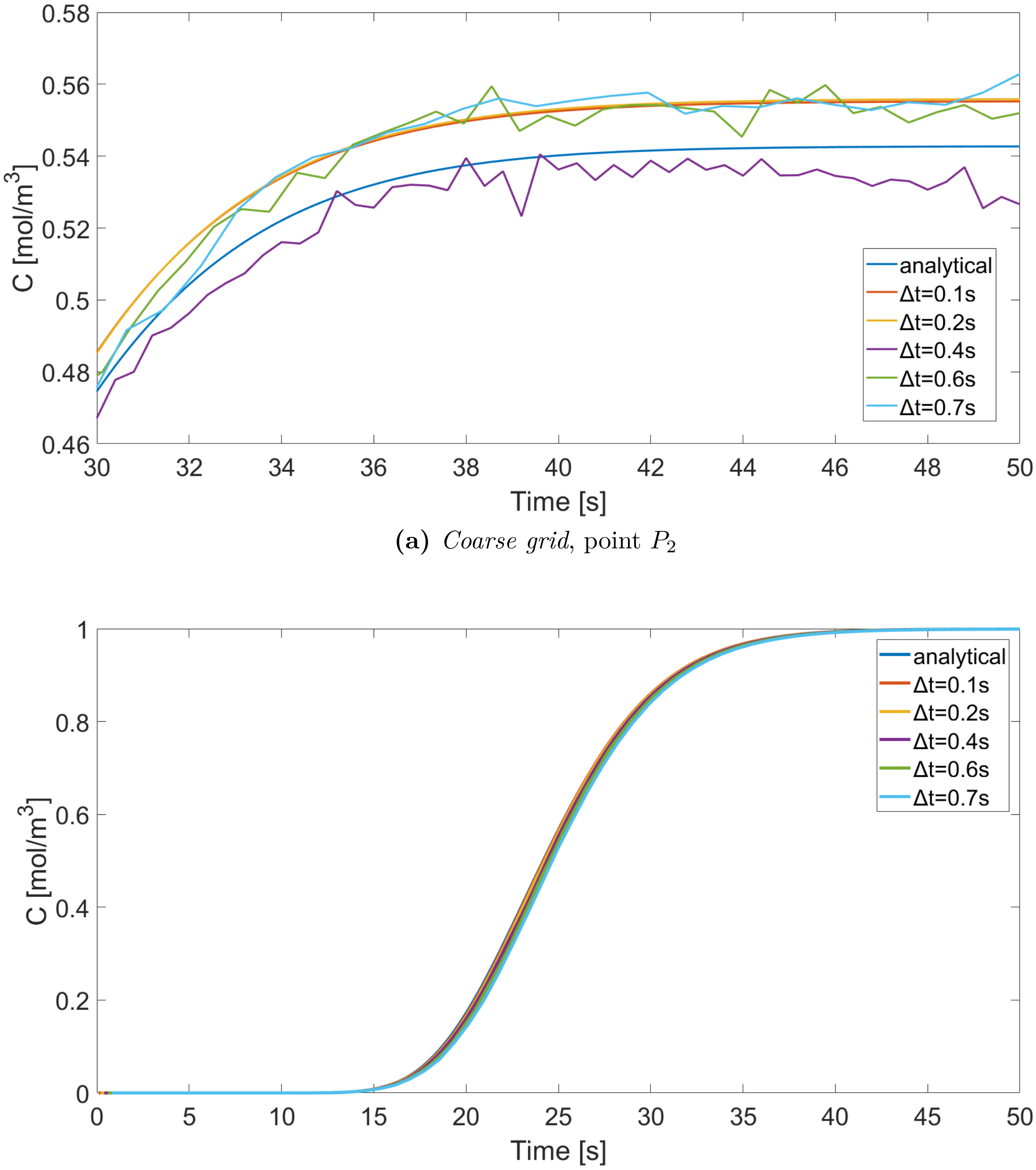
**Table 2.1:** Parameters for Traces and Fix case

The idea is now to consider five different time step values in the range given by Equation (2.4) and (2.5)
($\Delta t = [0.1, 0.2, 0.4, 0.6, 0.7]s$) and evaluate local modifications, by means of breakthrough curve at the given points $P_1$ and $P_2$, and global variations, by means of relative error. Once the first condition will be achieved, the final choice will be the maximum $\Delta t$, which gives the most accurate solution, in terms of $err_r$. The process allows to immediately reveal the instability reduction by changing the time step, whereupon to have a global perception over the whole control volume.
Regarding *Fix*, the solution is only slightly affected by changing the time step, in the considered range. Assuming $\Delta t = 0.1\,s$ or $0.7\,s$, differences are qualitatively (see Figure 2.2) and quantitatively ($err_r = 0.324$ and $err_r = 0.326$ respectively) negligible.
On the other hand, dealing with *Traces*, as it is observed in Figure 2.1, only $\Delta t = 0.1s$ and $\Delta t = 0.2s$ offer acceptable solutions, in terms of lack of oscillations in the concentration evolution, although the correct peak value is not reached. Since the relative error is practically the same for both cases, hence $\Delta t = 0.2s$ is assumed as reference parameter.

As for what concerns *Adaptive*, calibration of parameters deals with (see Section 1.3):

- $\tau_h$: spatial tolerance;
- $\tau_t$: time tolerance;
- $p_{min}$: minimum allowed triangle area;
- $h_{max}$: maximum length of triangle edge;
- $h_{min}$: minimum length of triangle edge;
- $N_{max}$: maximum number of elements in $\mathcal{T}_h$;
- $N_{min}$: minimum number of elements in $\mathcal{T}_h$;
- $\Delta t_{max}$: maximum time step;
- $\Delta t_{min}$: minimum time step;

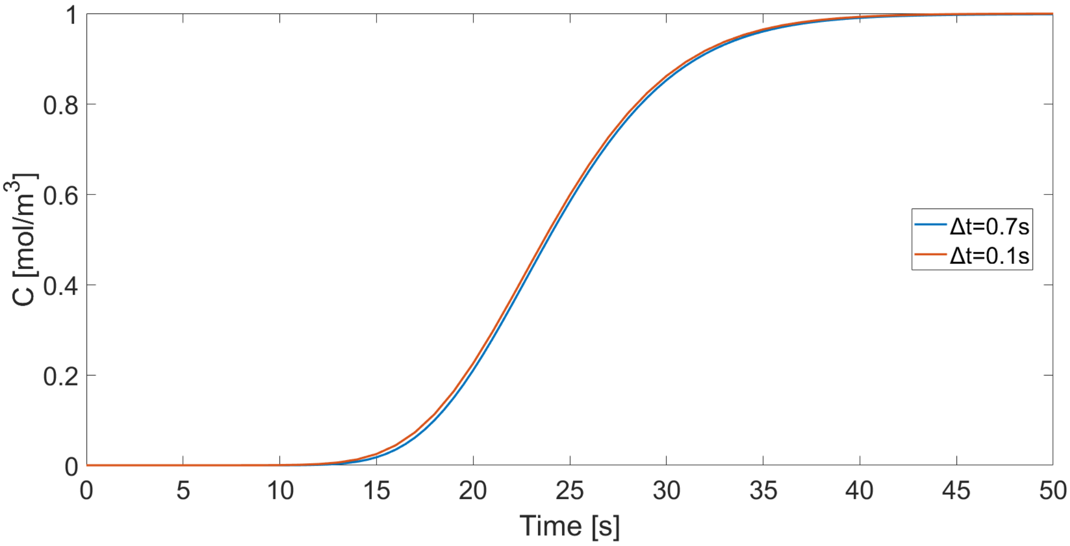
In choosing them, we follow two view-points. The former, by similarity with scientific literature ([9], [3], [11]) and the previous thesis work [19] so $N_{min} = 10^3$, $\Delta t_{max} = 30s$, $h_{min} = 10^{-12}$ have been immediately fixed, without performing any sensitivity analysis, because they affect the adaptive solution in a minor way.
The latter, on the other hand, needs to evaluate two aspects. Starting from data, taken from [19], the consistency of the mesh elements distribution and the accuracy of the solution are checked. The former aspect is a necessary, but not sufficient condition and it is adopted to provide an initial selection over the possible alternatives. To give a better explanation, we focus on Figure 2.3b, which can not be considered consistent with the solution, because of the cluster of elements along the western border. Dirichlet condition has been applied, therefore the profile is constant and none gradient along $x$ and $y$ should be generated and a coarse area might be present, such as in Figure 2.3c. The issue derives from a numerical inefficiency, related to the wrong choice of spatial and time tolerances of error estimator (see Appendix A.2), which are forcing the adaptive code to allocate a larger number of elements, only due to the Dirichlet imposition. This should lead to the attitude to choose $\tau_h$ and $\tau_t$ in accordance with the given problem and not arbitrarily imposed to the minimum value. Finally $\tau_s = 1$ and $\tau_{\Delta t} = 10^{-4}$ are set.
The second step is based on the relative error calculation. Table 2.2 contains all the results, whereupon $p_{min} = 10^{-4}$ and $h_{max} = 40$ ensue. Furthermore, $\Delta t_{min} = 0.5s$ is fixed due to computational reasons.



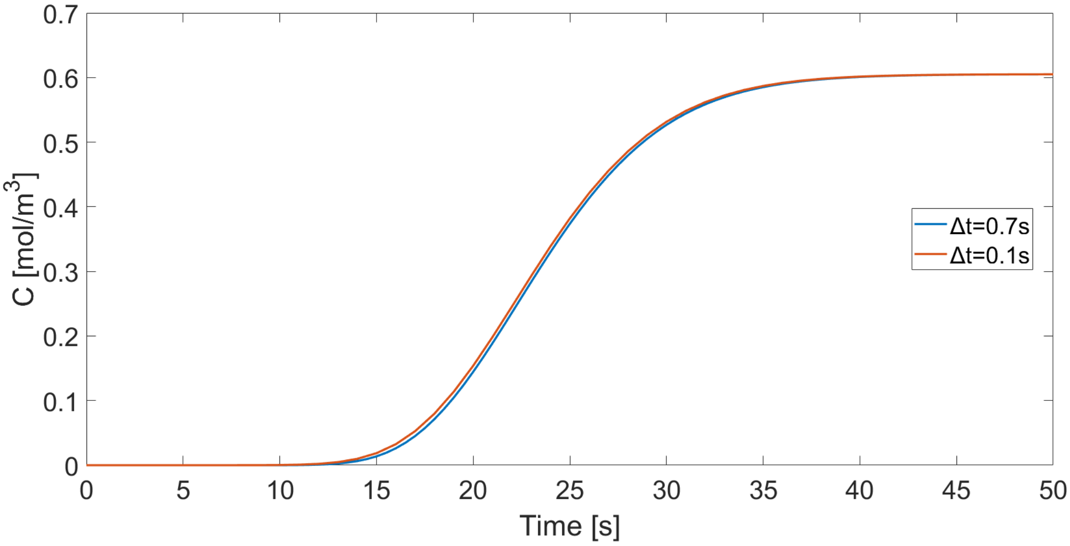
**(a)** *Coarse grid*, point $P_2$

**(b)** *Coarse grid*, point $P_1$

**Figure 2.1:** $\Delta t$ definition by means of local BTC for *Traces*

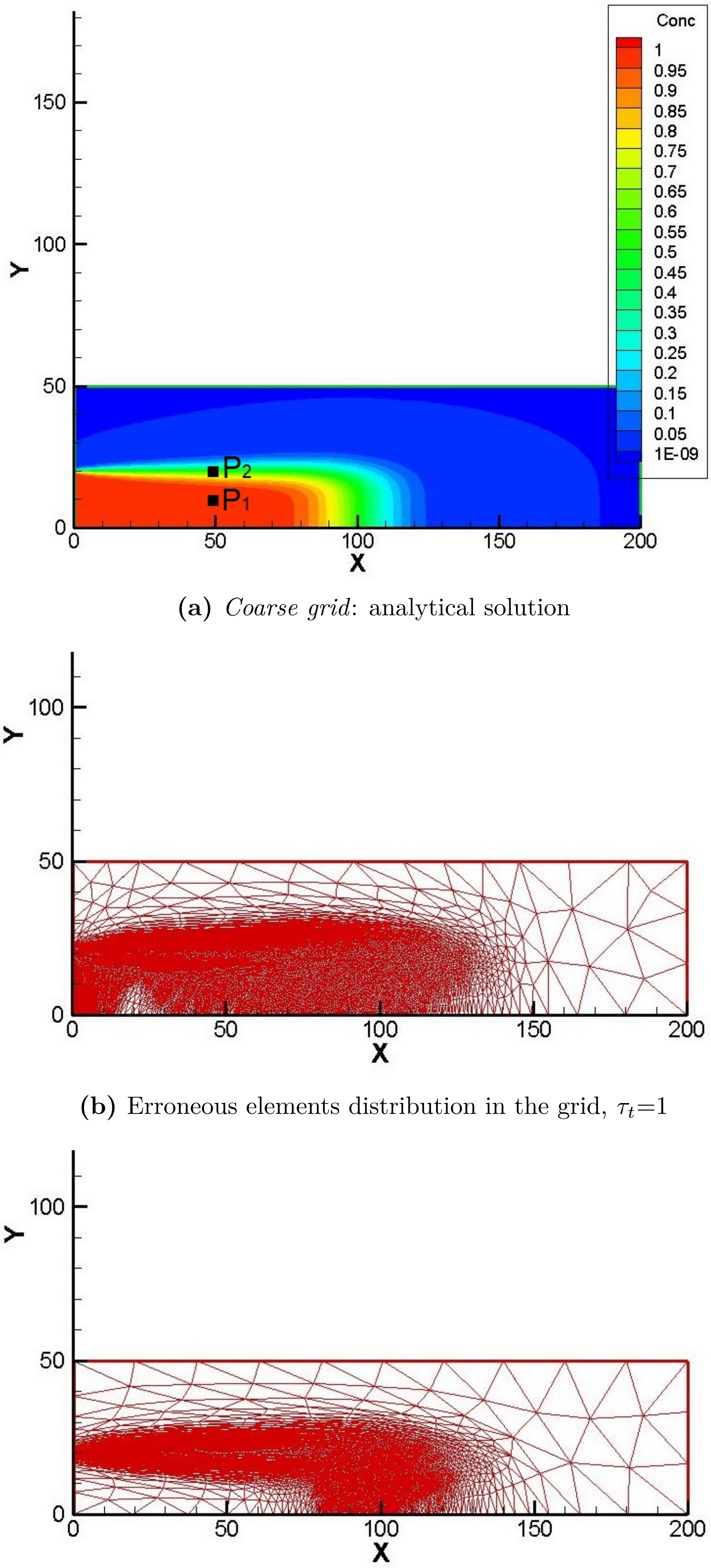


**(a)** *Coarse grid*, point $P_1$



**(b)** *Coarse grid*, point $P_2$

**Figure 2.2:** $\Delta t$ definition by means of local BTC for *Fix*



**(a)** *Coarse grid*: analytical solution

**(b)** Erroneous elements distribution in the grid, $\tau_t$=1

**(c)** Acceptable elements distribtion in the grid

**Figure 2.3:** *Coarse grid*: elements distribution in the mesh, with the associated analytical solution

| $p_{min}$ | $h_{max}$ | $err_r$ [$\log_{10}(c) > -4$] |
|---|---|---|
| $10^{-3}$ | 40 | 0.0425 |
| $10^{-4}$ | 10 | 0.0412 |
| $10^{-4}$ | 40 | 0.0401 |

**Table 2.2:** Adaptive parameters calibration for *coarse grid*

| Adaptive | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Mesh | $N_{min}$ | $N_{max}$ | $\Delta t_{min}$ | $\Delta t_{max}$ | $\tau_s$ | $\tau_{\Delta t}$ | $h_{min}$ | $h_{max}$ | $p_{min}$ | $\alpha_L$ | $\alpha_T$ |
| Coarse | $10^3$ | $1 \cdot 10^4$ | 0.5 | 30 | 1 | $10^{-4}$ | $10^{-12}$ | 40 | $10^{-4}$ | 1 | 0.1 |
| Medium | $10^3$ | $2 \cdot 10^4$ | 0.5 | 30 | 1 | $10^{-4}$ | $10^{-12}$ | 40 | $10^{-4}$ | 1 | 0.1 |
| Fine | $10^3$ | $4 \cdot 10^4$ | 0.5 | 30 | 1 | $10^{-4}$ | $10^{-12}$ | 40 | $10^{-4}$ | 1 | 0.1 |

**Table 2.3:** Grid size variation: parameters for *Adaptive* case

### 2.2.2 Results

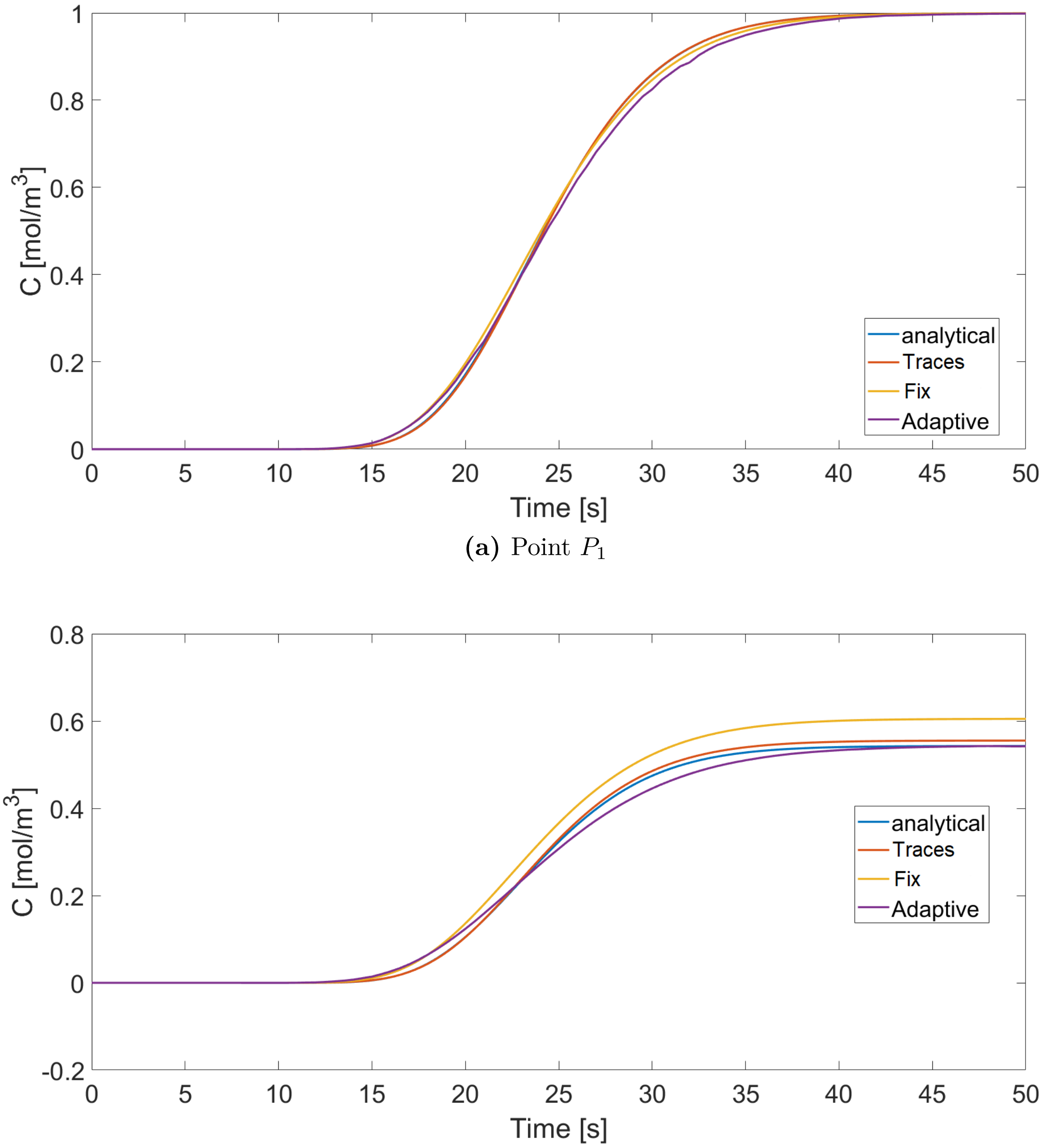
Local breakthrough curves (BTC) at the given points are depicted in Figures 2.4, 2.5, 2.6. At first sight, regarding point $P_1$, all trends, independently on the type of mesh, describe the concentration evolution in an appropriate way. On the contrary, $S$ curves of $P_2$ for *Fix* and *Traces* do not catch the correct constant value, despite the refinement. However, the gap is negligible.
Error trend is depicted on Figure 2.8 and quantitative values are reported in Table 2.4. All the codes reproduce in a suitable way the test case, with a minimum discrepancy in the relative error of 5%, which could be considered acceptable, for sake of accuracy. It is important to clarify that for *Adaptive*, maximum number of elements has been taken into account to represent Figure 2.8.
We note the constraint for the relative error in Equation (2.2) implies a bound on the number of elements. For *Fix* and *Traces*, 47% of the total cardinality is considered (independently on the refinement level), therefore 53% of the domain is characterized by negligible concentration values in the order of magnitude. Instead, *Adaptive* reaches 98%. This represents the perfect example of space adaptation, in fact the majority of triangles elements are laid in the region, where the front is spreading.

We now focus on Figure 2.8, where the behaviour of *Traces* seems to diverge, despite the mesh refinement. The reason comes from the time step, in fact a calibration for a suitable $\Delta t$ might have be done for each kind of mesh level and not only for the coarse one. According to Equation (2.5), decreasing $h$, $\Delta t_{max}$ decreases as well, however, the choice of maintaining a single value has been adopted due to computational cost, especially towards *Fix*. In this case, the *coarse mesh* is coupled with an appropriate time step and it could yield a better distribution than the *fine grid*.
Another interesting comparison is depicted on Figures 2.7a and 2.7b which respectively show the CPU trend against number of unknowns (edges for *Traces* and vertices for *Fix* and *Adaptive*) and relative error. Simulations have been run with a

processor Intel(R)Core(TM)i7-7500 CPU @2.70 GHz-2.90 GHz and 12 GB of memory RAM installed. It is important to underline all the codes are implemented with the same direct solver, UMFPACK, or rather a multi-frontal Gauss LU factorization. This makes the comparison fair, in terms of computational cost, because system of equations are solved in the same manner. However, the difference is remarkable, *Traces* could save one order of magnitude of computational memory and the accuracy in solution is always maintained at high level.

| | **Coarse Grid** | **Medium Grid** | **Fine Grid** |
|---|---|---|---|
| Code | $err_r$ | $err_r$ | $err_r$ |
| Traces | 0.0317 | 0.0592 | 0.1161 |
| Fix | 0.3238 | 0.1228 | 0.0490 |
| Adaptive | 0.0401 | 0.0391 | 0.0153 |

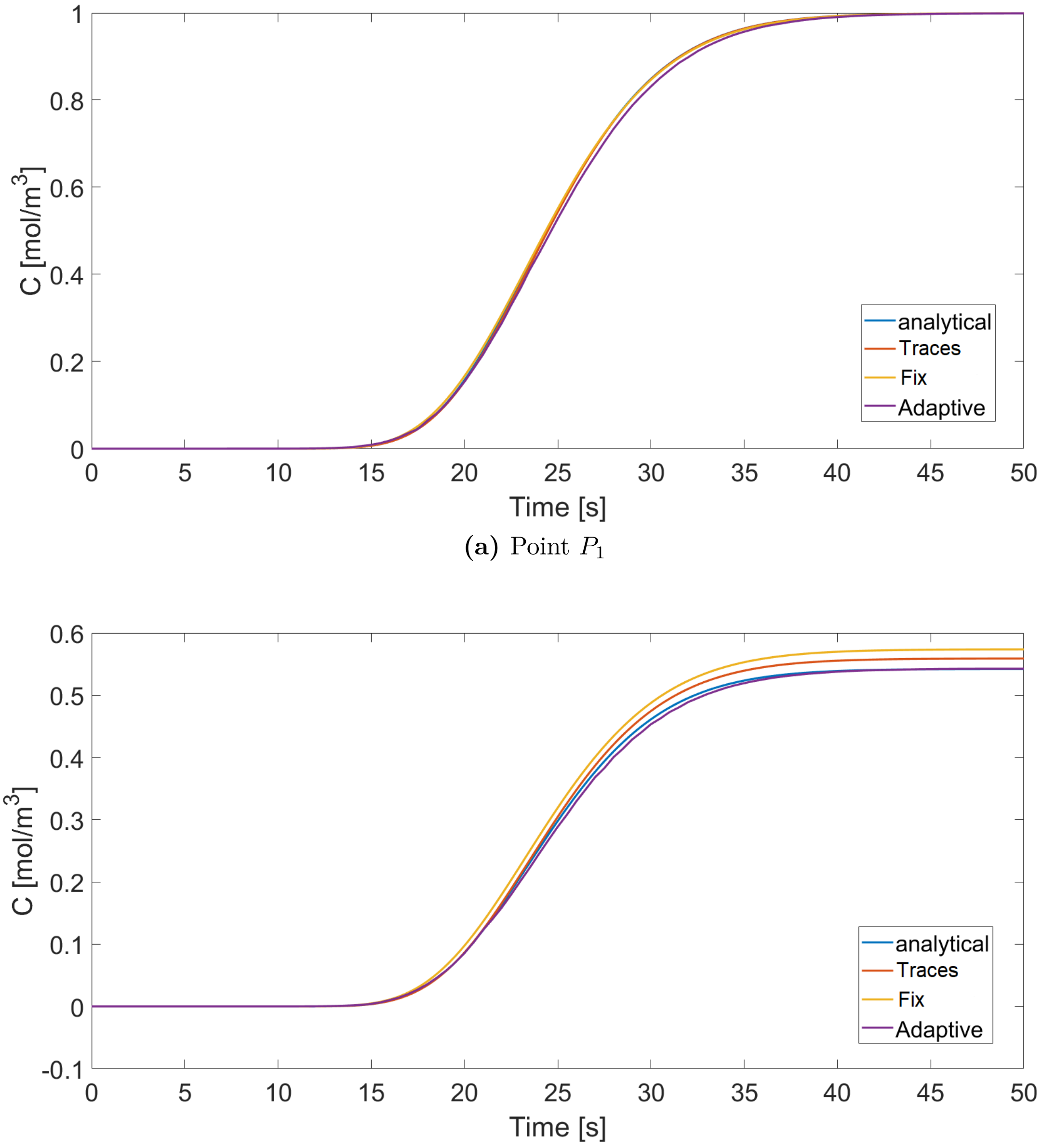
**Table 2.4:** Grid size variation: relative error



**(a)** Point $P_1$

**(b)** Point $P_2$

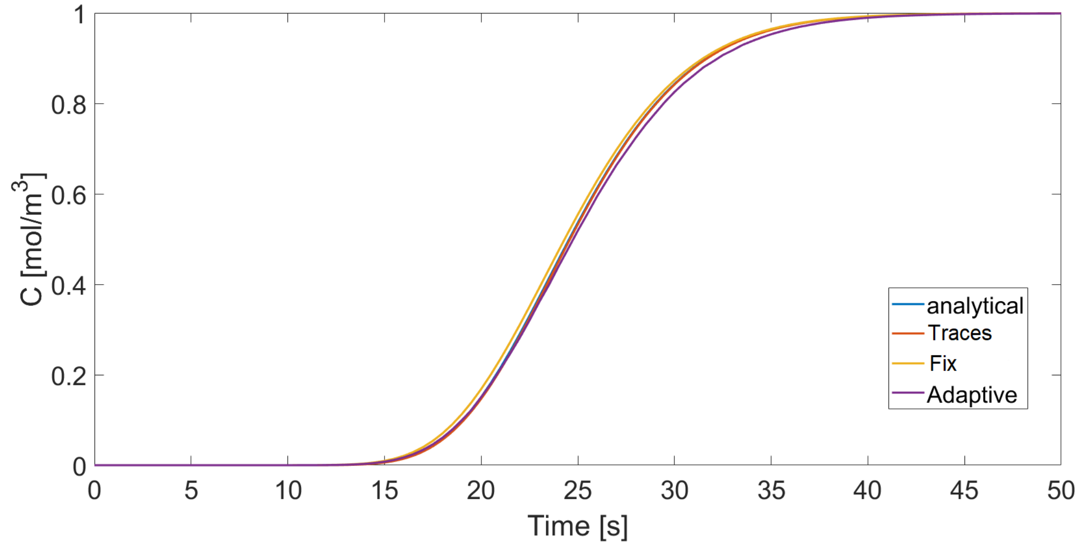
**Figure 2.4:** *Coarse grid*: local BTC



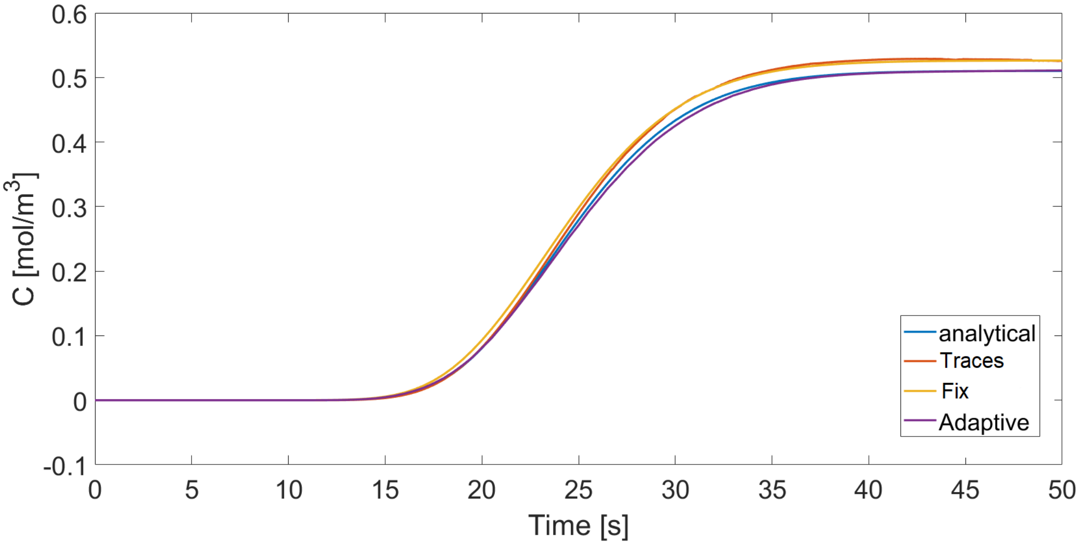
**(a)** Point $P_1$

**(b)** Point $P_2$

**Figure 2.5:** *Medium grid*: local BTC

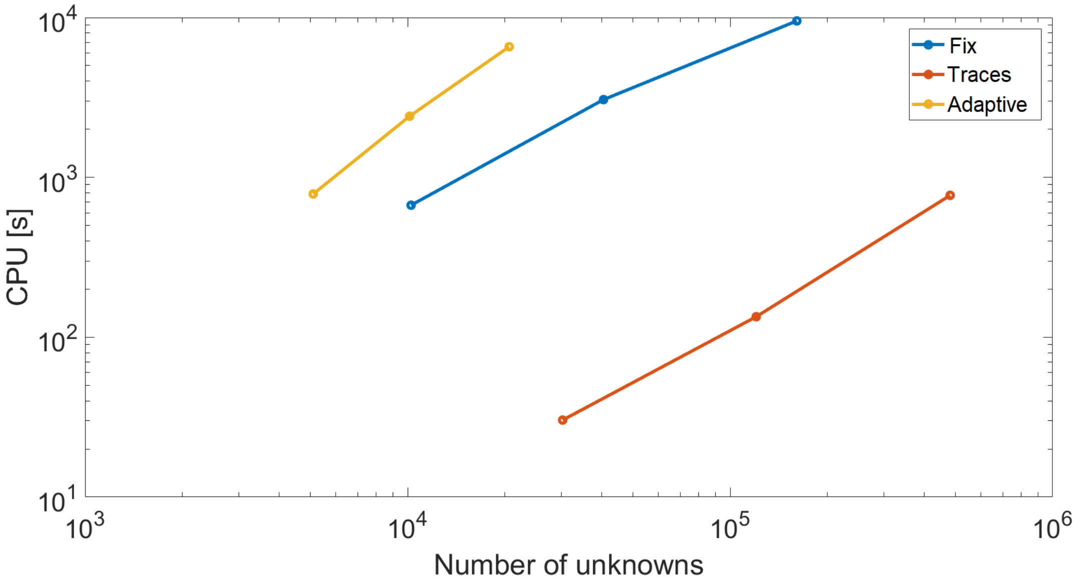


**(a)** Point $P_1$

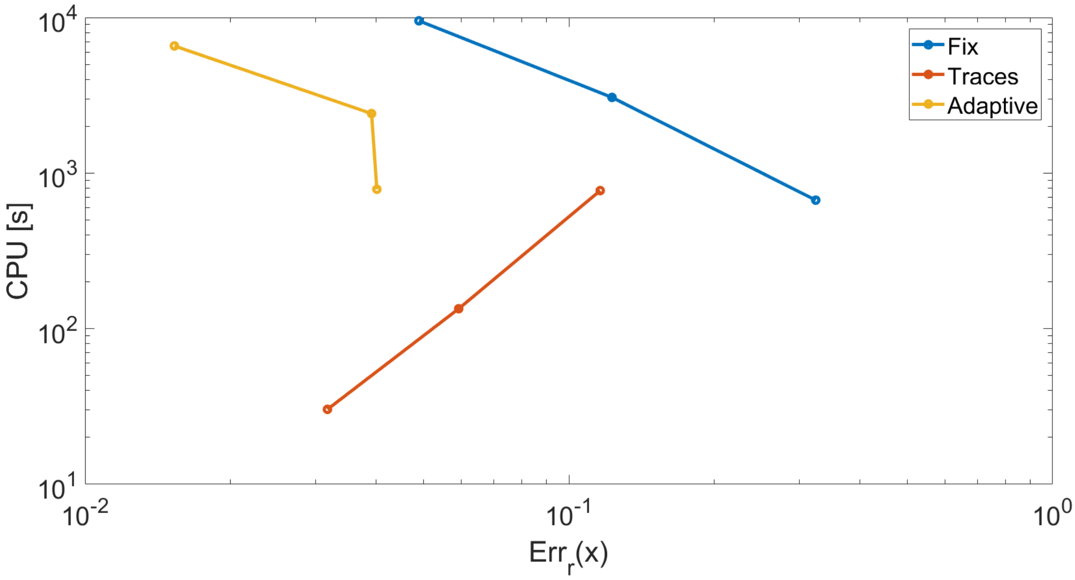


**(b)** Point $P_2$

**Figure 2.6:** *Fine grid*: local BTC



**(a)** CPU trend against number of unknowns



**(b)** CPU trend against relative error

**Figure 2.7:** CPU trend

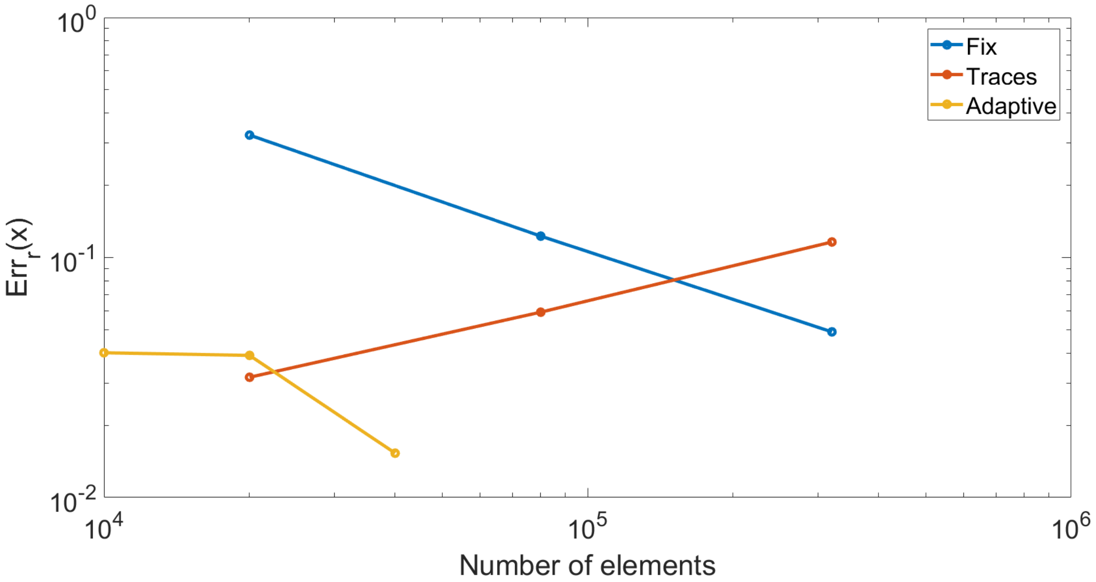


**Figure 2.8:** Grid size variation: relative error against number of elements

## 2.3 Dispersivity variation

According to Section 1.2.2, in case of transport dominated by advection, discretization of the convective term could induce numerical inaccuracies in the final result. The relative strength between diffusive and advective flux can be quantified in terms of grid Péclet number, which is defined as:

$$Pe_h = \frac{|u|\,h}{D} = \frac{h}{\alpha_L} \tag{2.6}$$

with $D = |u|\alpha_L$.
$Pe_h$ can be adopted as reference parameter to carry out different transport scenarios. The objective of this section is to derive numerical simulations to figure out effects upon dispersivity variations, whereupon *low, medium, high* values of grid Péclet number are taken into account. At the same time, *medium grid* is adopted for spatial discretization, since it offers an acceptable trade-off, between computational cost of simulations and accuracy of the solution. Time step is always fixed to $\Delta t = 0.2\,s$ ($\Delta t_{min} = 0.5\,s$).
Numerical inaccuracies may arise due two reasons both related to:

**False Diffusion:** also called numerical diffusion is an artificial smearing of jumps, discontinuities. Neglected terms in the Taylor series expansion of first derivative contribute to the rise of diffusion. Further details in Appendix B.2

**Numerical Dispersion:** causing overshoots, oscillations. Dispersion means that different components of Fourier expansion (of numerical solution) move with different velocities, for example shorter wavelengths move slower than the velocity of the flow.

We focus only on false diffusion (in Appendix B.2 $Pe_h = 2$ is assumed as critical value for the generation of false diffusion) and it is important to underline that for *Adaptive* case, none technique has been employed to limit the false diffusion. The automatic refinement allows to decrease the mesh size and so the $Pe_h$ could remain below the critical value in the region of interest. Furthermore, according to Equation (2.6), it is not possible to furnish a single value of $Pe_h$, because the mesh size varies throughout the domain, therefore a scalar map ensues.

### 2.3.1 Parameters

Péclet number is made varied, by changing longitudinal dispersivity.
Table 2.5 and 2.6 contain the parameters adopted for each code used. As for what concerns *Adaptive*, quantities enlisted in Table 2.3 are initially employed, however, since the problem is becoming more and more difficult, because the concentration front tends to be sharper and sharper, a refinement in terms of minimum triangle area will be provided. Different values of $p_{min}$ will be used and relative error will be evaluated.

| **Traces and Fix** | | | | |
|---|---|---|---|---|
| *Type* | $Pe_h$ | $\alpha_L$ | $\alpha_T$ | $h$ |
| Low | 0.7 | 1 | 0.1 | 0.70 |
| Discrete | 7 | 0.1 | 0.01 | 0.70 |
| High | 70 | 0.01 | 0.001 | 0.70 |

**Table 2.5:** Parameters for Traces and Fix case

| **Adaptive** | | |
|---|---|---|
| *Type* | $\alpha_L$ | $\alpha_T$ |
| Low | 1 | 0.1 |
| Discrete | 0.1 | 0.01 |
| High | 0.01 | 0.001 |

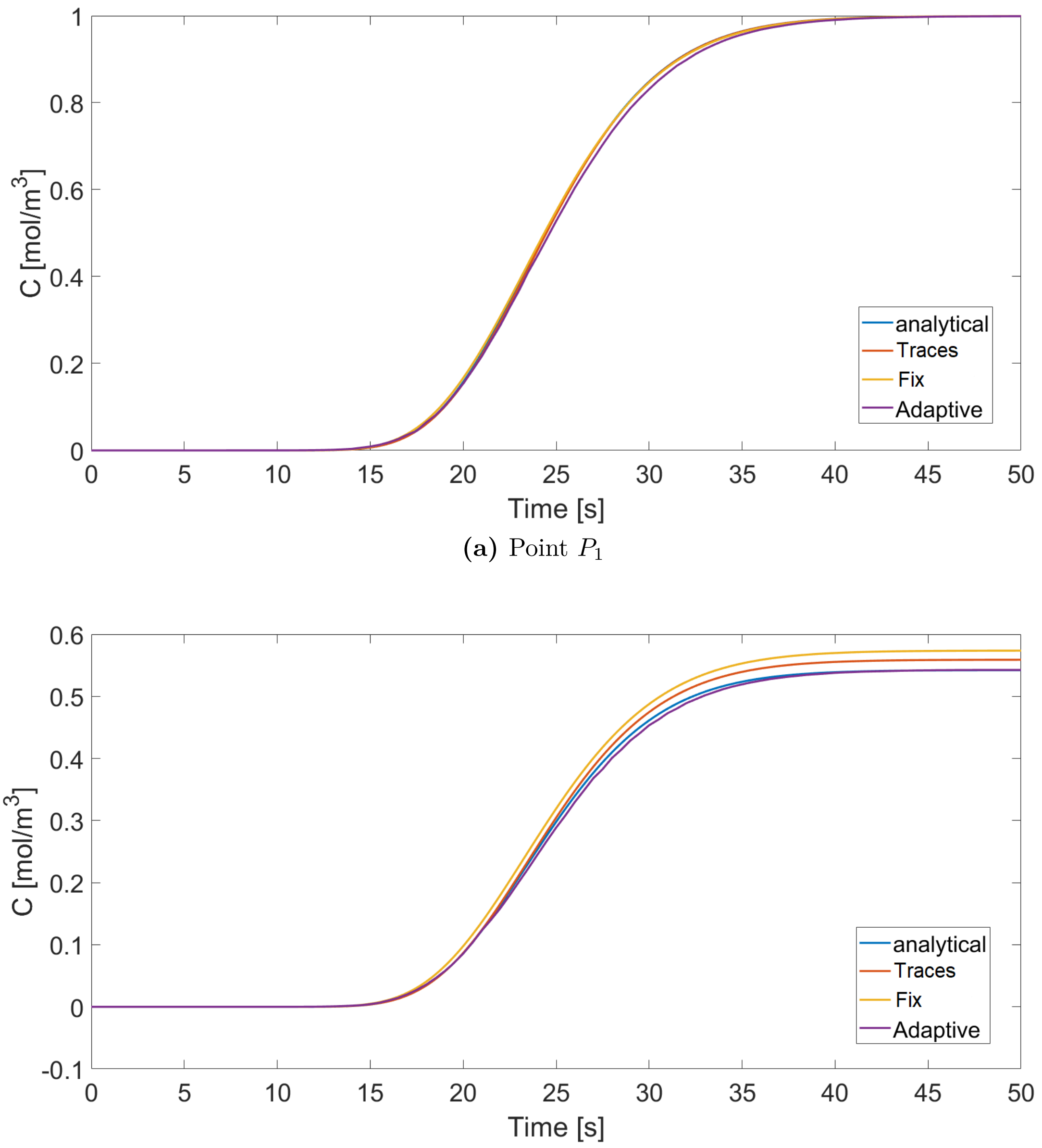
**Table 2.6:** Parameters for Adaptive case

## 2.3.2 Results

### Case *Low Péclet*

Figure 2.11a depicts the concentration map, provided by the analytical solution. Points $P_1$ and $P_2$ lay respectively in the initial part of the concentration front (red region) and in the mixing area (green zone). The solute is quite spread along the longitudinal direction, while the transversal diffusion is limited, as consequence of the choice of $\alpha_L$ and $\alpha_T$. This phenomenon will become more and more accentuated, once we will investigate the case type *Discrete* and *High* (Figures 2.11b and 2.11c). Local BTC are depicted on Figure 2.9. Regarding $P_1$, the trend is well respected and none of the codes detect particular differences. On the other hand, $P_2$ lays in the mixing region, which is extremely subjected to gradients along the vertical direction and this may explain the inaccuracy, especially for *Fix* and *Traces*, in reaching the exact constant value.

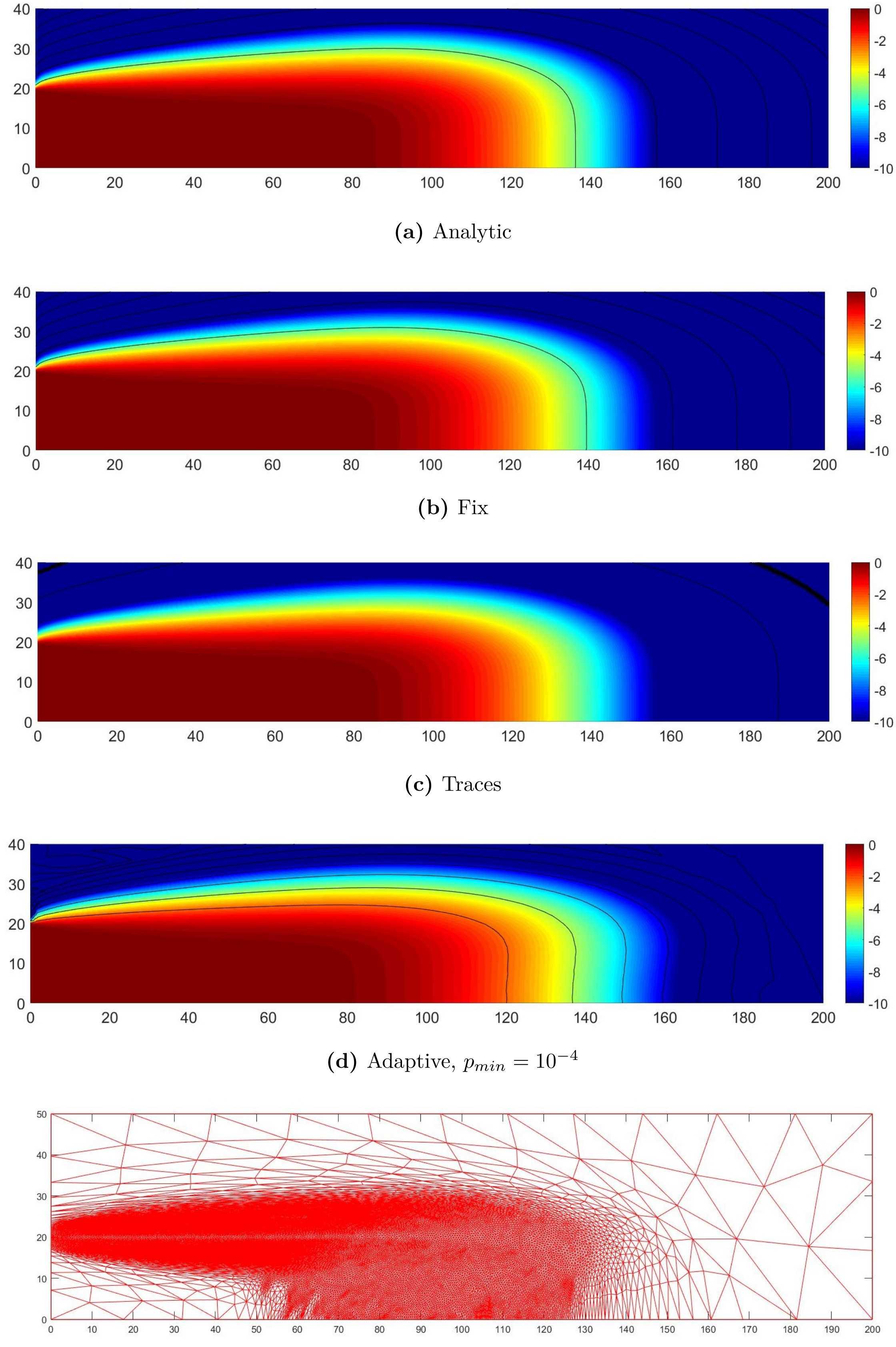
The same deviation does not arise for *Adaptive*, because of the high triangle density in the entire zone of interest (Figure 2.10e). It is important to remind that parameter $p_{min}$, in Table 2.3, is proportional to the minimum area, whose value is much smaller than the element area of the medium regular grid (0.125 $m^2$). Figure 2.10d collects the logarithmic maps of each code. Quantitative error values have been reported in Table 2.4.



**(a)** Point $P_1$

**(b)** Point $P_2$

**Figure 2.9:** *Low* $Pe_h$: local BTC



**(a)** Analytic

**(b)** Fix

**(c)** Traces

**(d)** Adaptive, $p_{min} = 10^{-4}$

**(e)** Adaptive grid

**Figure 2.10:** *Low* $Pe_h$: logarithmic maps at $t = 50s$

### Case *Discrete Péclet*

According to Table 2.6, $Pe_h > 2$ and artificial diffusion could incur, without any numerical technique to limit it. *Fix* is the code, which suffers most (see Appendix B.2) and this is quite evident from the shape of $S$ curve in Figure 2.12a. It intersects the analytical trend, almost at half of the initial injection, therefore the advective field is working suitably, whereas the slope of the curve is lower, meaning that diffusion has raised artificially. Analogously, the logarithmic concentration map, Figure 2.13b, is speeded up in the $x$ direction. This instability derives from the upwind scheme, employed for the discretization of the convective term, as discussed in Appendix B.2 and Section 1.2.2. Same phenomenon occurs in a little scale for *Adaptive*, in fact, roughly speaking, according to Figure 2.13d, it could describe in an accurate manner values of concentration in the order of $[10^{-2}, 10^{-3}]$, that coincide with the orange and yellow front ($x \in [100, 120]m$). On the contrary, focusing on Figure 2.13e, $\forall x > 120m$, the element size becomes coarser and coarser, whereupon the ensuing concentration field might becomes weak in accuracy. Table 2.7 includes spatial error calculations, where the constraint on concentration has been relaxed ($\varepsilon = -2.5$). As discussed before, different $p_{min}$ have been taken into account and, as expected, a reduction in the minimum triangle area is required. For *Traces* and *Fix*, in Table 2.7 only one row has been filled in, since the calibration of $p_{min}$ only interests *Adaptive*.

| Adaptive | | Traces | Fix |
|---|---|---|---|
| $p_{min}$ | $err_r$ | $err_r$ | $err_r$ |
| $10^{-4}$ | 0.059 | - | - |
| $10^{-6}$ | 0.060 | - | - |
| $10^{-9}$ | 0.051 | 0.033 | 0.893 |
| $10^{-12}$ | 0.172 | - | - |

**Table 2.7:** *Discrete* $Pe_h$: relative error

### Case *High Péclet*

Local BTC and logarithmic concentration map are respectively represented in Figure 2.14 and 2.15. Same kind of considerations, adopted for the *Discrete case*, could be done, but the artificial diffusion effects are even worse. Another aspect to pinpoint is the oscillation in local solution for *Adaptive*, in fact the trend in Figure 2.14b is not smooth, despite the high value of element density. Moreover, in Figure 2.15d, the contour lines are very damp, but this is consistent with the spatial grid topology.

| Adaptive | | Traces | Fix |
|---|---|---|---|
| $p_{min}$ | $err_r$ | $err_r$ | $err_r$ |
| $10^{-9}$ | 0.118 | - | - |
| $10^{-12}$ | 0.110 | 0.10 | >1 |
| $10^{-15}$ | 0.172 | - | - |

**Table 2.8:** *High* $Pe_h$: relative error

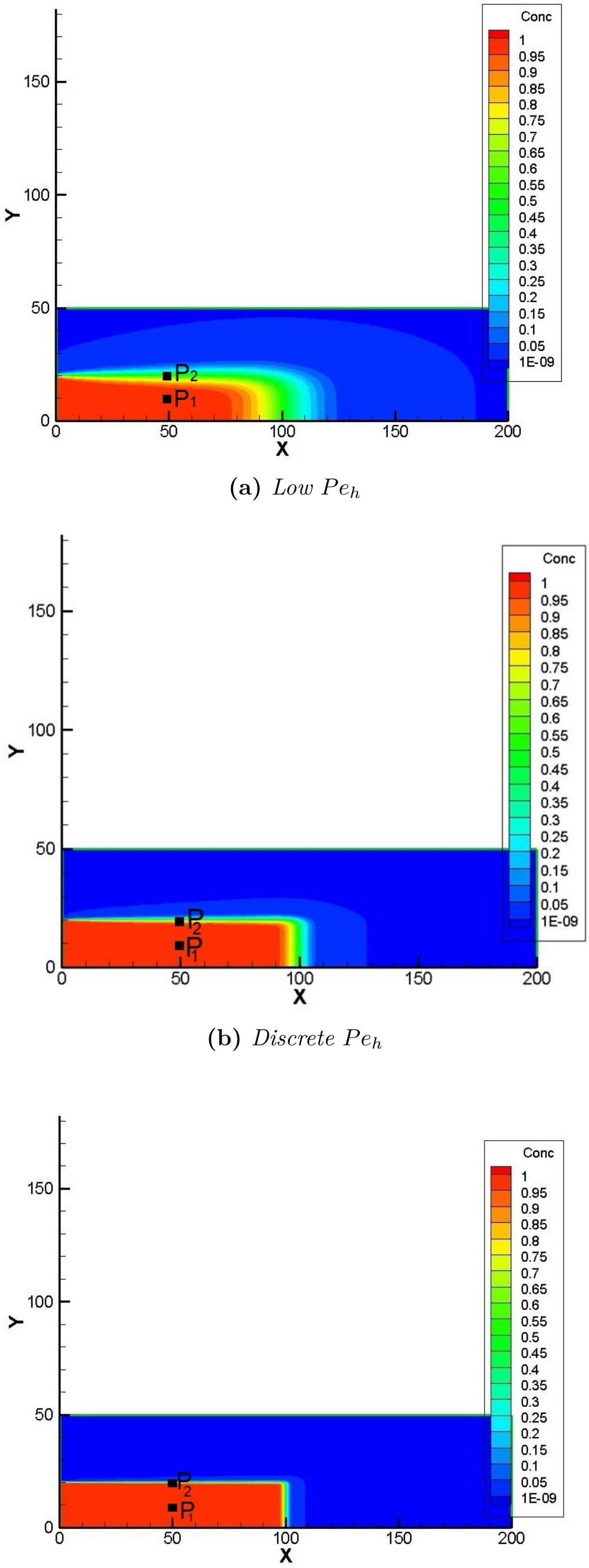
## 2.4 Code comparison

Final remarks might be given, in order to identify weak and strong aspects of each code. First of all, it is important to specify that in safe conditions, namely no artificial diffusion, *Fix, Adaptive* and *Fortran* yield acceptable results, which are consistent, stable and accurate with respect to the analytical solution, therefore the code verification has been checked and it could be considered valid also for other kind of problems, where the analytical solution is not available.
Pros and Cons are here underlined:

**Traces** : the code is well optimized and it realizes accurate solutions, with the minimum CPU expenditure, furthermore, the adoption of Discontinuous Finite Element (see Appendix A.1) allows to balance the effects of false diffusion in case of high Péclet values. On the contrary, calibration of input parameters, especially $\Delta t$ is not straightforward, since it is subjected to three constraints, which are strictly dependent on the mesh size.

**Fix** : the code developed is stable and practically independent on the time step chosen, thanks to the adoption of the correct value in the $\vartheta$-method (see Section 1.2.2). Convergence of the solution has been demonstrated, eventhough it requires a significant amount of elements, to reach very low error values. It is very costly, in terms of CPU, but the great shortage derives from a suitable technique to limit artificial diffusion (see Section 1.2.2).

**Adaptive** : the code has to be interpreted as an enhancement of *Fix*, in fact it allows to refine locally the mesh and obtain very accurate results and it can counter-balance the effects of false diffusion, without implementing any particular techniques. However it could induce oscillations in the concentration map, especially in the coarser regions.

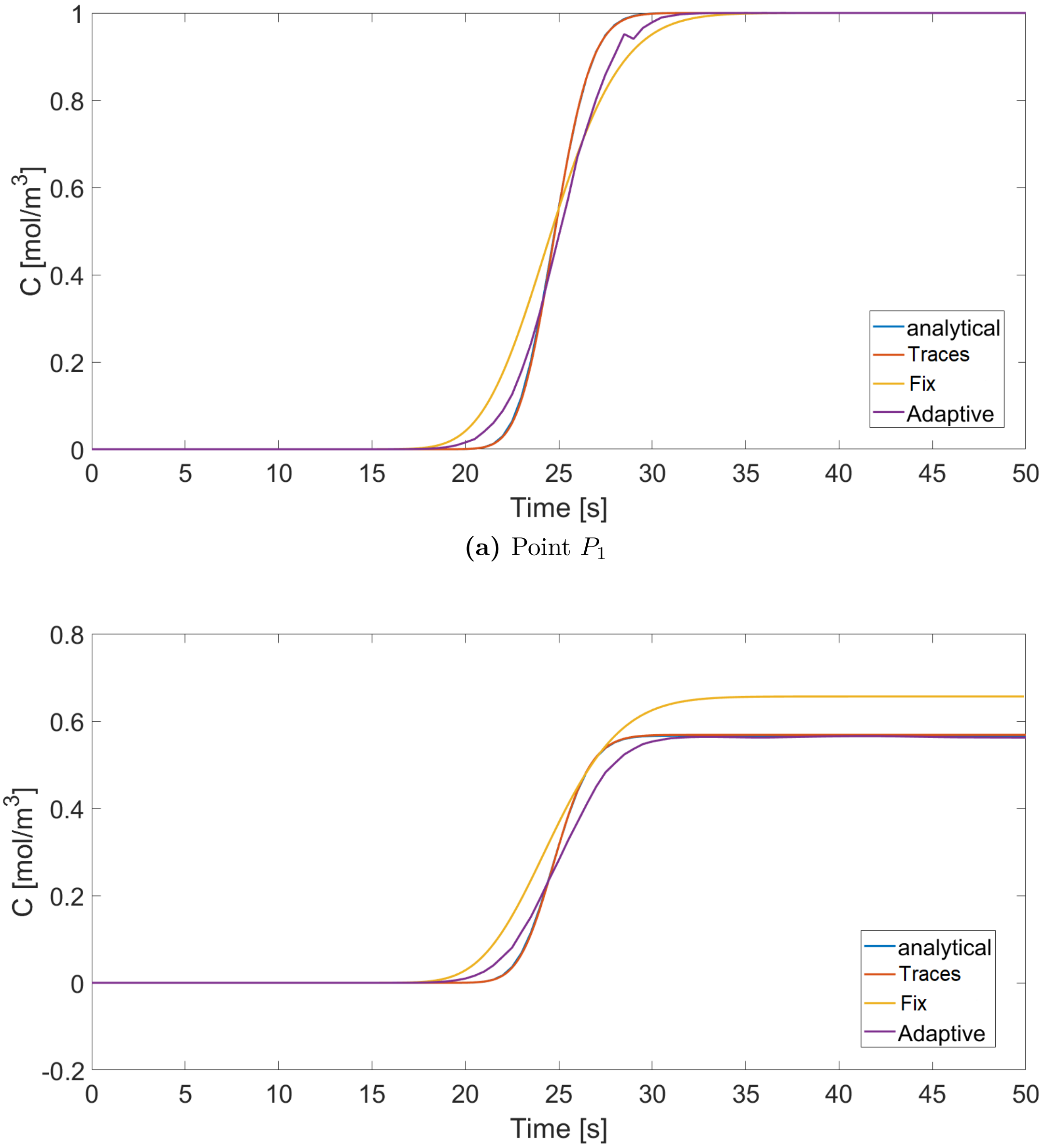


**(a)** *Low $Pe_h$*

**(b)** *Discrete $Pe_h$*

**(c)** *High $Pe_h$*

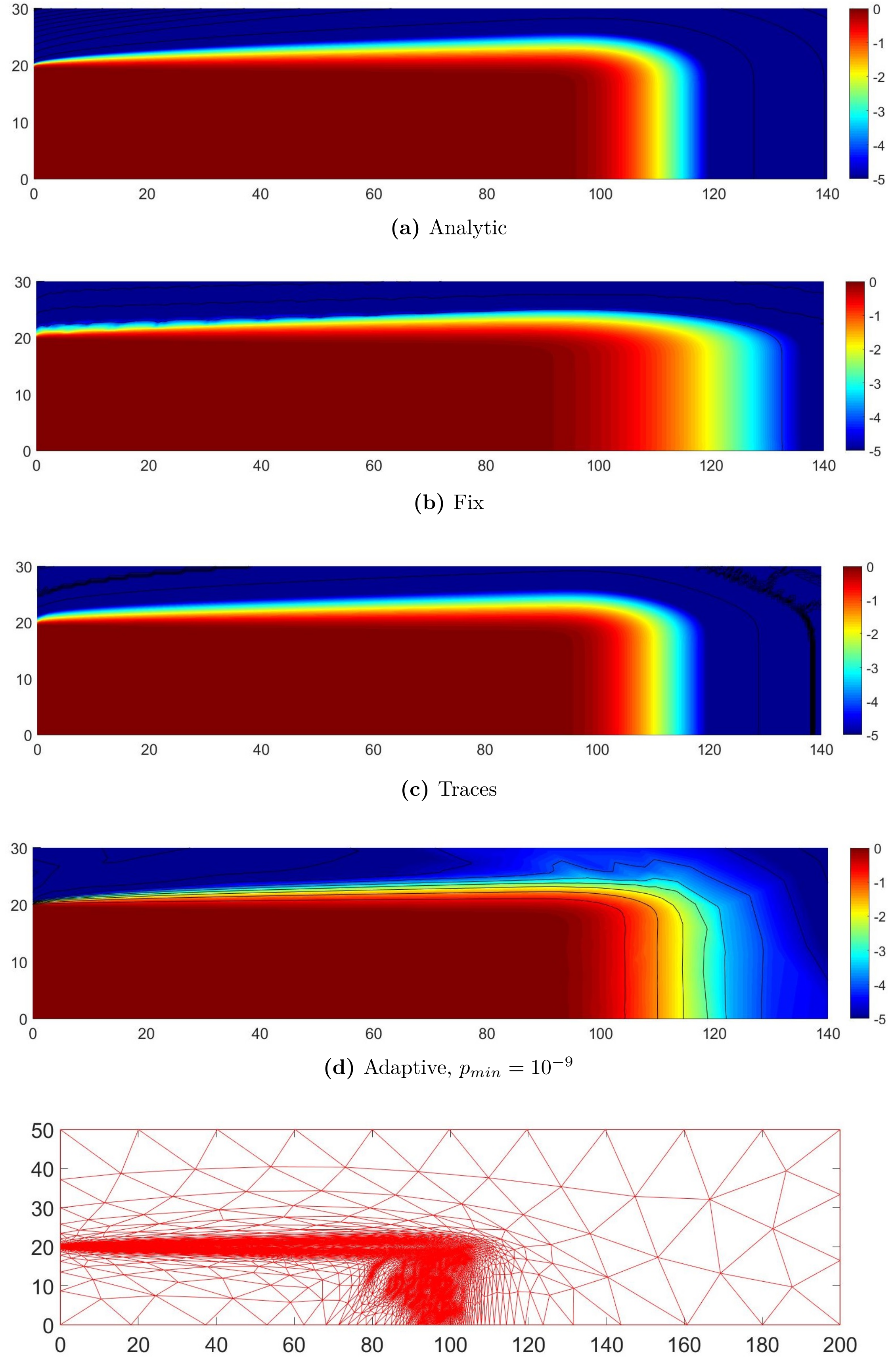
**Figure 2.11:** Analytical solution at time $t = 50s$



**(a)** Point $P_1$

**(b)** Point $P_2$

**Figure 2.12:** *Discrete* $Pe_h$: local BTC



**(a)** Analytic

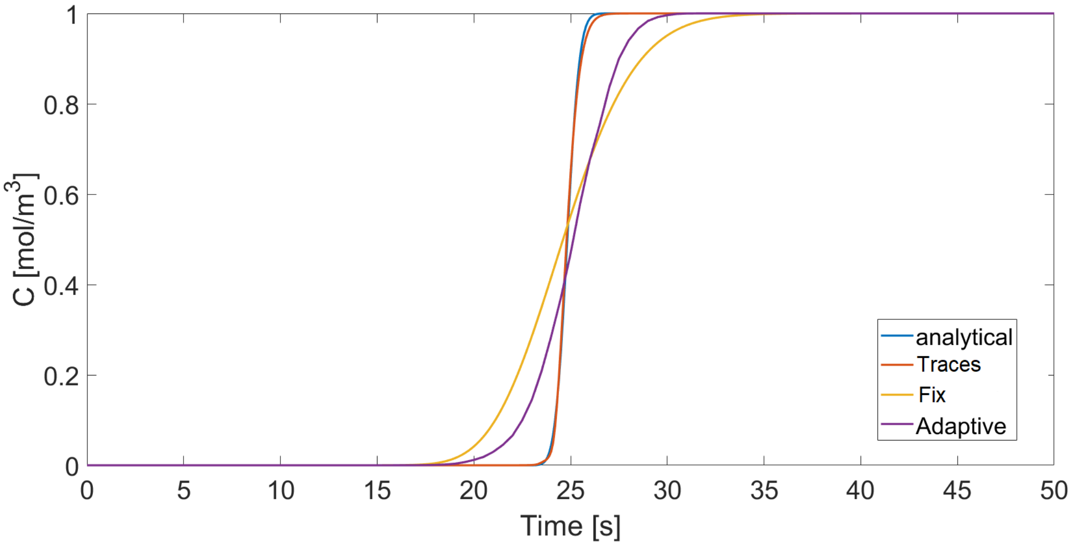
**(b)** Fix

**(c)** Traces

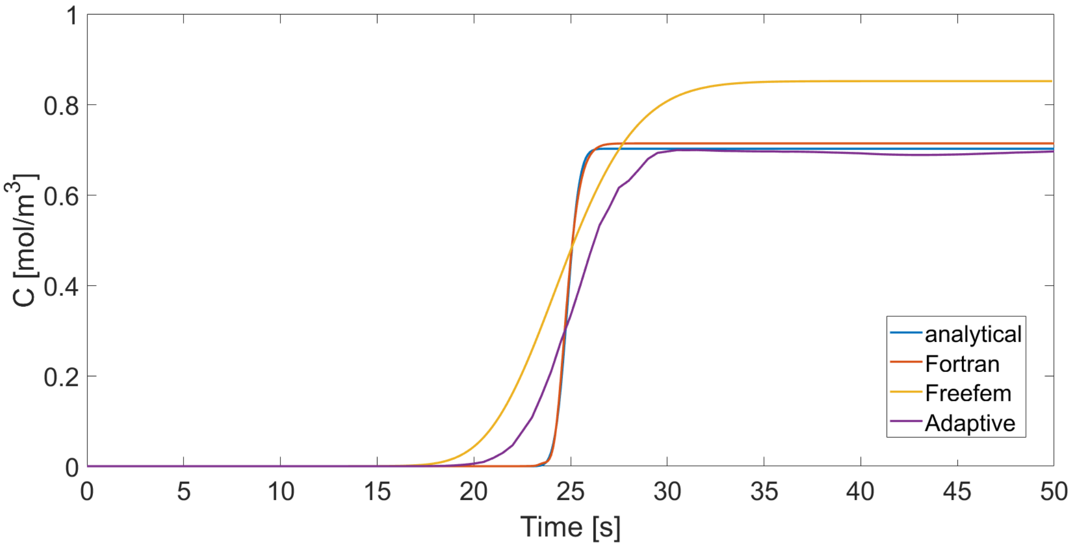
**(d)** Adaptive, $p_{min} = 10^{-9}$

**(e)** Adaptive grid

**Figure 2.13:** *Discrete* $Pe_h$: logarithmic maps at $t = 50s$

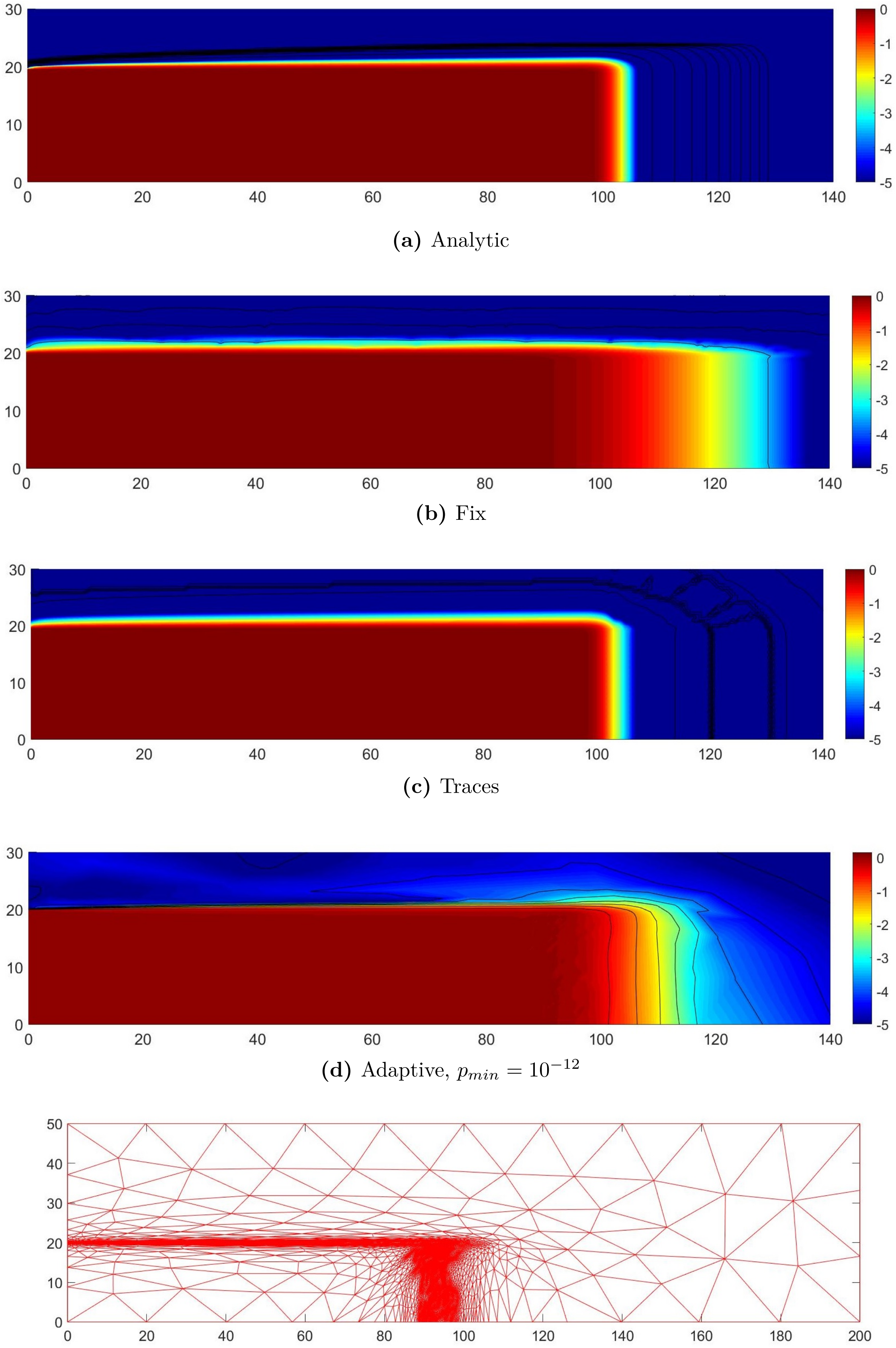


**(a)** Point $P_1$



**(b)** Point $P_2$

**Figure 2.14:** *High* $Pe_h$: local BTC



**(a)** Analytic

**(b)** Fix

**(c)** Traces

**(d)** Adaptive, $p_{min} = 10^{-12}$

**(e)** Adaptive grid

**Figure 2.15:** *High* $Pe_h$: logarithmic map at $t = 50s$

# Chapter 3

# Numerical Results

This chapter is concerned with the results obtained from the approximation of the Darcy's law coupled with the Advection-Dispersion equation. Different simulations have been run, in order to test the behaviour of solute transport in highly heterogeneous field, with complex structure of velocity and concentration map.
This chapter is organized as follows:

Section 3.1 introduces the problem at hand, first the bounded heterogeneous domain is presented and then flow and transport equation are taken into account.

Section 3.2 deals with the observable variables and it identifies the quantities of interest.

Section 3.3 shows numerical results. First a general comparison among *Fix*, *Traces* and *Adaptive* is given, afterwards the aim is shifted towards the verification of the conservation properties and the effects upon the concentration distribution and evolution, for the adaptive case.

## 3.1 Problem Setting

### Domain

Let $\Omega \subset \mathbb{R}^2$ be a rectangular, heterogeneous bounded domain $x \times y$=$[0,\, B] \times [0,\, W]\, m^2$ with $B = 0.04\, m$ and $W = 0.14\, m$ (Figure 3.18a). Total time of observation is $(0, 400)\, s$. The correlation length is fixed to $l = 0.01\, m$, corresponding to $B/l = 4$ and $W/l = 14$. We consider a strongly heterogeneous field, in order to explore the effects of high level of complexity of velocity and concentration distribution on the grid adaptation strategy. Heterogeneity is quantified in terms of variance $\sigma^2$ of logarithmic permeability, which has been set equal to 5. The field has been generated on a uniform grid with 40 and 140 elements, respectively along $x$ and $y$ direction. Note this corresponds to characterize the conductivity field through 10 generation points per correlation length in the coarsest case, which we label as G1. The field is depicted on Figure 3.18a and we can immediately notice that for $y < 0.1\, m$, the domain is almost split symmetrically in a low (left-hand side) and high (right-hand side) permeability region, furthermore the horizontal section at $y \simeq 0.01\,m$, where the initial concentration has been fixed, corresponds to a stagnation zone, due to the low-velocity.

To design the reference mesh, to which space-time domain is discretized, we rely on a setting, characterized by a uniform numerical grid in space and a fixed time step across the simulation window. In this context, following [38], an appropriate discretization grid can be selected through a convergence analysis. Since this procedure has been rigorously employed in [19] and it is not a matter of interest for this thesis, we illustrate only the final results and we refer to [19] for further details. We note the convergence grid analysis has been performed only on *Fix*, but the reference grid will be used also in *Traces*.

First parameter to investigate is the time step. Coarse mesh G1 is taken into account and different simulations have been performed, assuming respectively
$\Delta t = 0.0625\,s$, $\Delta t = 0.125\,s, \Delta t = 0.25\,s$. We arbitrary state the section average breakthrough curve at the outlet border (Equation (3.4)), as criterion for the convergence. An upper bound of relative error of 5% among outlet breakthrough curve is fixed as accuracy target. In particular, we define:

$$e_{\Delta t_1} = \left| \frac{c_{\Delta t_1} - c_{\Delta t_2}}{c_{\Delta t_2}} \right| \tag{3.1}$$

as the relative error among two curves obtained with different and successively time step.

We assume that $c_{\Delta t_1}$ corresponds to the solution obtained with a lager time step, whereas $c_{\Delta t_2}$ with a lower. If relative error was lower than 5%, the maximum $\Delta t$ should be the convergent one. From Figure 3.1, it is possible to note that the constraint is fulfilled by $\Delta t = 0.0125\,s$.

Since the convergence time step has been selected, we now determine the grid size. Three different mesh refinement levels, labelled as G1, G4, G8, are taken into account. Referring to Table 3.1, the difference in the cardinality is related to the number of generation points per correlation length. Section averaged concentration at the northern border is still appointed as discriminant variable to identify the convergent grid.

| mesh | x elements | y elements | total number of triangles |
|---|---|---|---|
| G1 | 40 | 140 | 11200 |
| G4 | 160 | 560 | 179200 |
| G8 | 320 | 1120 | 716800 |

**Table 3.1:** Mesh levels for grid convergence analysis

We focus on Figure 3.2, where the blue curve is assumed to be the reference solution, hypothetically found with an infinite number of elements (see [38] and [19] for further details). It is subjected to uncertainties, which are represented as error bars, whereupon they give an idea of all probable and acceptable values. Since all points of red curve lay in those accuracy ranges, grid G6 (403200 elements) can be identified as the convergent mesh. It is the reference grid either for *Fix* and *Traces.*

Regarding the adaptive case, calibration of parameters is performed in [19], since it is not matter of this thesis, we simply note down final values:

- $\Delta t_{min} = 0.125\, s$;
- $\Delta t_{max} = 30\, s$;
- $N_{min} = 10^3$;
- $N_{max} = 10^4$;
- $\tau_h = 1$;
- $\tau_t = 10^{-4}$;
- $h_{max} = 0.06$;
- $h_{min} = 10^{-12}$;
- $p_{min} = 10^{-9}$;

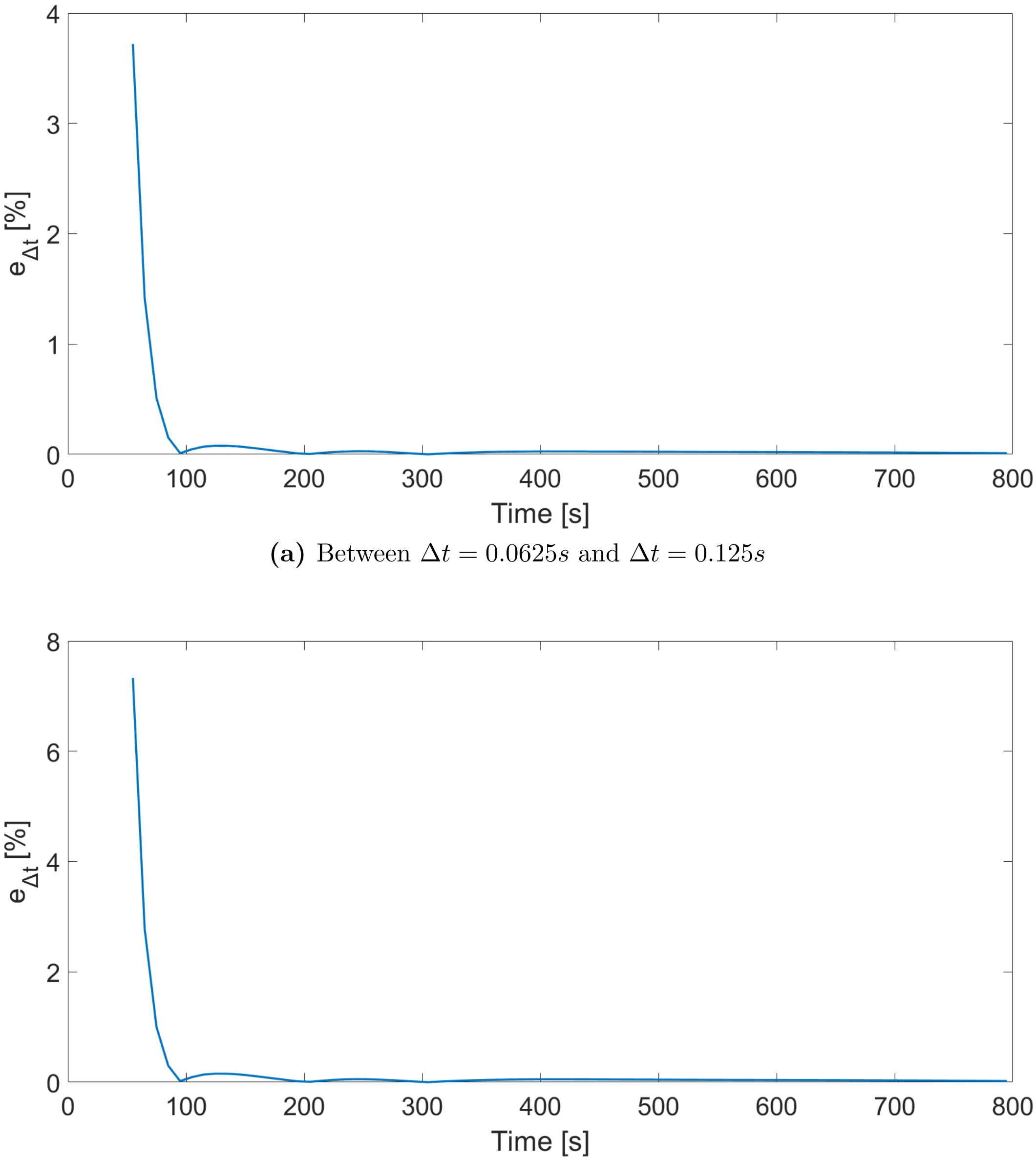
See Section 1.3 for further details.

### Flow and Transport

Velocity field is solved by means of Darcy's law and continuity equation, enlisted in Problem (1.14). A constant flux $q_0$ is imposed on the northern border, while a constant pressure $P_0$ is fixed at the southern wall.
The following data have been taken into account:

- Viscosity at 296 $°K$: $\mu$= 0.932 $Pa\, s$;
- Water density: $\rho$= 1000 $kg/m^3$;
- Gravitational acceleration: $g = 9.81\, m/s^2$;
- Porosity: $\varphi$= 0.35;
- Inlet pressure: $P_0 = 10000\, Pa$;
- Outlet Darcy's flux: $q_0 = 2.45 \cdot 10^{-4}\, m/s$.

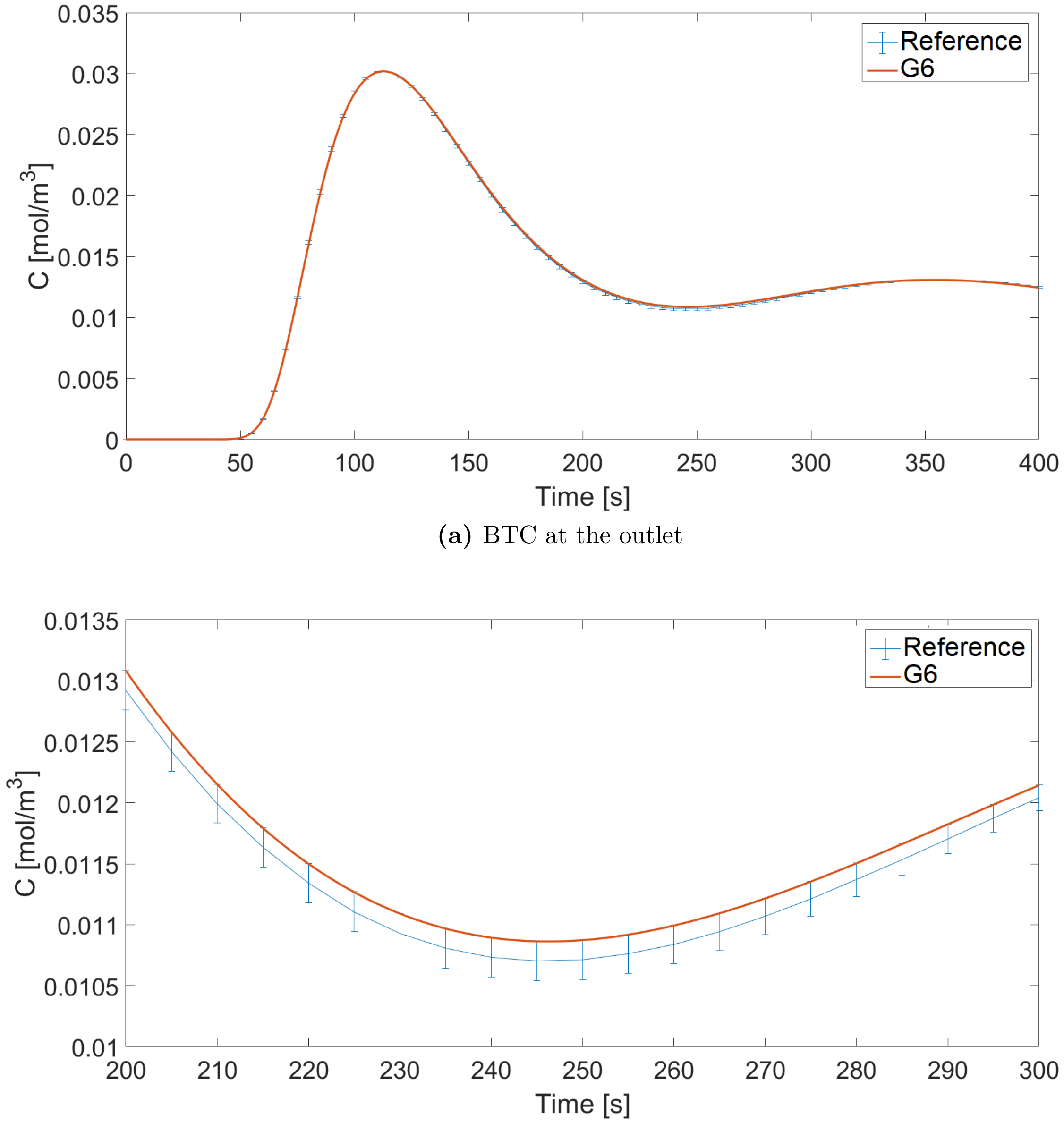
The discrete form of the flow problem is presented in Section 1.2.1. Logarithmic map of the velocity is depicted on Figure 3.18b and the complexity of the structure is evidenced by the presence of a clearly low-velocity region (approximately [0, 0.02] × [0.04, 0.06]) and a preferential channel along the eastern border. This concept is enforced by the streamlines in Figure 3.18c, where a dot represents the position of a particle at time $t$.



**(a)** Between $\Delta t = 0.0625s$ and $\Delta t = 0.125s$

**(b)** Between $\Delta t = 0.125s$ and $\Delta t = 0.25s$

**Figure 3.1:** Relative error trend at different time steps



**(a)** BTC at the outlet

**(b)** Zoom

**Figure 3.2:** RE and relative error bars and solution for mesh size of 403200 elements

Along the western border, the density of dots is much higher, because of low-velocity region, which delayed the flow.

For what concerns transport problem, dispersivity values have been set to
$\alpha_L = 10^{-3}\,m$ and $\alpha_T = 10^{-3}\,m$. We also need to add boundary conditions to the ADE (1.20). No Dirichlet imposition has been fixed, while assumption of impermeable wall is valid for western and eastern boundary
$((-\mathbf{D}\nabla c) \cdot \mathbf{n} = 0))$.
An initial function of concentration $c_0 = c(\mathbf{x}, t = 0)$, depicted on Figure 3.3, is imposed on the domain and it follows a Gaussian distribution:

$$c_0 = c(\mathbf{x}, t = 0) = exp\left\{ -\frac{(y - 0.01)^2}{0.00001} \right\} \tag{3.2}$$

Complete description of weak formulation and discretization is furnished in Appendix B.1 and Section 1.2.2.

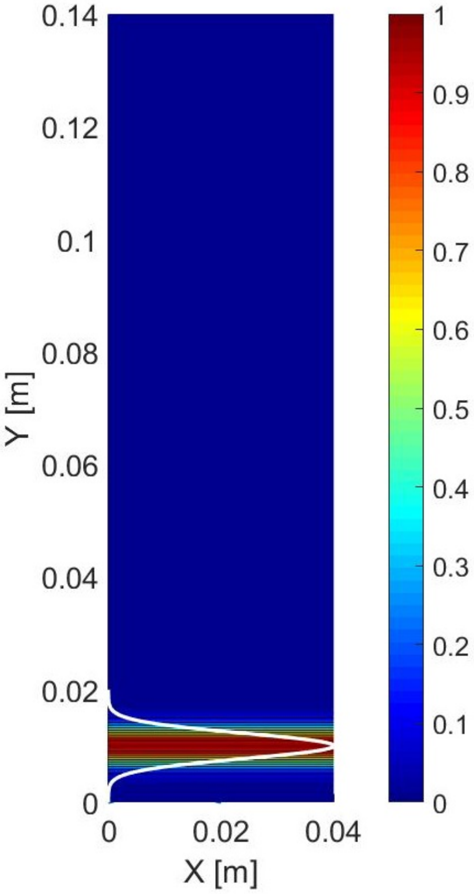


**Figure 3.3:** Spatial map of initial concentration. White curve represents Gaussian distribution of Equation (3.2)

## 3.2 Outputs of interest

We introduce here the quantities which constitute the key target outputs for the purpose of this work.
Temporal variation of solute concentration at given locations within the domain:

$$c(\mathbf{x}_i, t) = c(x_i, y_i, t) \tag{3.3}$$

where $\boldsymbol{x_i}$ identifies the points of investigation.
In particular we select $P_1(0.01, 0.04)$, $P_2(0, 0.08)$, $P_3(0, 0.12)$, $P_4(0.04, 0.04)$, $P_5(0.04, 0.08)$, $P_6(0.02, 0.12)$ (see Figure 3.18c). First three points lay in the slow-region (left-hand side part of the control volume), while the others in the fast. The objective is to figure out any changes in the solution, due to different distribution of advective field.
We also considered section-averaged concentration, mimicking typically observed breakthrough curve (BTC):

$$\overline{c_{out}} = \frac{1}{|\Gamma_{out}|} \int_{\Gamma_{out}} c(\mathbf{x}, t) d\Gamma \quad \forall t \in [0, 400]\, s \tag{3.4}$$

where the subscript *out* corresponds to the northern boundary.
We then focus on globally integrated quantities, which can quantify spreading and mixing of the plume within the domain. To this end, we consider second centred spatial moment of the concentration plume along $y$ direction, which has relevant role for the characterization of solute spreading. It is defined as:

$$S(t) = \frac{1}{M(t)} \int_{\Omega} (y - Y_{av}(t))^2 c(\mathbf{x}, t)\, d\Omega \tag{3.5}$$

where:

$$Y_{av} = \frac{1}{M(t)} \int_{\Omega} c(\mathbf{x}, t) y\, d\Omega \tag{3.6}$$

and

$$M(t) = \int_{\Omega} c(\mathbf{x}, t)\, d\Omega \tag{3.7}$$

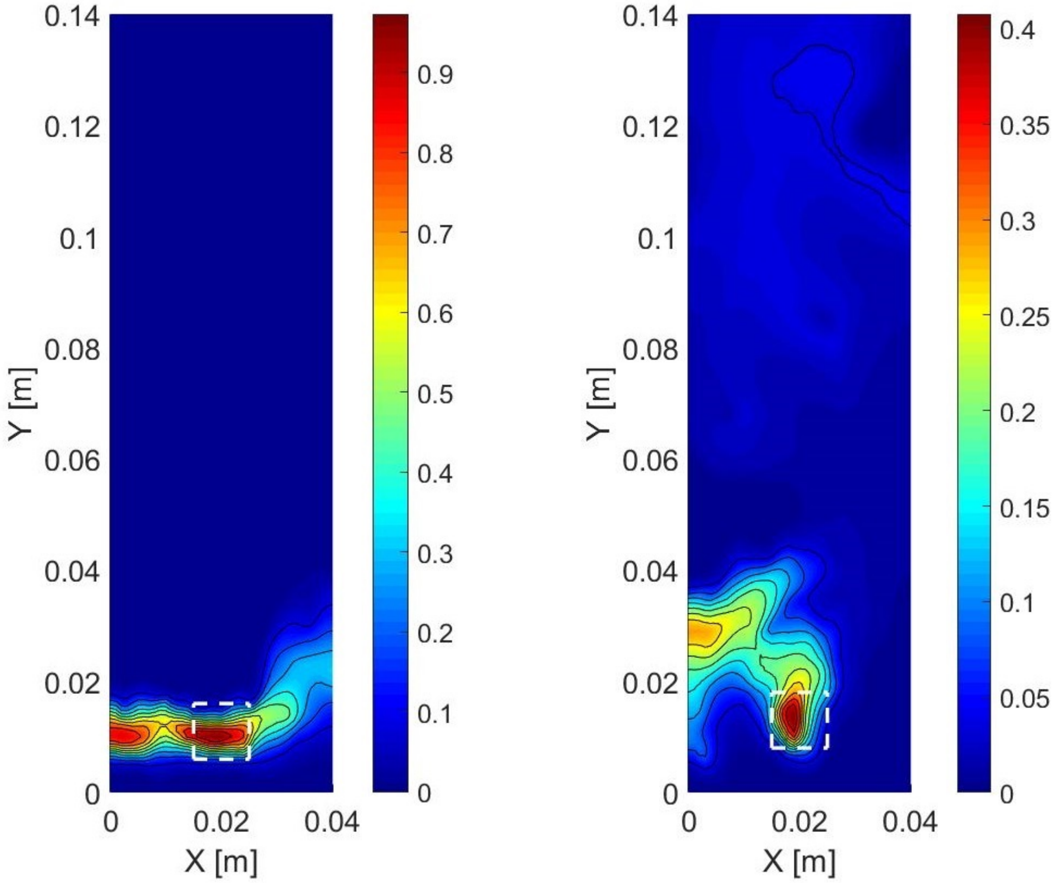
On the other hand, scalar dissipation rate:

$$\chi(t) = \int_{\Omega} \nabla c^T \mathbf{D} \nabla c\, d\Omega \tag{3.8}$$

which quantifies the rate of mixing of plume.

## 3.3 Results

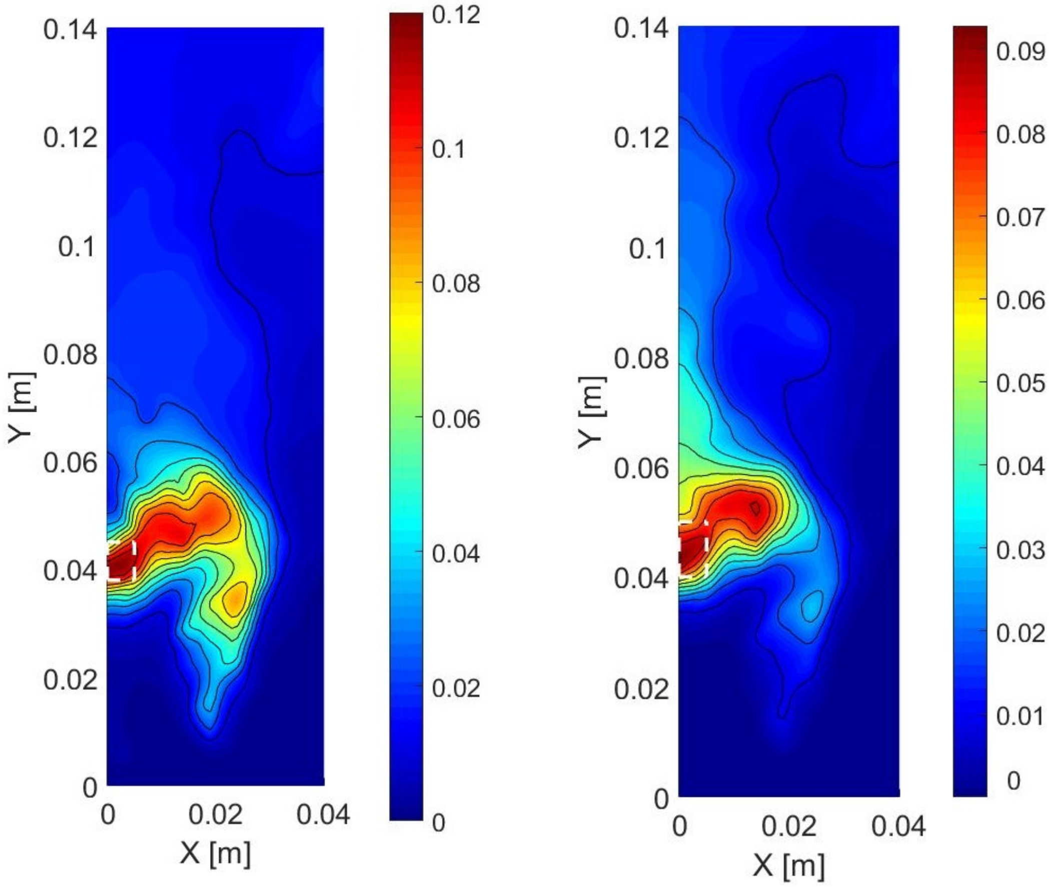
We here illustrate the comparison of numerical results, associated with the observables, described in Section 3.2 and obtained, relaying on *Fix, Traces* and *Adaptive.*
We start the analysis from an overview of the solute transport, considering some snapshots of the concentration evolution, depicted on Figure 3.4. During the first time instants, solute spreads more intensively, towards the eastern border, while in the left-hand side region it remains less diffuse (Figure 3.4a).



**(a)** Time t=5 s

**(b)** Time t=100 s

**Figure 3.4:** Spatial distribution of concentration at $t = 5\,s$ and $t = 100\,s$, *Fix* solution. Dashed white square encloses low velocity region.



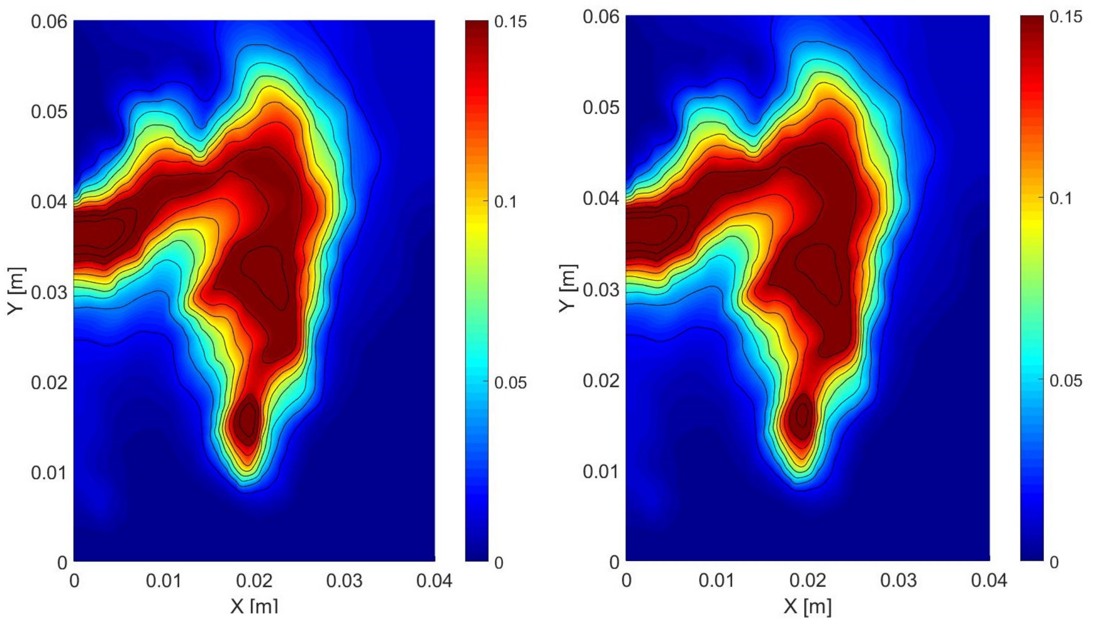
**(a)** Time t=300 s

**(b)** Time t=400 s

**Figure 3.5:** Spatial distribution of concentration at $t = 300\,s$ and $t = 400\,s$, *Fix* solution. Dashed white square encloses low velocity region.

Along time, the initial Gaussian distribution (Figure 3.3) starts being deformed and for $t > 50s$ mass flows out the northern border (Figure 3.13a). Referring to Figure 3.18a, it is possible to observe that part of the solute mass is delayed, due to the presence of two low-velocity bottom regions (represented as white dashed square), one around $x = 0.02\,m$ and $y = 0.01\,m$ and the other around $x = 0\,m$ and $y = 0.05\,m$. The former is much effective at the beginning of the simulation (Figure 3.4), because the initial condition is still the dominant aspect in the concentration distribution, whereas, once the solute tends to spread upwards, it runs into the second slow zone (Figure 3.5).

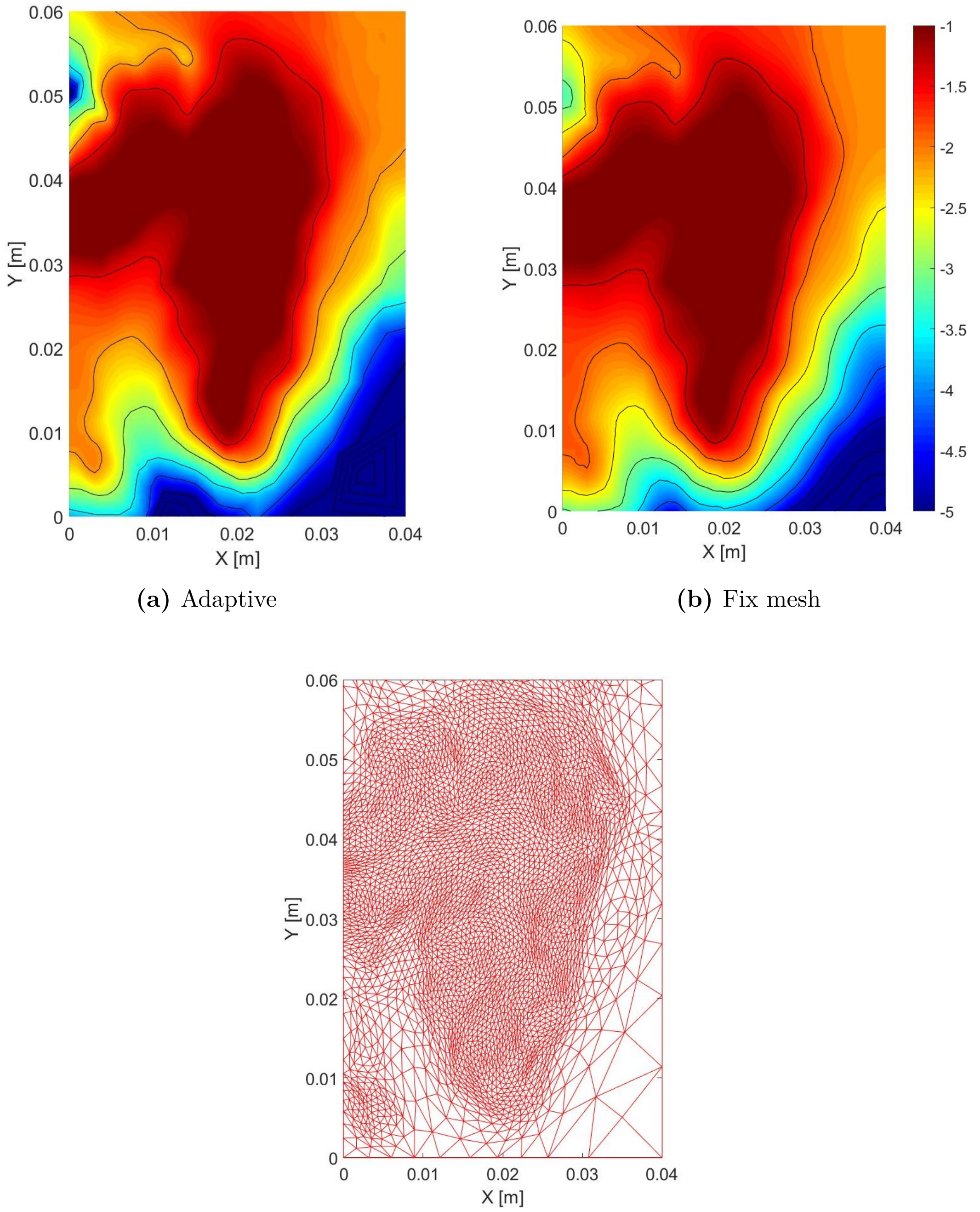
The comparison of concentration map between *Fix* and *Traces* do not detect any particular differences, in fact, focusing on Figure 3.6, they exhibit the same shape of contour lines and values in the concentration field. Therefore, we focus on *Adaptive* and, in order to provide a rigorous analysis, for time $t = 200s$, concentration field is presented in logarithmic scale, otherwise small values are critical to evaluate. Results are plotted in Figure 3.7, they are very similar, however some oscillations (of the order $< 10^{-5}$) appear in the solution (bottom-right region of the control volume). The emergence of these oscillations is probably linked to the interpolation of the concentration, in a region, where mesh elements are large. However, for sake of accuracy, the order of magnitude is very low, whereupon it could be neglected.



**(a)** *Fix* solution

**(b)** *Traces* solution

**Figure 3.6:** Spatial distribution of concentration at t=200 s

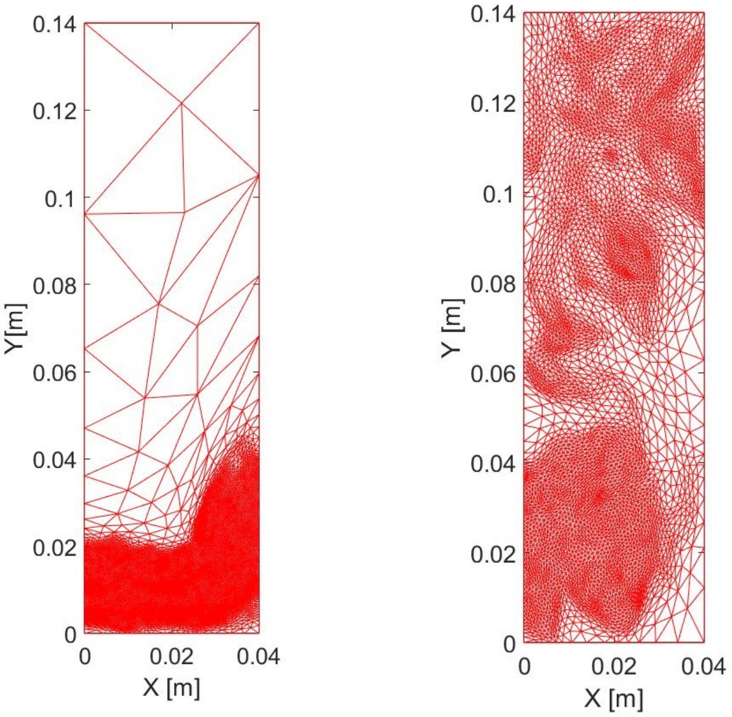


**(a)** Adaptive

**(b)** Fix mesh

**(c)** Grid

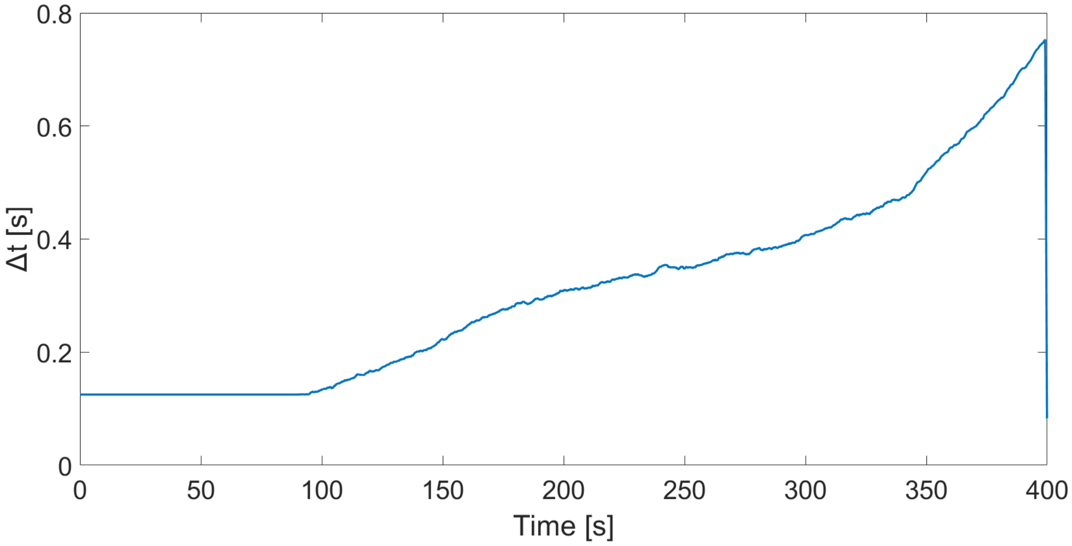
**Figure 3.7:** Spatial log-distribution of concentration with associated grid, at time $t = 200\,s$



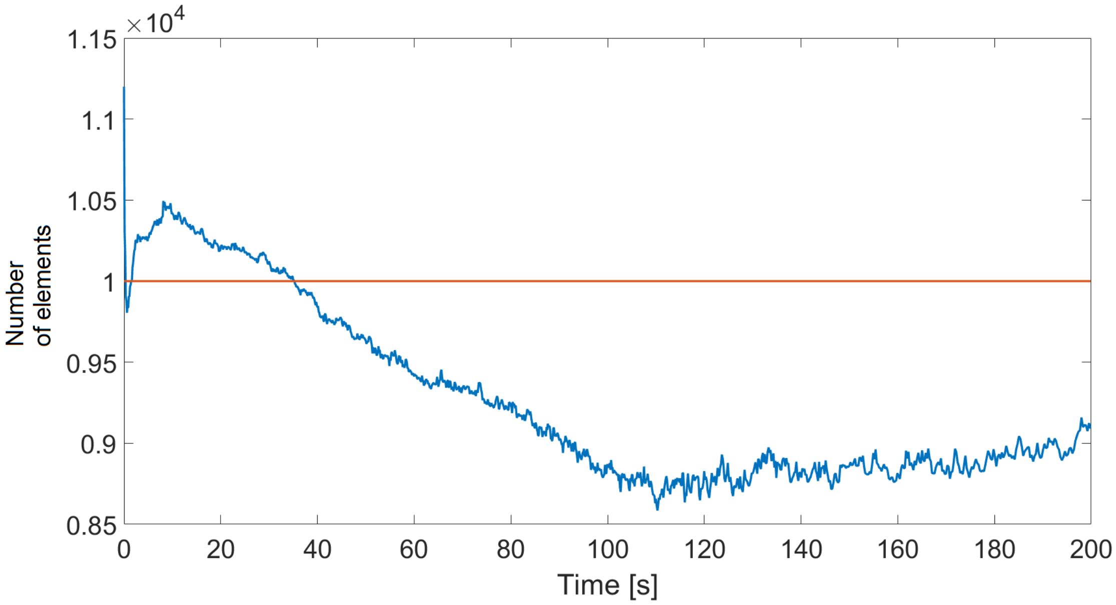
**(a)** Adaptive grid at time $t = 5s$ **(b)** Adaptive grid at time $t = 100s$

**Figure 3.8:** Adaptive mesh topology

Figure 3.9 shows the trend of time step and number of elements against time, whose evolution could be split in two phases. In the first instants, $t < 100\,s$, time step is constant, it coincides with $\Delta t_{min}$, because of the rapid temporal variation of the concentration field. As time advances, values of $\Delta t$ larger than $\Delta t_{min}$ are allowed.

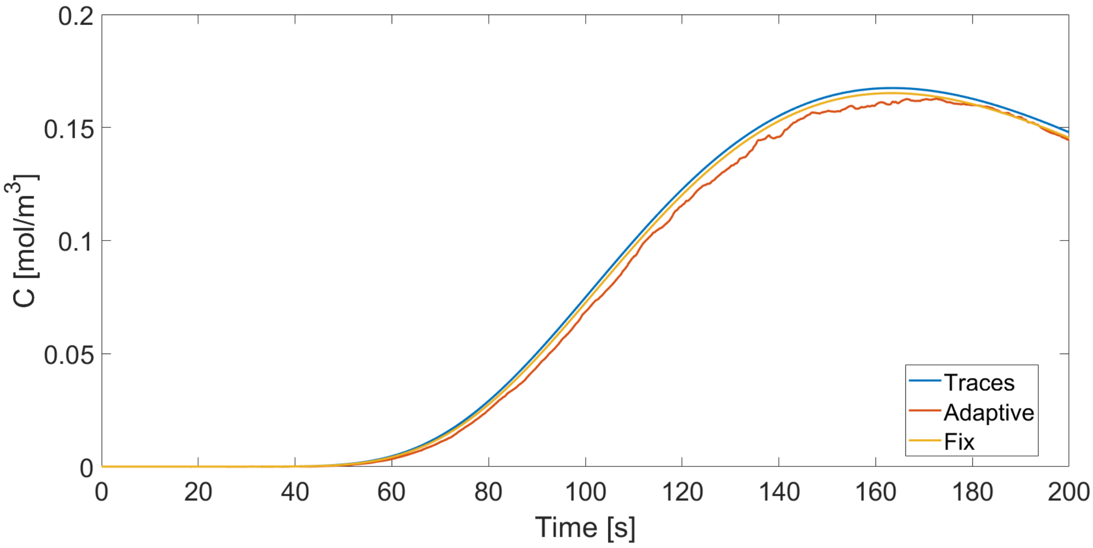


**(a)** Time step

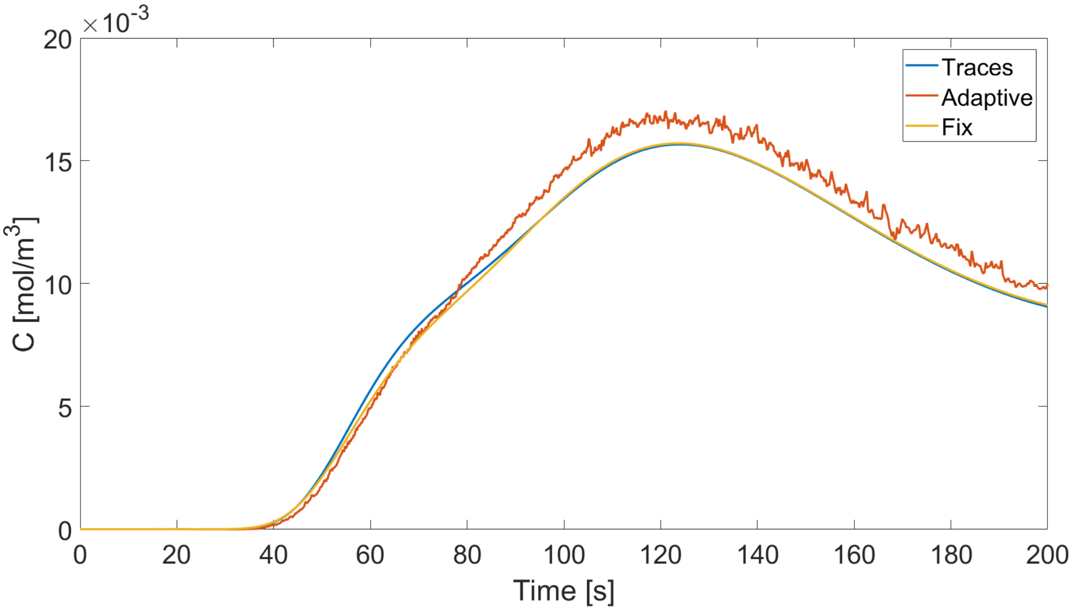


**(b)** Mesh elements

**Figure 3.9:** Mesh elements and time step as function of time

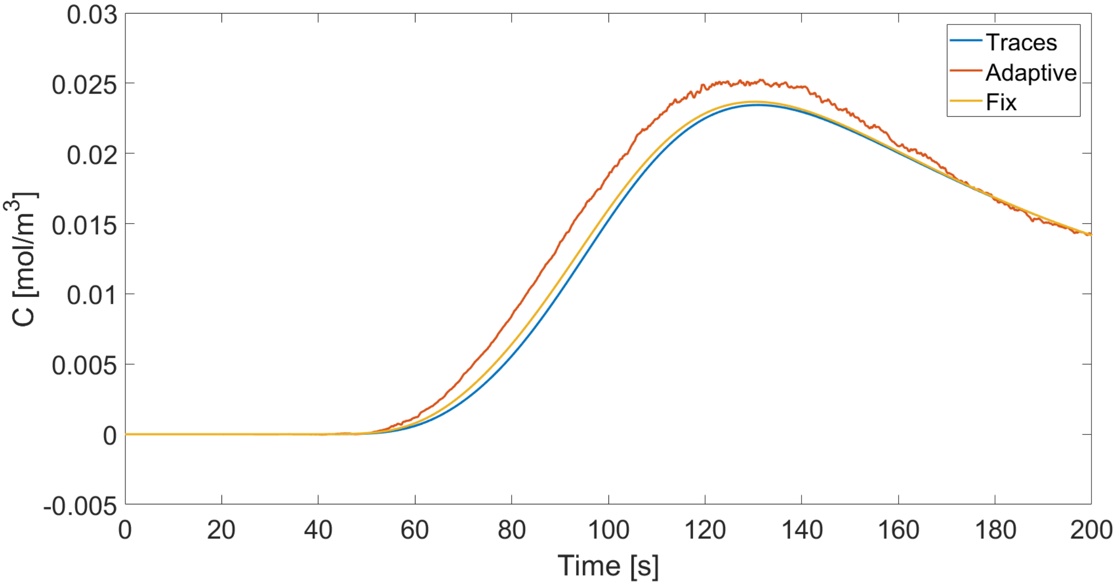


**(a)** $P_1(0.01, 0.04)$

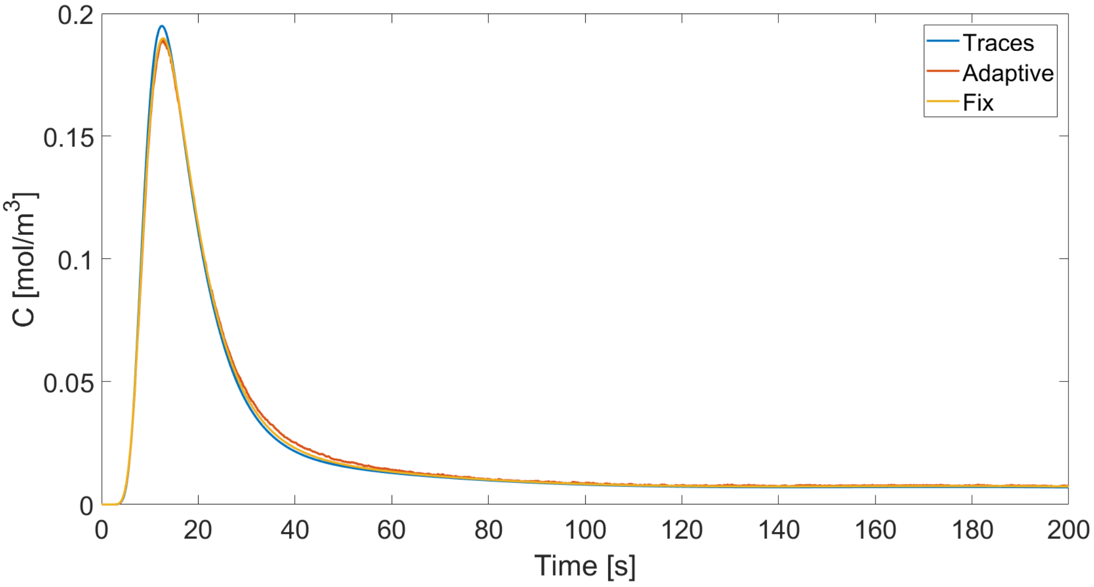


**(b)** $P_2(0, 0.08)$

**Figure 3.10:** Local concentration evolution, points in the slow velocity region

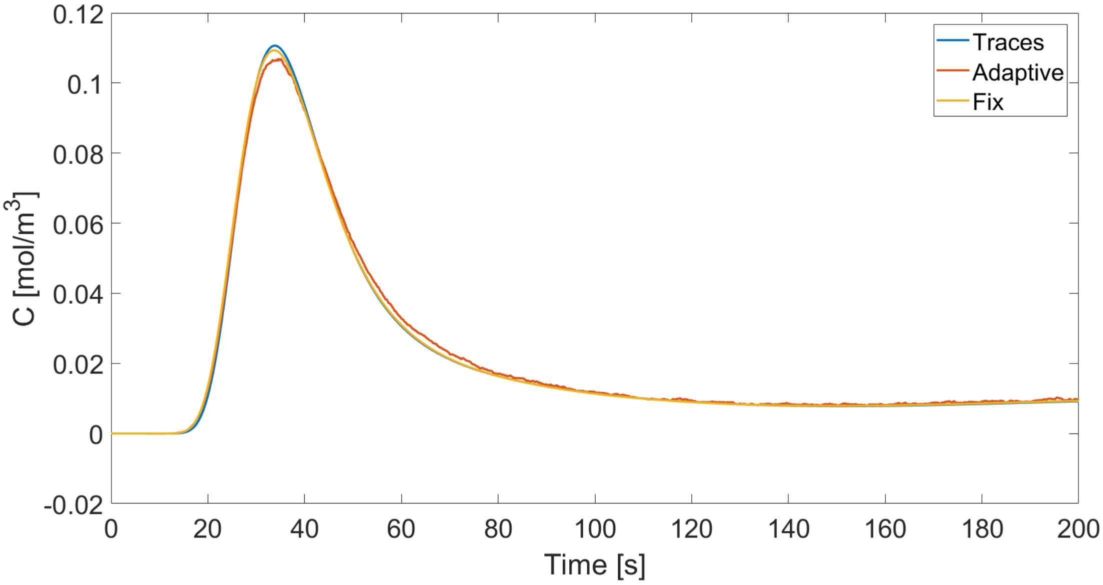


**(a)** $P_3(0, 0.12)$

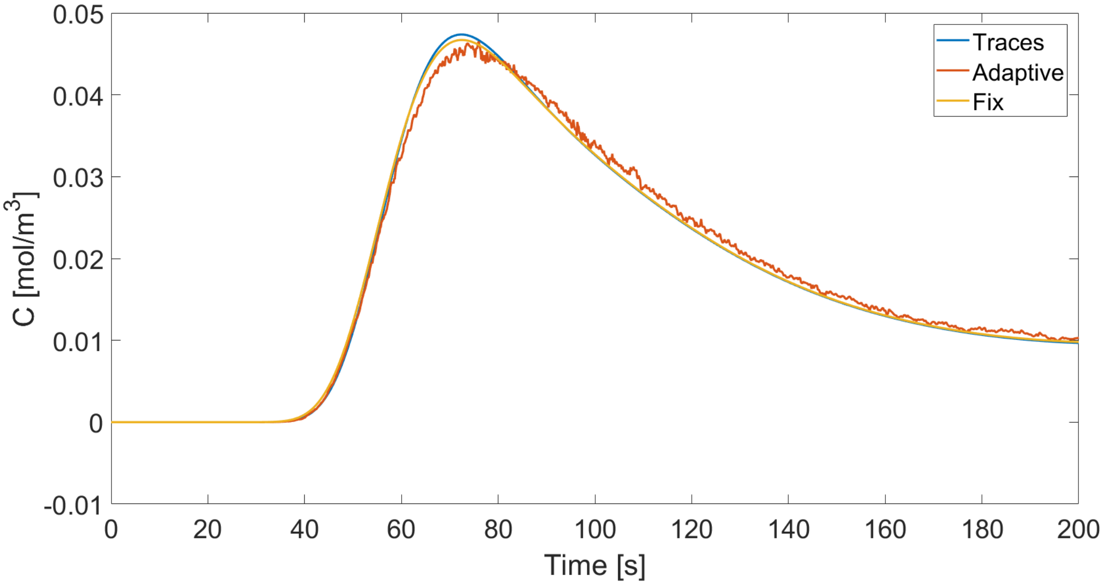


**(b)** $P_4(0.04, 0.04)$

**Figure 3.11:** Local concentration evolution. Top: point in the low-velocity region; bottom: point in the high-velocity region

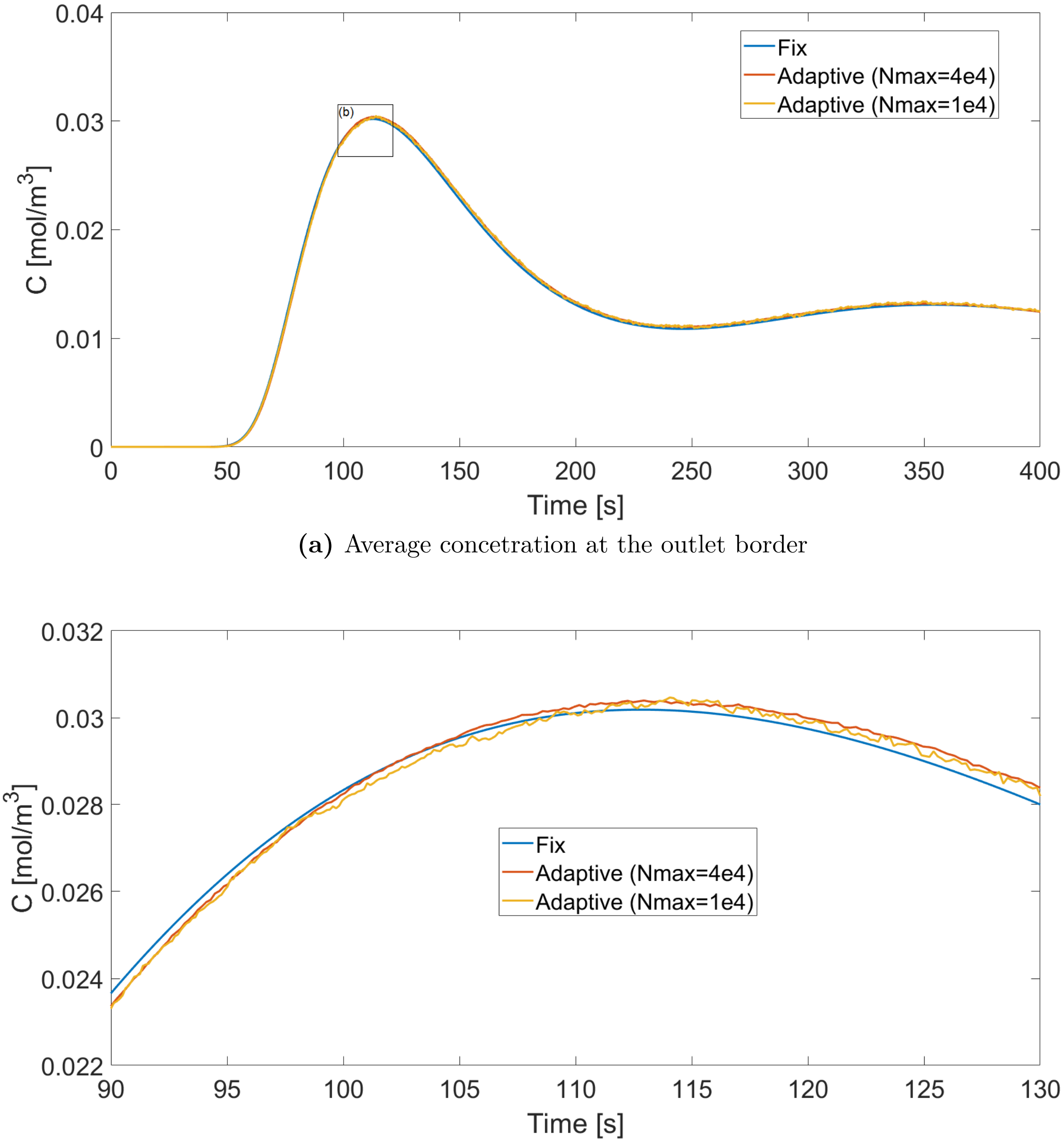


**(a)** $P_5(0.04,\, 0.08)$



**(b)** $P_6(0.02,\, 0.12)$

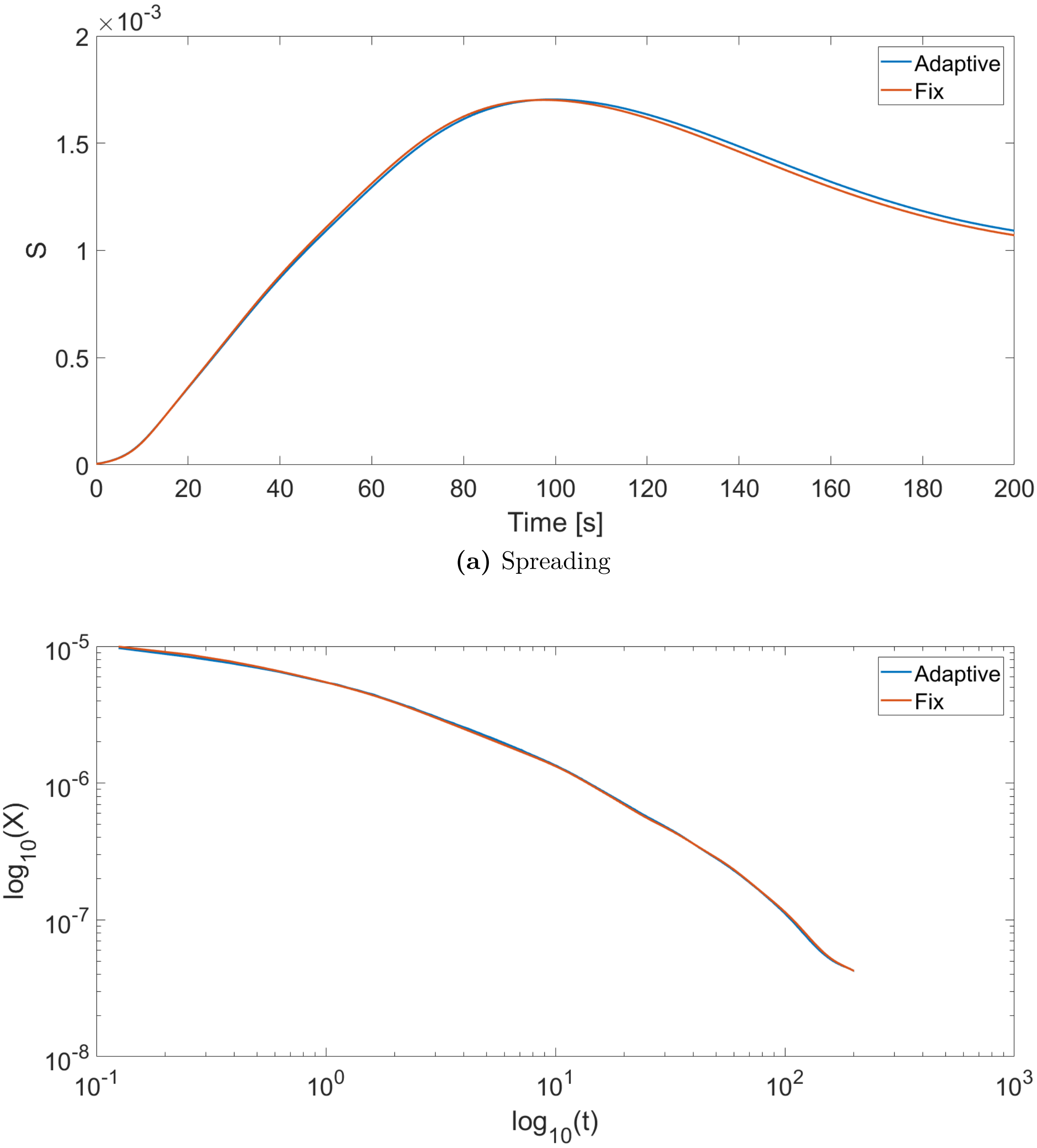
**Figure 3.12:** Local concentration evolution, points in the fast velocity region



**(a)** Average concetration at the outlet border

**(b)** Zoom at the peak

**Figure 3.13:** Section averaged concentration evolution, *Fix* and *Adaptive*



**(a)** Spreading

**(b)** Mixing

**Figure 3.14:** Spreading and Mixing comparison, fix and adaptive solutions

This is because the solute plume spreads over an increased portion of the domain and diffusive/dispersive process gain importance, leading to a reduced time variation. On the other hand, in terms of mesh cardinality, we note a significant decline, because of the constraint on the maximum number of elements. For $t > 100\,s$ the mesh size reaches a sort of equilibrium, in fact it maintains a flat behaviour. Time step increases linearly, approximately with a constant slope, until $t = 350\,s$, then, in the last part of simulation, the trend increases even more.

From Figures 3.10 to 3.12, we depict concentration evolution of points, enlisted in Section 3.2. *Fix* and *Traces* give the same results, while there is a mild difference in the trend of *Adaptive*, especially in Figures 3.10b and 3.12b, which are marked out by intense oscillations.
Asymmetry is a recurring feature of all curves and this behaviour is linked to the level of heterogeneity of the conductive field, on the contrary the structure of Darcy's map largely influences the solute mass distribution either from a physical and numerical point of view. In the fast-velocity region, peak time (instant at which BTC reaches its peak value) is very low and the transport is convective dominated. Instead, in the slow-velocity region solute mass is mainly exchanged by diffusive/dispersive mechanism, which causes a delayed peak time and a long release of solute mass. These considerations could also be interpreted from a numerical point of view. We assume to carry out a sensitivity analysis, by comparing order of magnitude of convective and diffusive fluxes, including also the divergence term.
From an analytical point of view $c\nabla \cdot \mathbf{u} = 0$, but numerically, especially during velocity projection, this condition is not guaranteed and it could affect accuracy of the results. However in the high-velocity regions, the advection is the dominant term and Expression (3.9) holds:

$$\mathbf{u} \cdot \nabla c \gg c\nabla \cdot \mathbf{u} + \nabla \cdot (\mathbf{D}\nabla c) \tag{3.9}$$

Taking into account that a rapid peak time also implies a larger cardinality in the mesh (see Figure 3.9b), this leads to a smooth trend in Figures 3.11b and 3.12b. Numerical issues are limited by the physical process involved. On the contrary, in slow-velocity region, Expression (3.9) does not hold, left and right terms might have the same order of magnitude, it means the error in the fluid mass conservation is not negligible and it causes oscillations, such as in the case of Figure 3.10b.

It is evident, that *Traces* and *Fix* generate the same results, either from a local and global point of view, therefore there is no sense to provide a comparison among codes. Instead we take into account different strategy, indeed fixed space-time discretization, which is now the reference solution, versus adaptive method.
The main objective is now to detect weaknesses of the latter, assuming *Fix* as reference solution.

In Figure 3.13a, averaged concentration $\overline{c_{out}}$ is computed through Equation (3.4). At first sight, the general trend is well respected and the adaptive solution follows the fix behaviour. However, zooming on the peak, Figure 3.13b, it is possible to recognize the low level of accuracy, in fact oscillations are pronounced and they don't let to obtain a smooth trend. Raising $N_{max} = 4e4$, the quality is improved.

Spreading and mixing behaviours (Equations (3.5) and (3.8)) are plotted in Figure 3.14. About the former, as time increases also $S(t)$ does, since the solute initially spreads around its center of mass. On the other hand, when the trend starts to fall down, it means the solute has already exited the domain. No particular differences are detected between the adaptive and fix solution.

From the outputs evaluated, it is possible to infer some important considerations, which involves in particular the adaptive strategy.
Solute behaviour is generally well described, however a shortage in the accuracy and a remarkable presence of oscillations, especially for BTC, have been detected. A possible way to limit them is to raise the maximum number of elements, with an inevitable increase of the computational cost, therefore our target is to investigate the reasons of this behaviour to find another remedy. Following [5], a crucial issue in adaptive setting regards the matter of conservation for solute and fluid mass, hence these quantities are checked.
According to Section 1.4, Equation (1.48) is applied to project concentration map from one mesh to another. In order to verify the consistency of this procedure, we provide a global solute mass balance for *Adaptive*. This leads to calculate outcoming and incoming mass flow. We label as $M_0$ the initial mass content, as $M_{in}$ the mass inside the domain, whereas $M_{out}$ denotes the solute, which left the control volume, across the northern boundary. Since we neglect any type of source term inside the domain, the following expression holds:

$$M_0 = M_{in}(t) + M_{out}(t) \quad [mol] \tag{3.10}$$

where:

$$M_0 = \int_\Omega c(\mathbf{x}, t=0)\, d\Omega \tag{3.11}$$

We label as $M^*(t)$ the relative solute mass inside the domain $\Omega$ at time $t$:

$$M^*(t) = \frac{M(t)}{M_0} = \frac{1}{M_0}\int_\Omega c(\mathbf{x}, t) \quad \forall t \tag{3.12}$$

Regarding $M_{in}$ and $M_{out}$:

$$M_{in}(\hat{t}) = \int_\Omega c(\mathbf{x}, t=\hat{t})\, d\Omega \tag{3.13}$$

$$M_{out}(\hat{t}) = \int_0^{\hat{t}} M(\hat{t})\, dt = \int_0^{\hat{t}} \overline{c_{out}}\, u_{out}|\Gamma_{out}|\, dt = u_{out}|\Gamma_{out}| \int_0^{\hat{t}} \overline{c_{out}}\, dt \tag{3.14}$$

where:

- $\overline{c_{out}}$ is the average concentration at the outlet border;
- $u_{out}$ is the imposed velocity at the outlet border and it is constant;
- $|\Gamma_{out}|$ is the length of the outlet border.

Equation (3.14) could be evaluated in a discretized way and it reads:

$$M_{out}(T) = u_{out}|\Gamma_{out}| \sum_{t=0}^{\hat{t}} c_{out}\, \Delta t \tag{3.15}$$

Note that the term $\Delta t$ must be taken inside the summation, because it is not a constant. Table 3.2 shows the results, in particular, focusing on the column *err*, which represents the relative error, we could infer that the conservation is well respected.

| $M_{in}(t = 400s)$ | $M_{out}(t = 400s)$ | $M_0$ | err [%] |
|---|---|---|---|
| $7.55 \cdot 10^{-5}$ | $1.49 \cdot 10^{-4}$ | $2.24 \cdot 10^{-4}$ | 0.22% |

**Table 3.2:** Global mass balance for *Adaptive*

The global mass content remains constant, but this does not assure the local conservation. However, since solute transport depends on the advective field, we first handle with the local fluid mass conservation. According to Section 1.4, the stationary velocity field needs to be projected from one mesh to another, time by time. This is actually performed via *NCI* (non-conservative interpolation) scheme. The aim is to assess effects, related to the employment of this technique, and determine critical features.
We need to calculate local flux balance at each triangle element, therefore matrix *Raviart* ($N_t \times 6$), with $N_t$ number of triangle elements, is implemented and built with the structure in Table 3.3.

| triangle T | $q_{1,T}$ | $q_{2,T}$ | $q_{3,T}$ | $\sum_{E=1}^{3} q_{i,T}$ | $\log_{10} \sum$ |
|---|---|---|---|---|---|

**Table 3.3:** Structure of the matrix *Raviart*

The sixth column of the matrix gives information on the fluxes balance at local level, therefore a barycentric value for each triangle is provided. The latter has to be interpreted as the numerical approximation of the local divergence of velocity. Since the remeshing occurs at each time step, $N_{step}$ matrices have been created, where $N_{step}$ identifies the number of time steps. Figure 3.15 shows the logarithmic flux balance distribution in the fixed space domain, so Figure 3.16 does, with respect to the interpolated results, provided by *NCI* upon adaptive grid at time $t = 200\, s$.
Flux balance configuration in Figure 3.15 is consistent with the continuity principle, since the range of values in the log-scale is sufficiently low. On the contrary, for what concerns the results, provided by *NCI*, involving the adaptive grid in Figure 3.16, we note that:

- Mass conservation is not fulfilled, since the range in the logarithmic scale can not be approximated at zero;
- Error map follows the mesh structure, in fact the coarse (top) region shows the least accuracy, whereas the fine (bottom) zone shows acceptable results,

in terms of conservation of fluxes balance. However, this aspect is due to the small size of triangles, instead of a correct flux interpolation, so it is restricted to a particular case of the mesh topology. To understand it, we focus on the highlighted region in Figure 3.16a. It is composed by elements, whose surface area has almost the same order of magnitude of minimum allowed value $p_{min}$, hence each of those triangles are enclosed in a single element of the original flow grid, therefore only in this case *NCI* is acceptable.

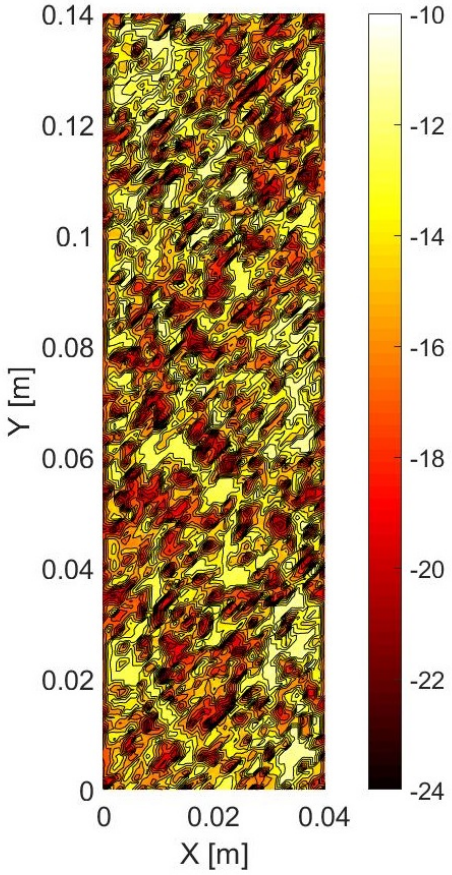
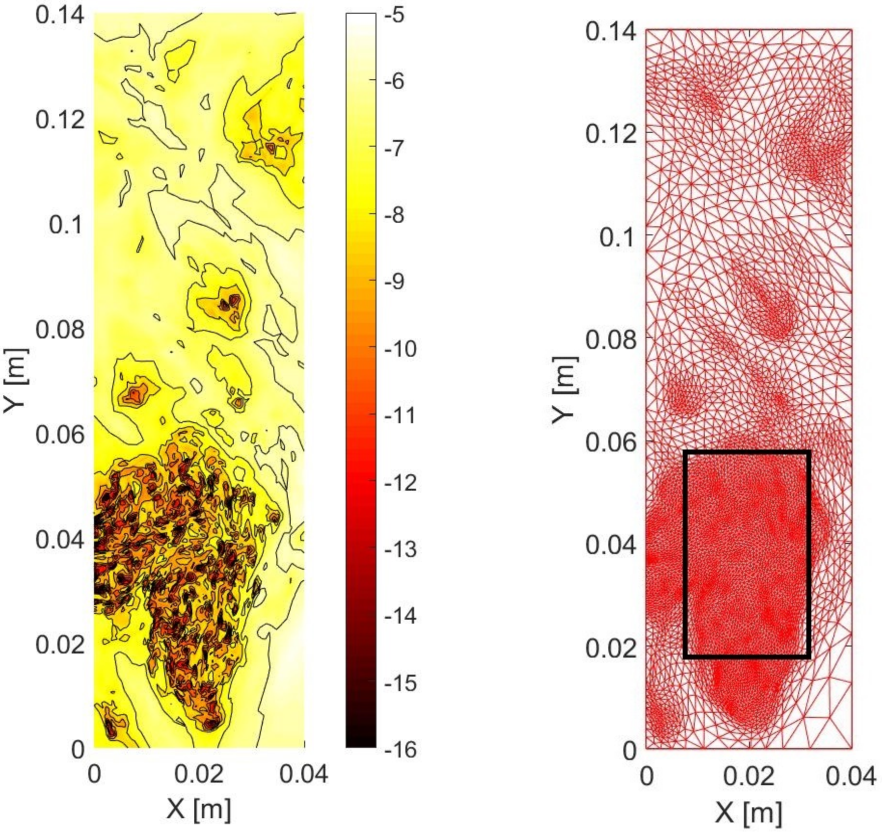


**Figure 3.15:** Flux balance map in logarithmic scale as direct numerical solution of Darcy's law



**(a)** Logarithmic flux balance map **(b)** Grid

**Figure 3.16:** Flux balance map and associated grid at time t=200 s

We have demonstrated that *NCI* does not ensure fluid mass conservation throughout the grid.

To figure out if the lack of fluid mass conservation leads to the oscillations, we perform a simulation, where the flux balance, computed through Equation (1.49), is checked at each time step on the elements, where points $P_2(0, 0.08)$, $P_4(0.04, 0.04)$ and $P_6(0.02, 0.12)$ lay (see Figure 3.18).

Two triangular sub-domains $\Omega_1$ and $\Omega_2$ have been created, such as in Figure 3.17. They have fixed boundaries, so, while remeshing takes place within the domain, vertices of these subregions are anchored. Step by step, fluxes balance is calculated, through Equation (1.49) on the elements, where points are located. The main advantage of this layout is to control the flux balance and guarantee a better approximation, because we are forcing adaptation procedure to limit the coarsening of the desired region, where points lay.

$P_2 \in \Omega_1$, $P_6 \in \Omega_2$ and $P_4$ only belongs to the common domain $\Omega$.

We remind this is not a method to correct numerical issues of *NCI*, but it is a simple way to have a perception of how the error in the fluid mass conservation affects the concentration evolution.

*NCI* is always the non-conservative interpolation scheme, which is actually performed during the adaptation procedure. We label as *New* the results just obtained, with the introduction of two sub-domains $\Omega_1$ and $\Omega_2$.

The analysis consists in observing changes in the local BTC and average value of log-flux balance (Table 3.4), computed in the time window $[0, 200]\,s$. Flux balance has been calculated on the triangle, where the point lays. Every step the element changes due to adaptation. From the transport simulation, we obtain $N_{step}$ values of local mass balance in the given triangles, with $N_{step}$ number of time step, however we calculate the arithmetic average to consider a single value.

We focus on Figure 3.19a, where the fluctuation is considerably reduced, the trend is much smoother and it suits better to the reference behaviour, depicted as solution of *Fix* . This improvement accrues from the enhancement in the fluid balance, which directly affects the concentration evolution at the given point. On the other hand, in Figure 3.19b, oscillations are still present, but this is consistent with the restrained gap between old and new average flux balance value. Figure 3.20 has been included in the analysis, in order to ensure that sub-domain do not modify the whole field. Despite a mild change in the average flux conservation, caused by the different mesh topology along time, *NCI* and *New* concentration evolution for point $P_4$ are practically the same.

| | **NCI** | **New** |
|---|---|---|
| $P_2$ | -6.3 | -15.9 |
| $P_4$ | -7.3 | -6.9 |
| $P_6$ | -6 | -6.8 |

**Table 3.4:** Average log-flux balance value between *NCI* and *New*

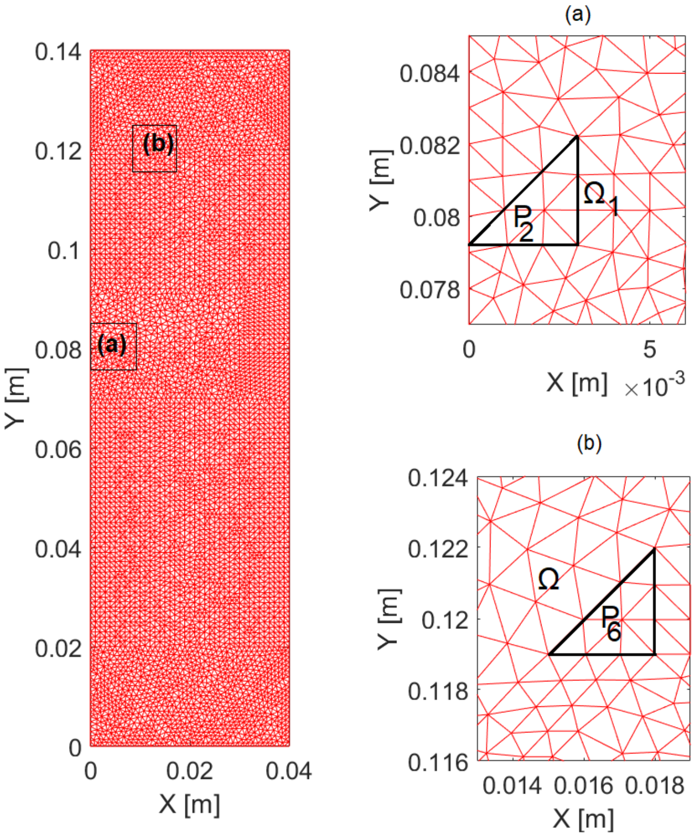
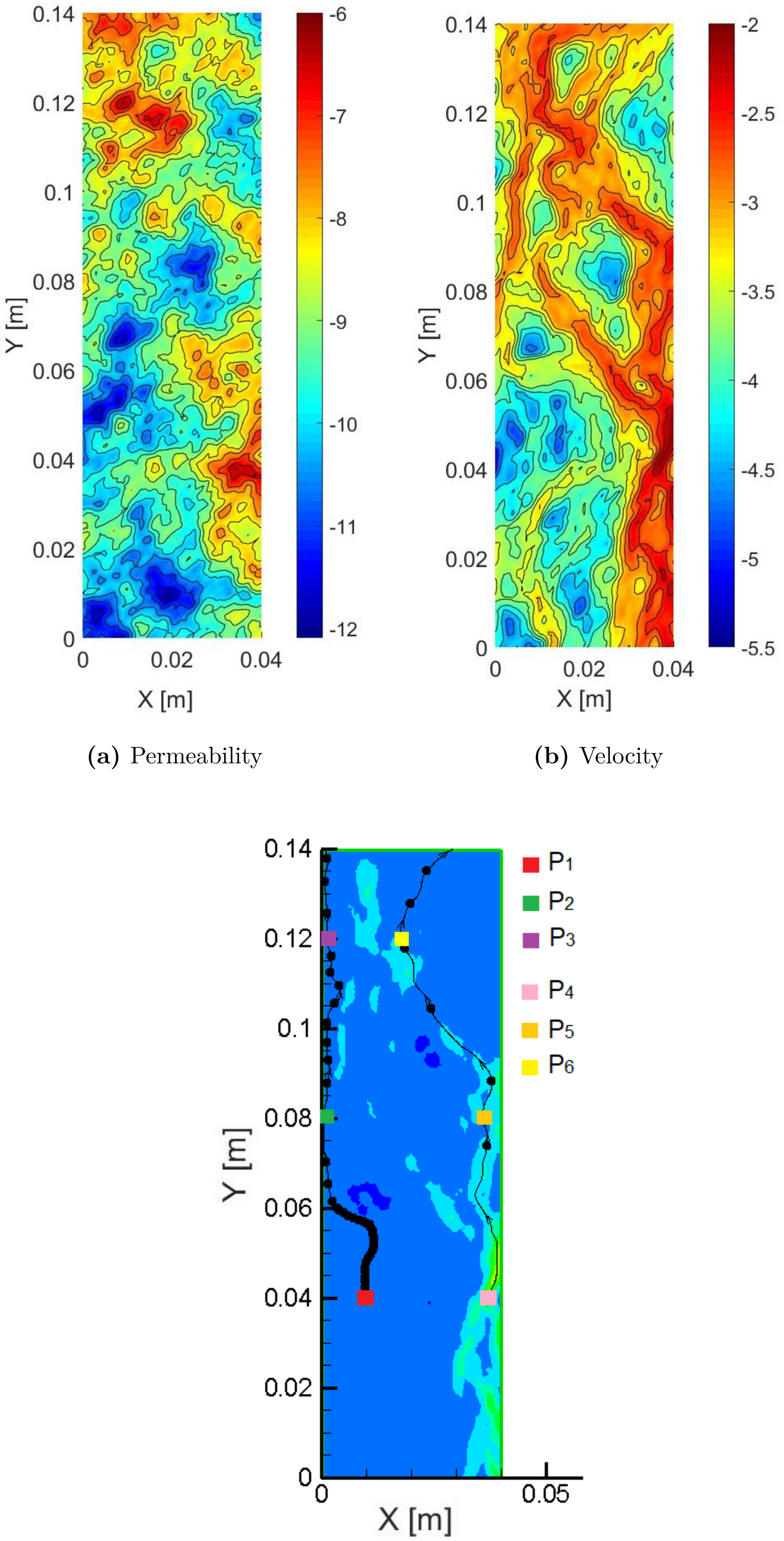


**Figure 3.17:** Internal sub-domains $\Omega_1$ and $\Omega_2$

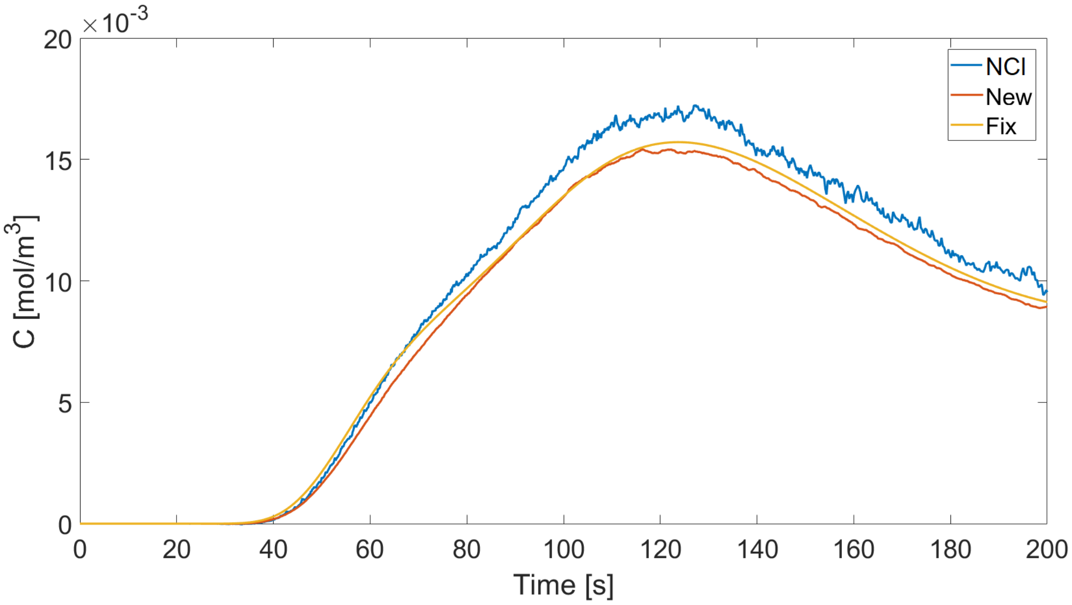


**(a)** Permeability

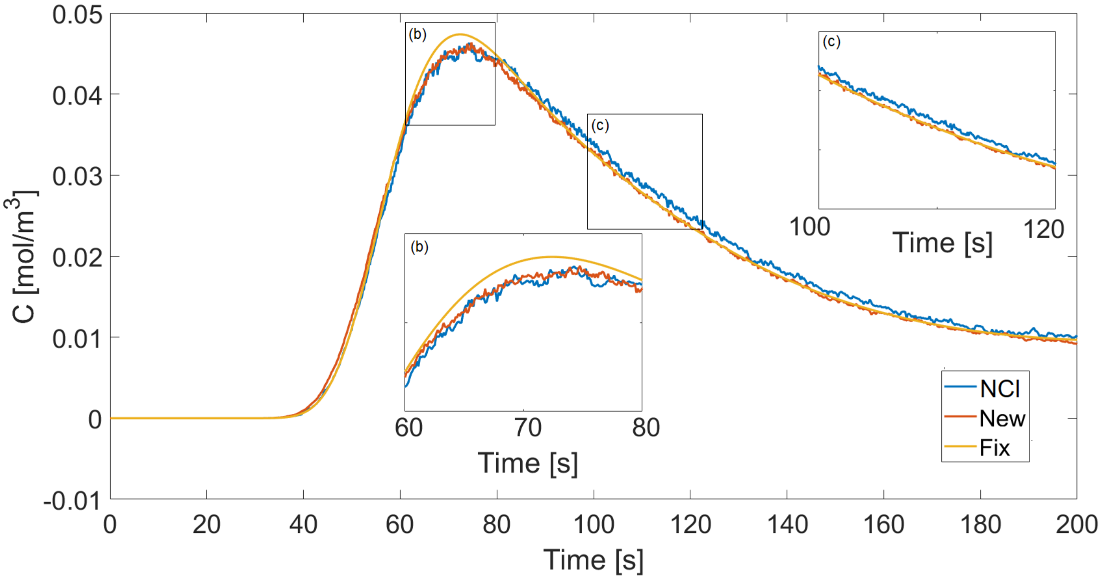
**(b)** Velocity

**(c)** Streamlines

**Figure 3.18:** Top: logarithmic field of permeability and velocity. Bottom: streamlines representation



**(a)** $P_2(0, 0.08)$



**(b)** $P_6(0.02, 0.12)$

**Figure 3.19:** Local concentration evolution. Top: point in $\Omega_1$; bottom: point in $\Omega_2$

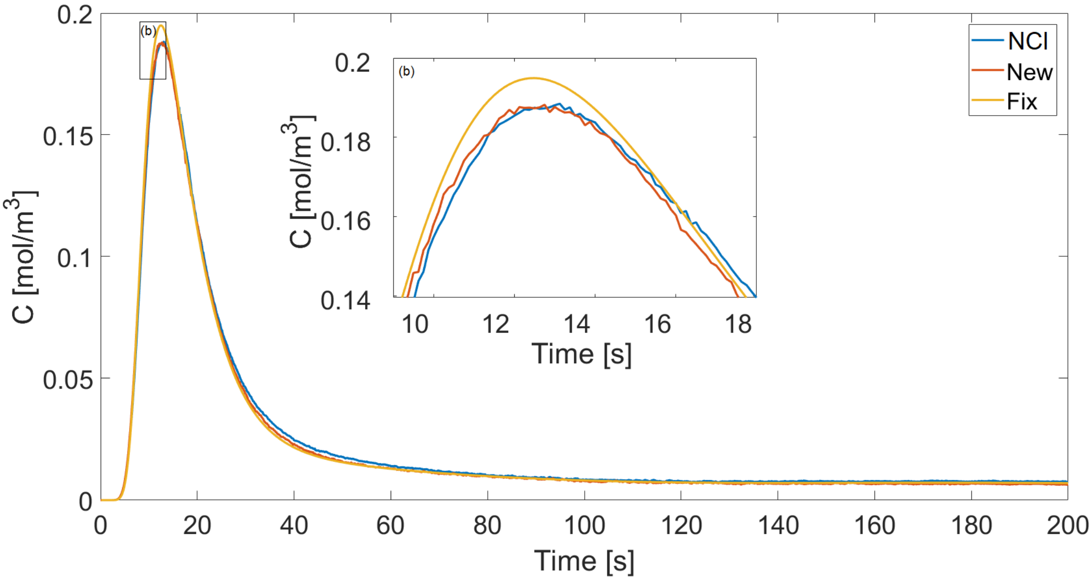


**Figure 3.20:** Local concentration evolution for $P_4(0.04, 0.04)$

# Chapter 4

# Flux Interpolation

The employment of the adaptive setting forces us to project variables from one grid to another. According to Section 1.4, the matter of conservation is a fundamental theoretical aspect to abide by, but it assumes a deep meaning also from a numerical point of view. In order to avoid artificial mass generation, flux balance must hold at each time step, however results and simulations performed in Chapter 3 suggest that this condition is not fulfilled. They also reflect a lack of accuracy in the concentration evolution, mainly due to the wrong interpolation method, intrinsically yielded with a *Freefem* subroutine during the adaptation procedure.
The aim of this Chapter consists in implementing a conservative interpolation algorithm, labelled as *CI* (conservative interpolation) and examining it, with respect to the existent version of *Freefem*, labelled as *NCI* (non-conservative interpolation). A conservative interpolation scheme is tailored to the highly heterogeneous system and anisotropic mesh topology.
The work is organized as follows:

Section 4.1 introduces the new algorithm *CI* with a deep focus on the main steps.

Section 4.2 presents the results, in terms of velocity and flux balance map, obtained through the new algorithm *CI*. Two mesh refinement strategies have been taken into account.

## 4.1 Algorithm

To distinguish the notation and simplify the reading of this section, we remind the nomenclature:

- *NCI* indicates the non-conservative interpolation scheme, explained in Section 1.4, which is implemented as a *Freefem* subroutine and has generated the oscillating solutions seen in Chapter 3;

- *CI* represents the conservative interpolation method, which has been introduced, in order to overcome the limits of the actual formulation.

We denote as *flow grid*, the mesh where Darcy's problem has been solved, and as *transport grid*, the adaptive mesh, generated along the simulation. According to

Figure 4.1, we state to use the following notation:
Flow grid:

- $\mathcal{T}_h^0$: flow grid;
- $T$: generic triangle in $\mathcal{T}_h^0$;
- $E_i$: $i^{th}$ edge of triangle $T$ in $\mathcal{T}_h^0$;
- $B_i$: $i^{th}$ vertex of the triangle $T$ in $\mathcal{T}_h^0$;
- $R_i$: intersection point;
- $S_i = (\overline{R_i - R_{i+1}})$: segment, composed by two consecutive intersection points;
- $N_c$: number of segments $S_i$;
- $M_i = (x_M, y_M)$: middle point of $S_i$;
- $q_{i,j}$= flux associated to segment $S_i$, which composed the edge $L_j$ (see transport notation);
- $Q_i$: stationary flux, computed through Darcy's law, associated with $E_i$.

Transport adaptive grid:

- $\mathcal{T}_h^n$: transport adaptive grid;
- $K$: generic triangle in $\mathcal{T}_h^n$;
- $V_i$: $i^{th}$ vertex of the triangle $K$;
- $L_i$: $i^{th}$ edge of triangle $K$ in $\mathcal{T}_h^n$;
- $N_i = (x_N, y_N)$: middle point of $L_i$;
- $q_i$: unknown flux, associated with $L_i$.

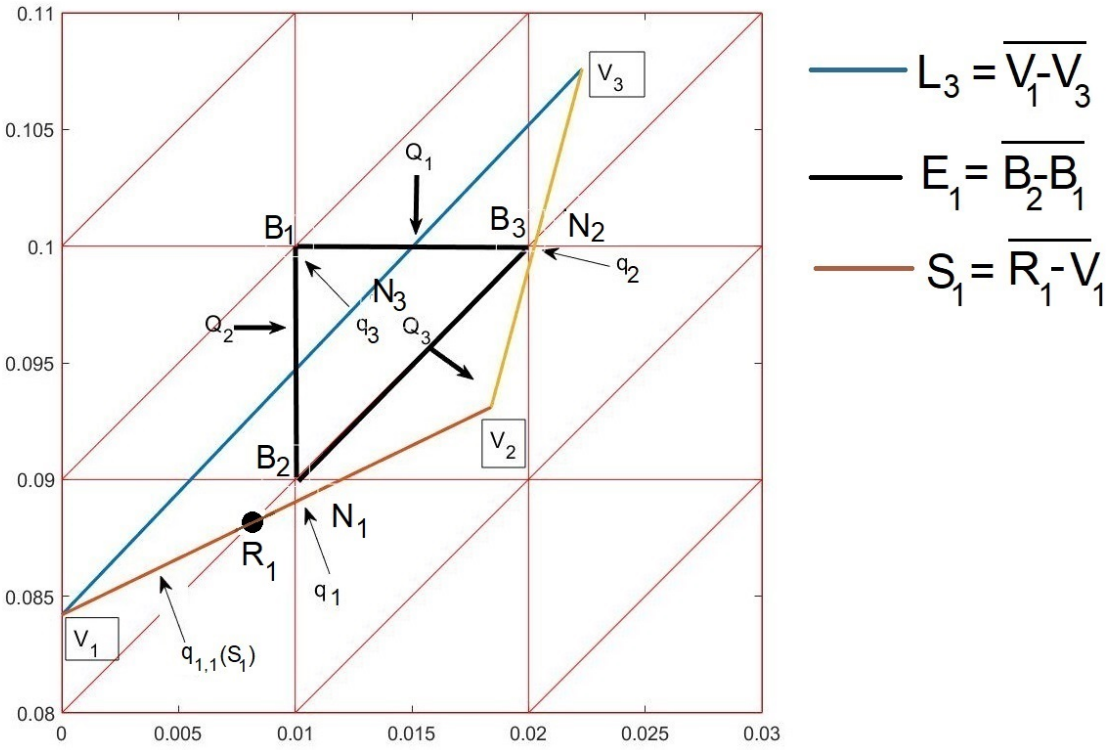


**Figure 4.1:** Sketch of meshes projection

The *CI* algorithm has been implemented in *Freefem* and it reads as follows:

**ALGORITHM**

1. **Input data** :
   they encompass connectivity matrices of flow grid $\mathcal{T}_h^0$ and fluxes $Q_i$. In particular we need to determine the association triangle-vertices, triangle-edges, edge-vertices and edge-triangles, in order to completely characterize the mesh topology.

2. **Read mesh** :
   it consists in reading new adaptive mesh $\mathcal{T}_h^n$ and extrapolate connectivity matrix triangle-vertices;

3. **Locate triangle K** :
   it consists in pinpointing vertices of triangle $K$ in the flow grid $\mathcal{T}_h^0$, indeed in which triangle $T$ a vertex lays;

4. **Calculate intersection** :
   it calculates intersection point $R_i$ of edge $L_i$ with triangle elements in $\mathcal{T}_h^0$. We determine the position of $R_i$ (both triangle $T$ and edge $E_i$) in the flow grid. Once the edge $E_i$ is found, from the connectivity matrix is possible to shift from the actual triangle to the next one (we remind a single edge is shared at most with two triangles). This is a loop which allows to move along the whole edge and detect all the intersections. Moreover, once two consecutive points $R_i$ and $R_{i+1}$ have been found, it is possible to calculate the length of the segment $S_i$ and its middle point $M_i = (x_M, y_M)$;

5. **Interpolate velocity** :
   it provides a conservative interpolation for each segment $S_i$, which composes the edge $L_i$. The computation is different from the *NCI* method, previously explained, in fact, to ensure fluid conservation in the triangle $T$, we impose to calculate the velocity at the middle point, as result of fluxes balance, otherwise numerical errors would affect the value. As consequence, it could be interpreted as a direct estimation and not an interpolation, such as described in Section 4.1;

6. **Sum all the contributions** :
   all partial fluxes, associated with the segments $S_i$, which compose the edge $L_i$ are summed up, to determine the whole flux $q_i$.

The *CI* algorithm is governed by two main loops, the first rules the reading of triangles $K$, which belongs to adapted mesh, in order to change at each step the vertices to find and so the edges, while the second has been employed to calculate all the intersections with the flow grid.
In order to deeply figure out the process developed, steps 3-4-5-6 are now rigorously detailed.

### Locate triangle K:

The first aspect to take into account is the full localization of element $K$, through the position of its vertices $V_i$. According to Figure 4.2, the procedure is based on the determination of the square area $A_i$, generated by $V_i$ with $B_i$ (set of vertices of generic triangle $T$). Therefore:

- $A_1$: area composed by vertices $B_1 - B_2 - V_i$;
- $A_2$: area composed by vertices $B_2 - B_3 - V_i$;
- $A_3$: area composed by vertices $B_3 - B_1 - V_i$;
- $A_{tot}$: area composed by vertices $B_3 - B_1 - B_2$;

Adopting this strategy, if the following expression holds:

$$V_i \in T \quad \text{if } A_{tot} \simeq \sum_{i=1}^{3} A_i \tag{4.1}$$

position of $V_i$ is uniquely defined. In the case $V_i$ coincides with a vertex $B_i$, all triangles sharing it satisfy the Constraint (4.1), therefore the code identifies the first acceptable element and then the position is adjusted, based on the slope of edge $L_i = \overline{V_i - V_j}$.

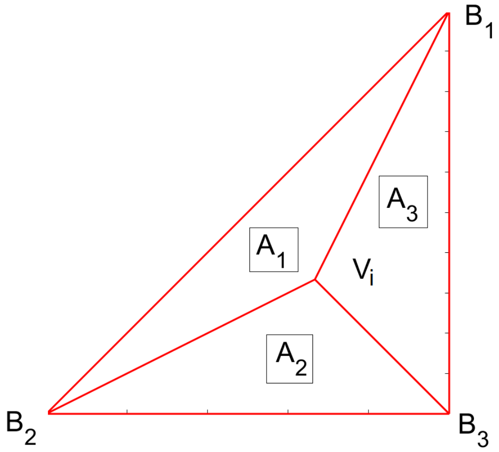


**Figure 4.2:** Determination of the vertex position in the flow grid

### Calculate intersections:

According to Figure 4.3, we remind that $B_i$ corresponds to the $i^{th}$ vertex in the flow grid (it is marked with a square), $R_i$ represents the $i^{th}$ intersection point (it is marked with a circle), $T_i$ identifies the $i^{th}$ triangle of the flow grid (it is enclosed in a box), $V_i$ is the $i^{th}$ vertex of the triangle in the transport grid, $L_i$ is the edge of $K$ with vertices $V_i$ and $V_j$.

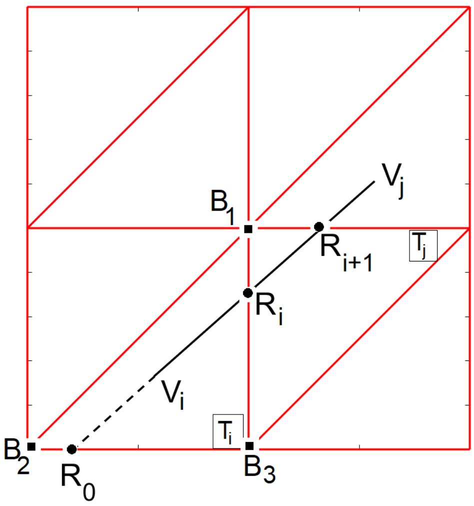


**Figure 4.3:** Calculation of the intersection

Since all vertices $V_i$ have been located in the flow grid, it is possible to calculate intersections, occurring between $L_i$ and the edges $E_i$ of triangle $T_i$. As already described, this step has been implemented, through a conditional loop, which allows to move along the edge $L_i$, starting from $V_i$ and stopping at the triangle of the second vertex $V_j$. The actions performed are:

1. Calculation of all the possible intersections with the lines generated by points $B_i \in T_i$;
2. Adoption of a constraint to limit the solution to the given triangle $T_i$;
3. Adoption of a constraint to limit the solution in a range of coordinates determined by the maximum and minimum between the second vertex $V_j$ and the previous intersection point, which is automatically updated at each step. For example in the determination of $R_i$, the range of acceptability is composed by point $[V_i - V_j]$, while to calculate $R_{i+1}$, the constraint is shifted to $[R_i - V_j]$ and so on.
   This strategy allows to remove solutions which do not lay on the given edge $L_i$, such as in the case for point $R_0$;
4. Calculation of the length $|S_i|$ and middle point $M_i = (x_M, y_M)$ of the segment $S_i$, composed by two consecutive intersection points. According to Figure 4.3, $M_i$ would be the middle point of $S_i = \overline{V_i - R_i}$;
5. Determination of the adjacent triangle $T_j$, which shares the edge $E_i$. According to Figure 4.3, $R_i$ is the intersection point between $L_i = \overline{V_i - V_j}$ and $E_3 = \overline{B_3 - B_1}$, in particular the latter is shared between $T_i$, the initial triangle, and $T_j$, the next element to consider.

This procedure is implemented for all vertices $V_i$, in order to determine all middle point $M_i$ and length $|S_i|$, because are fundamental parameters to estimate the flux, associated to segment $S_i$.

**Interpolate velocity:**

This is the core of the code *CI*, because it provides a conservative technique to interpolate the velocity field on the adaptive grid and makes the flux balance constraint fulfilled. The process is split in two phases, first we determine the pointwise velocity $\boldsymbol{u_M} = (u_1(x_M, y_M),\, u_2(x_M, y_M))$ and then the flux $q_{i,j}$.
Once coordinates of point $M_i$ are known, the estimation of the correct velocity needs to guarantee flux balance on the two triangles, created by the vertical and horizontal lines held from $M_i$ itself. According to Figure 4.4 and following the definition of flux, this leads to:

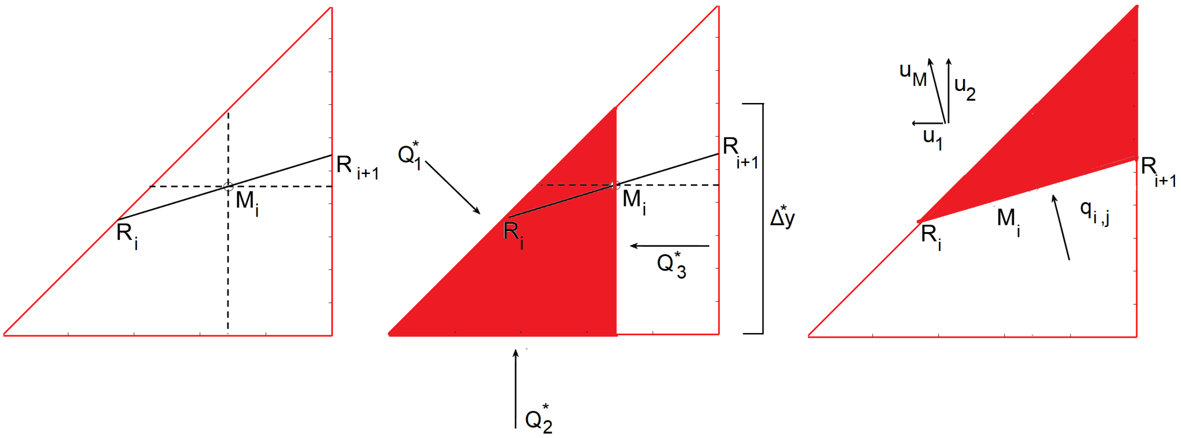


**Figure 4.4:** Velocity field interpolation and flux calculation

$$Q_3^* = -(Q_1^* + Q_2^*) = u_1(x_M, y_M)\,\Delta y^* \tag{4.2}$$

where $u_1(x_M, y_M)$ is the velocity component along $x$ direction, $Q_i^*$ is proportional to the flux $Q_i$.
In Equation (4.2), $u_1(x_M, y_M)$ is the only unknown. According to Figure 4.4, analogously it is possible to determine $u_2(x_M, y_M)$, which is calculating from the balance in the upper sub-triangle, formed by the horizontal line, held from $M_i$. This is an important consideration, because the linear interpolation would not offer accurate results. According to Section 1.4, velocity vector is known at the middle point of each edge in the flow grid. A plane could be constructed:

$$z = ax + by + c \tag{4.3}$$

where coefficient $a, b, c$ are found from the resolution of the system:

$$\begin{cases} z_1 = ax_1 + by_1 + c \\ z_2 = ax_2 + by_2 + c \\ z_3 = ax_3 + by_3 + c \end{cases} \tag{4.4}$$

where $(x_i, y_i)$ are the coordinates of middle point of the edge in the flow grid, while $z_i$ is the velocity ($u_1$ and $u_2$) in the considered point. Substituting in Equation (4.3), $(x_M, y_M)$ the resulting $u_M(x_M, y_M)$ is slightly different from the velocity obtained with the previous method. There is almost a gap of 5% per element, among two

schemes, which could affect a lot the interpolation technique, since the numerical error is propagated for each triangle considered.
Following the conservative approach, we denote as $q_{i,j}$ the flux associated to the segment $S_i = \overline{R_i - R_{i+1}}$, which composes the edge $L_j$.
We remind the definition of both edge and normal vector, respectively $\boldsymbol{L_j}$ and $\boldsymbol{n_{Lj}}$:

$$\boldsymbol{L_j} = \begin{bmatrix} cos(\beta) \\ sen(\beta) \end{bmatrix} \quad \boldsymbol{n_{Lj}} = \begin{bmatrix} sen(\beta) \\ -cos(\beta) \end{bmatrix}$$

where $\beta$ is the angle formed by the segment $L_j$ with the horizontal.
According to Figure 4.4, flux $q_{i,j}$ is computed through the common formula:

$$q_{i,j} = \int_{Si} \mathbf{u} \cdot \mathbf{n} \, dS_i \tag{4.5}$$

The final step is the summation of all contributions $q_{i,j}$, which compose the edge $L_j$. Since each term $q_{i,j}$ is locally conservative on the triangle $T_i$ of the flow grid, the flux balance for triangle $K$ holds:

$$q_j = \sum_{i=1}^{N_c} q_{i,j} \longrightarrow \nabla \cdot \boldsymbol{u_h}|_K = \sum_{j=1}^{3} q_j|_K \simeq 0 \tag{4.6}$$

in fact it comes from a summation of conservative terms, where $N_c$ is the number of segments $S_i$.
Furthermore, the main novelty of this scheme is not only to abide by continuity constraint, but also to account for the heterogeneity of the porous medium. Flux $q_{i,j}$, associated to each segment $S_i$ is characterized by a different permeability value, consistent with the flow grid. Instead *NCI* only uses the middle point of the edge.

## 4.2 Results

Let us now focus on the presentation of numerical results, obtained through the new *CI* algorithm. Control volume and heterogeneity characterization of the system are identical to those considered in Chapter 3, whereas we initially present a simple case of mesh refinement, which does not use the adaptive remeshing explained in Section 1.3.
The aim is to verify the correctness of the code developed, by comparing the numerical solution of Darcy's law with the interpolated values. Both are estimated on the same grid.

### 4.2.1 Code Verification

The first numerical test consists in the verification of the code, assuming a very simple mesh refinement. Avoiding the adaptive procedure, a regular grid, labelled as initial flow grid (Figure 4.6a), is considered, a set of investigation points is defined a priori, their location is detected and then the considered elements are divided in three, connecting each vertex with the corresponding baricenter, leading to the generation of a new grid, labelled as transport grid (Figure 4.6b). Flow grid contains

112 elements, while the transport one 132 triangles.
The aim is the interpolation of fluxes on the transport grid, knowing the solution on the initial flow grid. It is important to note that the new mesh generated (Figure 4.6b) is conformal and consistent with the heterogeneity field, in fact, focusing on Figure 4.5, despite the refinement, edges $E_i$ of initial flow grid remain fixed and so the associated fluxes $Q_i$. The unknowns are the internal compounds $q_i$, but they do not affect the balance, because they are built internally to the triangle $T$, where continuity equation holds:

$$q_1 + q_2 + q_3 \simeq 0 \tag{4.7}$$

In this way, the velocity structured is not altered, despite the change of the initial flow grid, where the permeability field was defined. This is the main reason, we choose this test case, in fact it allows to directly compute the Darcy's problem on the transport grid and so compare it with the interpolated results, always based on the transport grid. Therefore we label as *DNS* the direct numerical solution via discretization of Darcy's law and *CI* the interpolated velocity field.

The code implemented for the interpolation is slightly different from the algorithm,

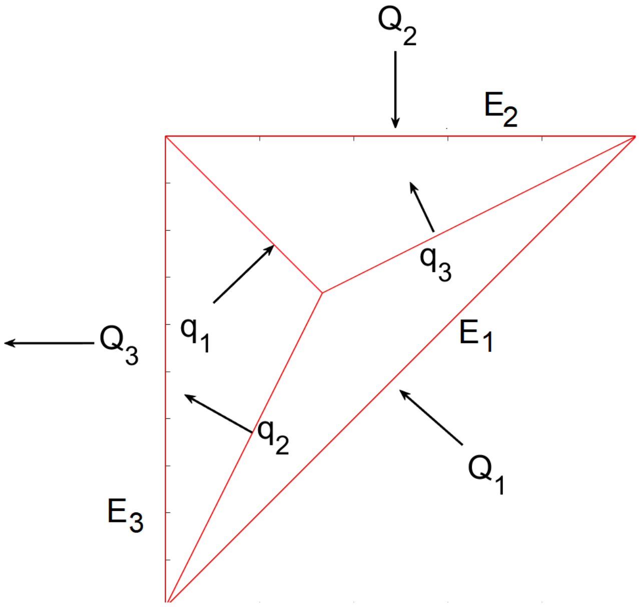


**Figure 4.5:** Flux balance on a refined triangle

previously introduced in Section 4.1, in fact it does not need to calculate intersections, since we simply add a point per triangle refined. Transport grid is not read as input data, but it is automatically generated upon the number and location of investigation points. Nevertheless the computation of the interpolated flux remains identical. Furthermore, this code prototype has been implemented in *Matlab* and not in *Freefem.*

In order to verify the consistency of the strategy developed, two kind of analysis are performed:

- a comparison among *DNS* and *CI* by means of an error estimator;

- a comparison based on a transport simulation, where the concentration evolution is calculated either with *DNS* and *CI* velocity field.

We denote as $err_{L^1}$, a scalar map of error estimator in $L^1$-norm, which is defined as:

$$err_{L^1}(\mathbf{x}) = \sum_{i=1}^{3} |q_i - \widehat{q}_i|_K \tag{4.8}$$

where $q_i$ is obtained from *DNS*, whereas $\widehat{q}_i$ is the interpolated solution from *CI*. $K$ is the considered triangle. We remind all quantities in Equation 4.8 are referred to the transport grid in Figure 4.6b.

Log-velocity field is depicted in Figure 4.7, while logarithmic spatial distribution of the error is represented in Figure 4.8a. We could immediately note that there is a mild difference only in the refined triangles, in fact in the other zone, flow field remains constant. The gap is negligible ($err_{L^1} \approx 10^{-14}$) and it does not affect the flux balance, as we could note from Figure 4.9, which represents a zoom of the square area $[0.02, 0.04] \times [0.06, 0.08]$ of the flux balance map.

We have demonstrated the interpolation algorithm provides results, which are consistent with the Darcy's law and they accomplish the continuity constraint, in the same manner as *DNS* does.

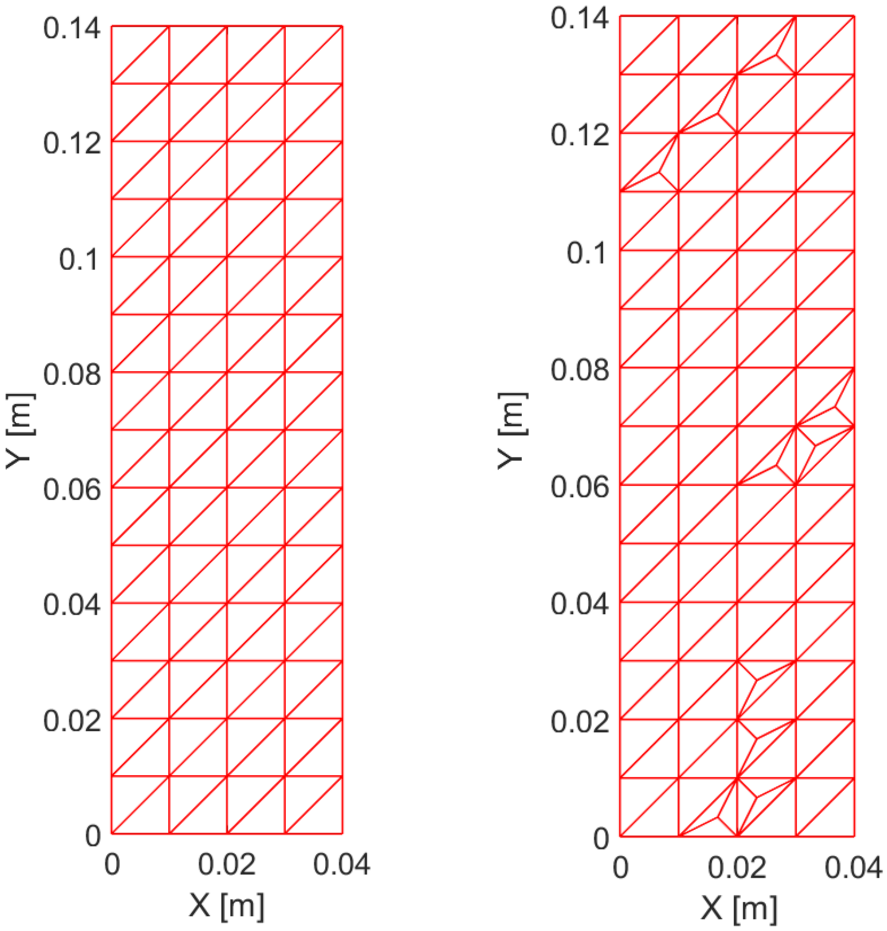
Therefore, the aim is now to evaluate the effects of the interpolated flow, generated through the algorithm, on the transport of species.

We run two types of simulations, which follow a fixed strategy in time and space, both are discretized on the transport grid in Figure 4.6b, but the former is based on the Darcy's *DNS*, the other on the interpolated velocity *CI*. In order to detect any differences in the solution, we identify seven points, all located in the refined triangle elements (see Table 4.1 and Figure 4.7). Concentration evolution is calculated along time.

We focus on Figures from 4.10 to 4.13, where we maintain the same nomenclature, previously introduced, *DNS* and *CI*. The former depics concentration evolution, obtained from the direct resolution of Darcy's law, the latter via interpolated velocity field. Both trend overlap each other, therefore we could state the *CI* algorithm is stable, for the sake of transport.

| **Point** | **x** | **y** |
|---|---|---|
| $P_1$ | 0.022 | 0.026 |
| $P_2$ | 0.036 | 0.071 |
| $P_3$ | 0.015 | 0.02 |
| $P_4$ | 0.025 | 0.006 |
| $P_5$ | 0.026 | 0.062 |
| $P_6$ | 0.034 | 0.006 |
| $P_7$ | 0.022 | 0.016 |

**Table 4.1:** Coordinates of investigation points



**(a)** Initial regular flow grid

**(b)** Example of simple local mesh refinement

**Figure 4.6:** Initial and refined mesh

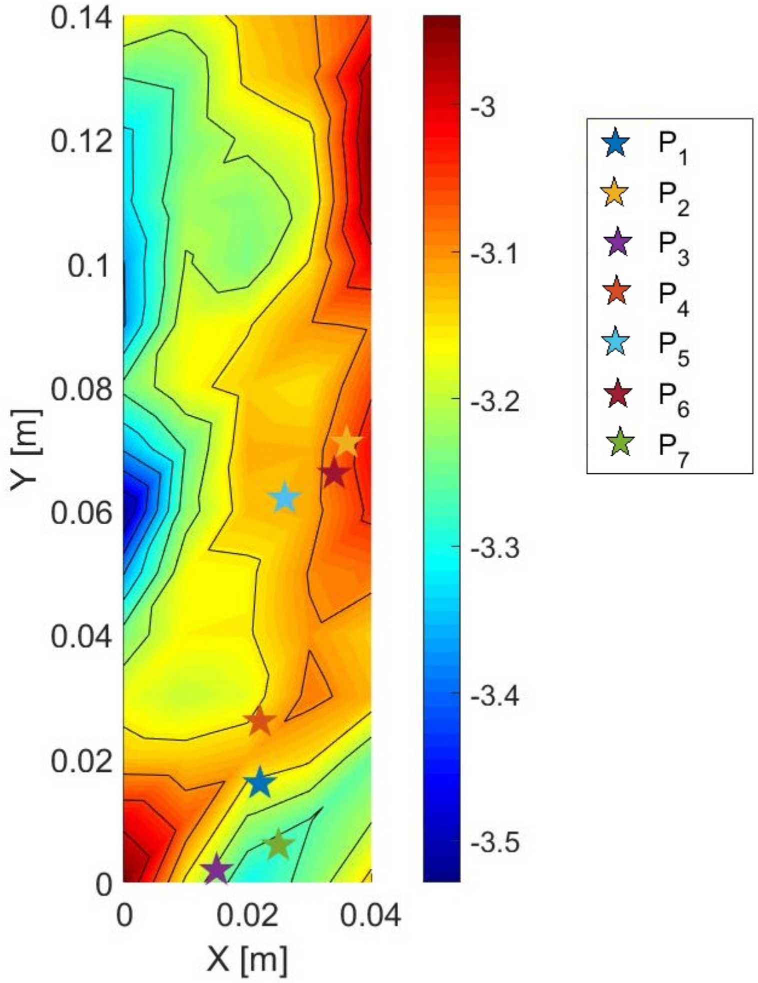
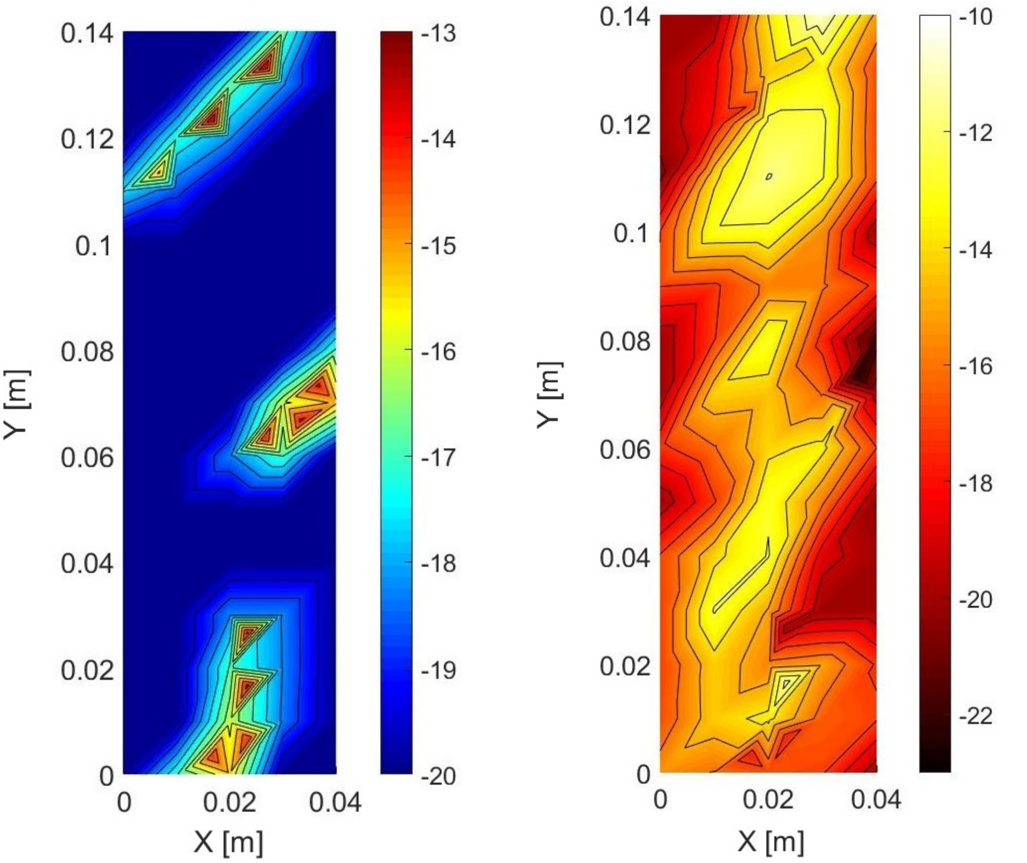


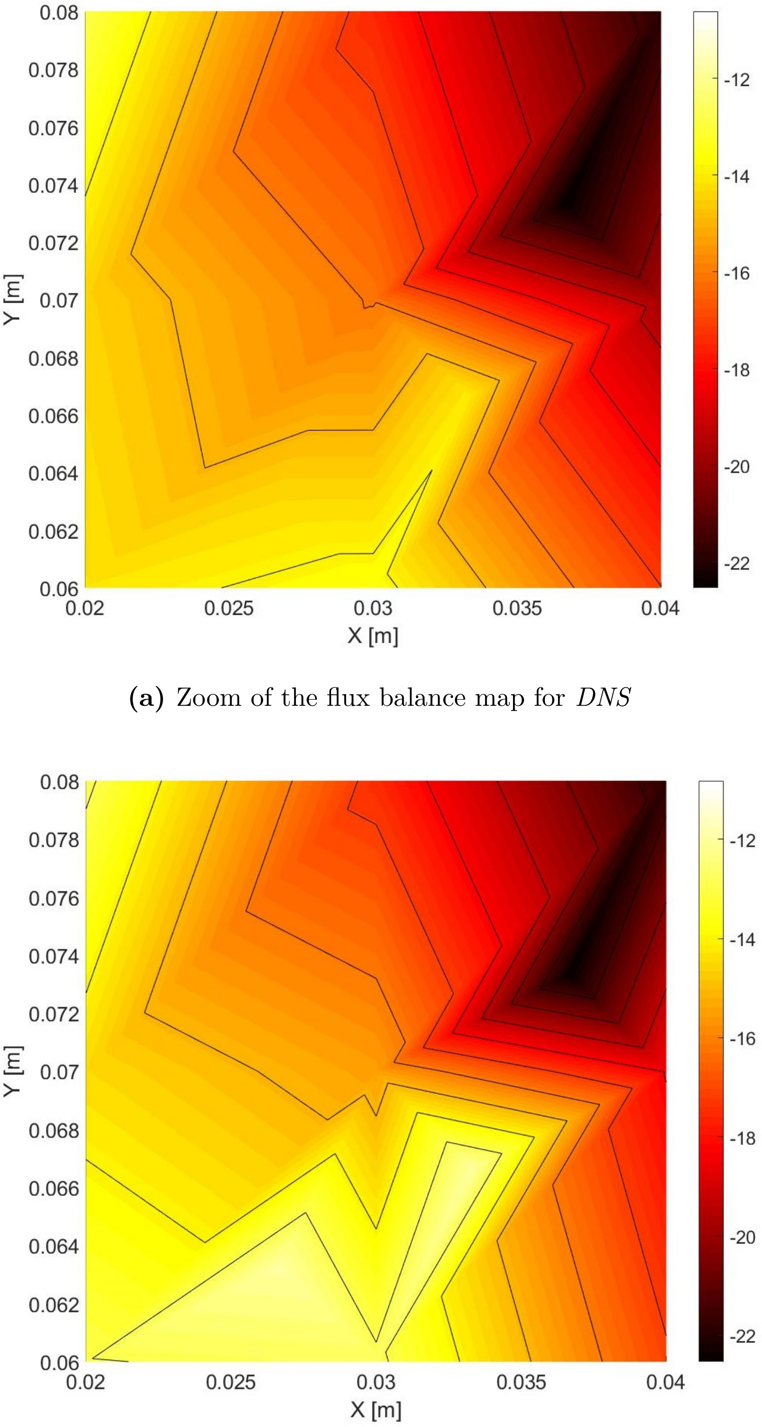
**Figure 4.7:** Log-velocity norm distribution, direct numerical solution through Darcy's law



**(a)** Spatial error distribution of Equation (4.8)

**(b)** Flux balance map of interpolated solution *CI*

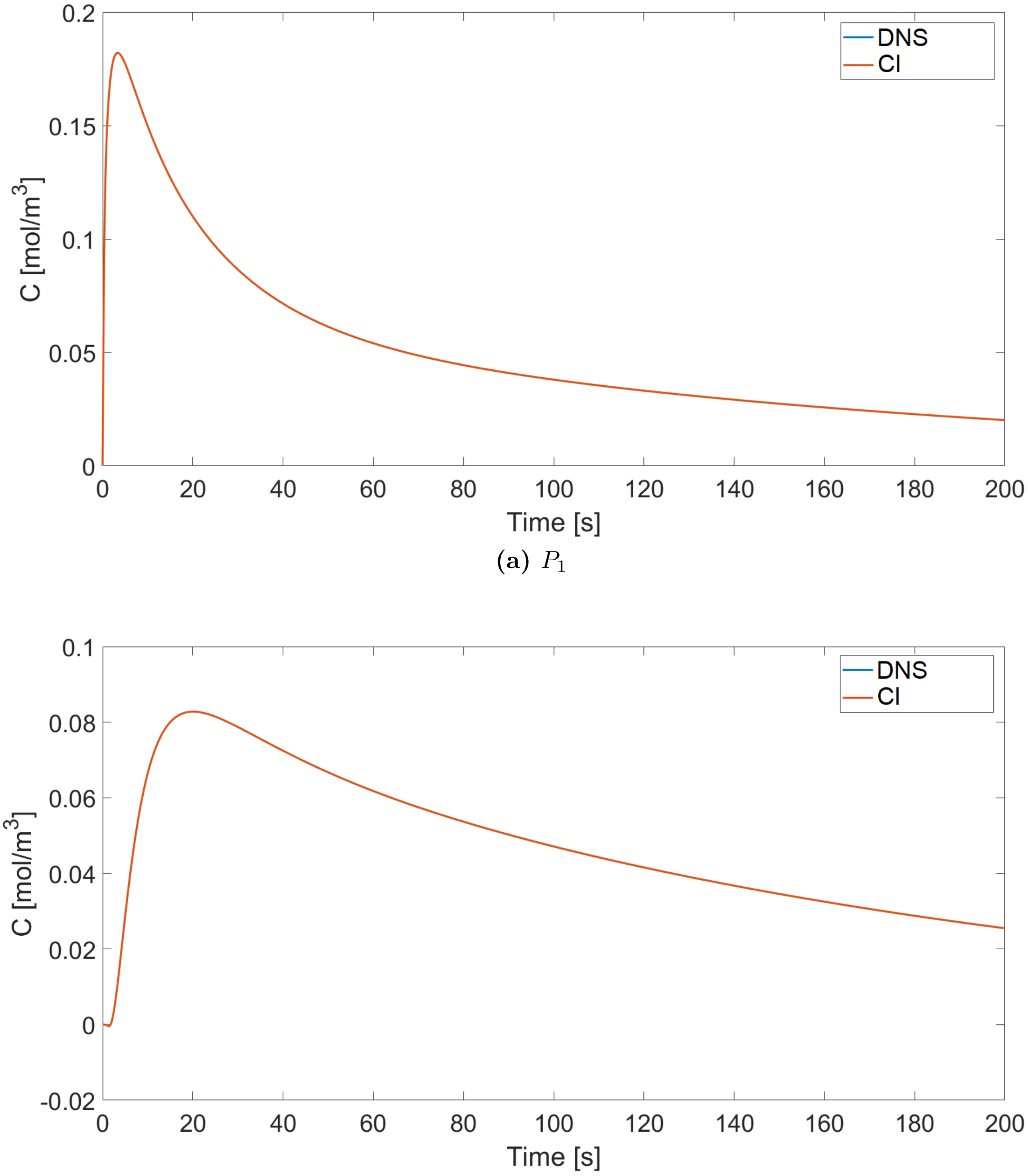
**Figure 4.8:** Spatial error distribution and flux balance map



**(a)** Zoom of the flux balance map for *DNS*

**(b)** Zoom of the flux balance map for *CI*

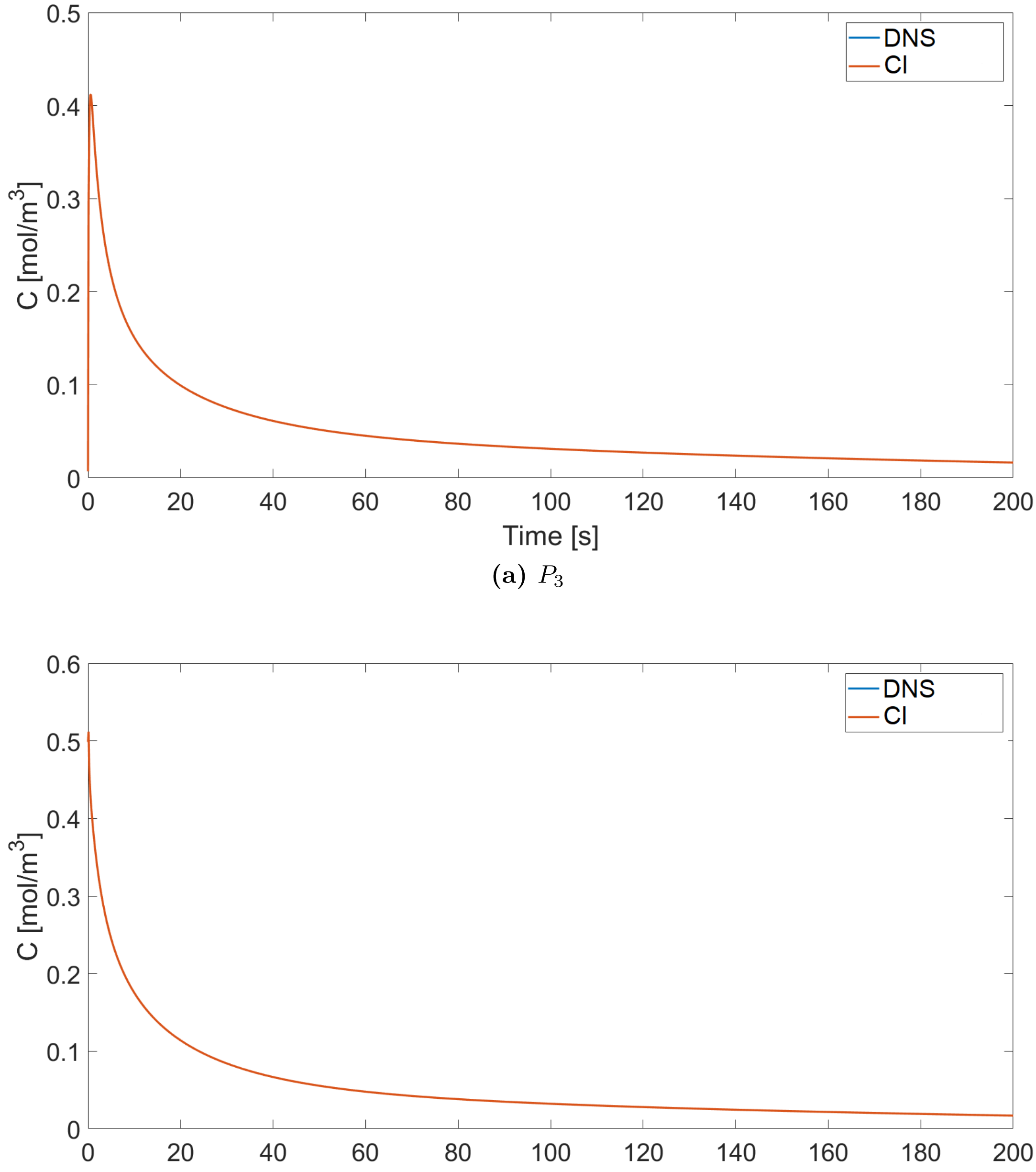
**Figure 4.9:** Test case for code verification



**(a)** $P_1$

**(b)** $P_2$

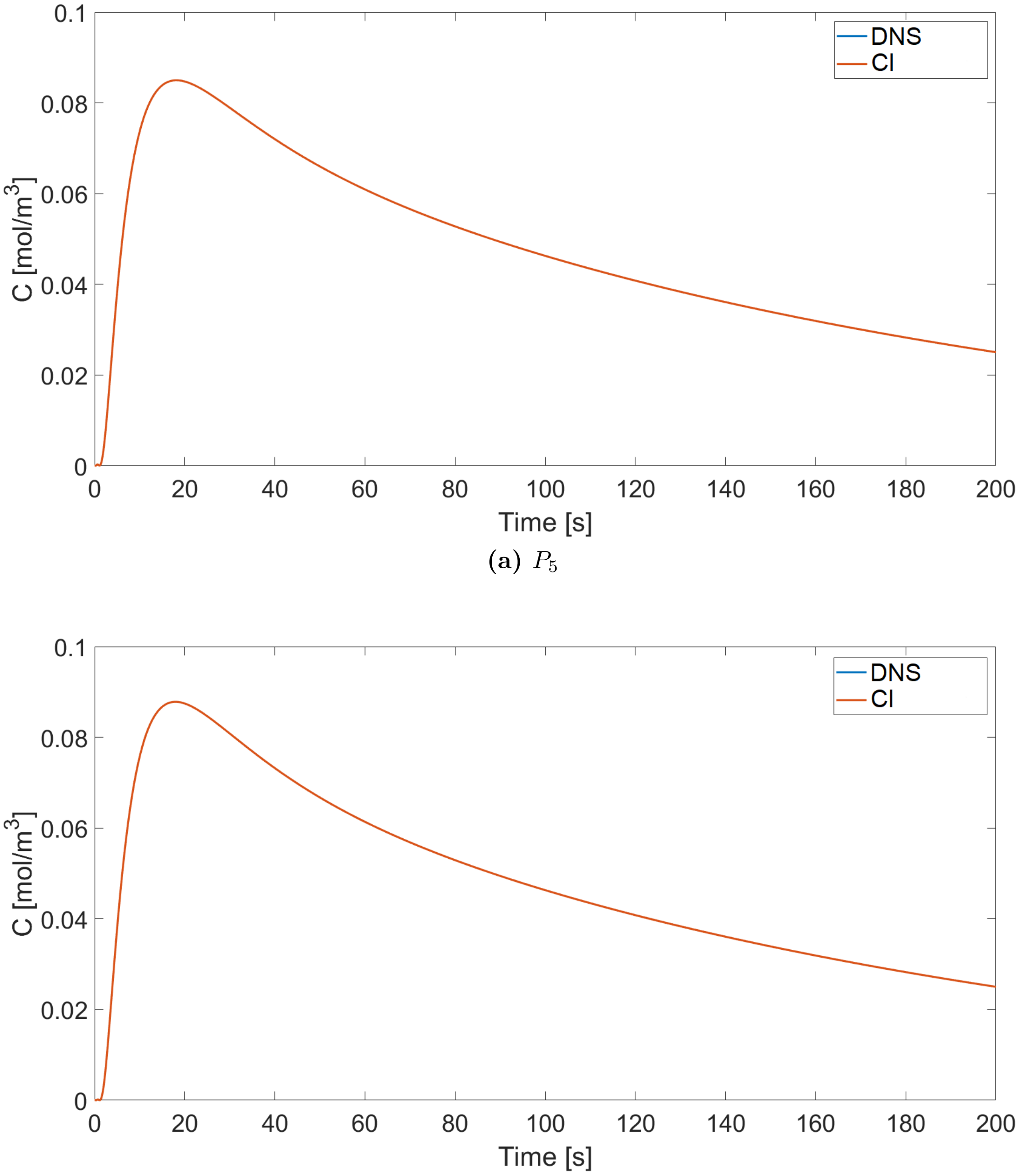
**Figure 4.10:** Concentration evolution of point $P_1$ and $P_2$



**(a)** $P_3$

**(b)** $P_4$

**Figure 4.11:** Concentration evolution of point $P_3$ and $P_4$



**(a)** $P_5$

**(b)** $P_6$

**Figure 4.12:** Concentration evolution of point $P_5$ and $P_6$

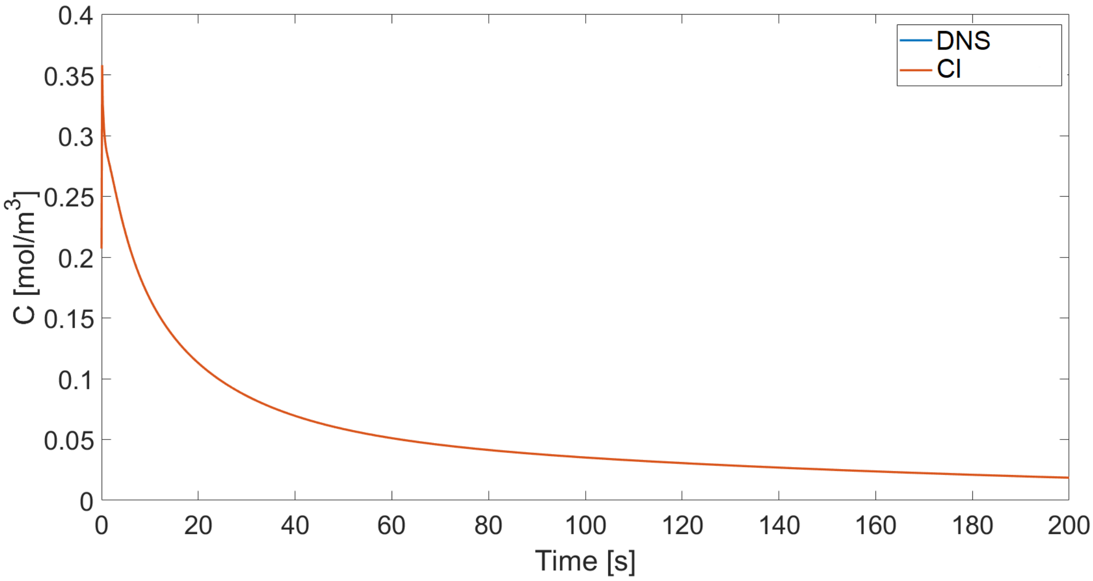


**Figure 4.13:** Concentration evolution of point $P_7$

### 4.2.2 Study case

From previous section, we have demonstrated that the *CI* (conservative interpolation) algorithm works and it is consistent with approximated Darcy's field, however the mesh, taken into account to run the simulation, was almost regular, therefore now the difficulty level (so the computational cost) is raised by considering as transport grid, an adaptive mesh generated during the simulation.
This section deals with the comparison of results among *NCI* (non-conservative interpolation scheme) and the *CI* algorithm implemented, in order to detect pros and cons of the latter. Control volume and heterogeneity characterization are those of Chapter 3, while two cases, in terms of mesh size, are considered, one coarser, the other finer. The former includes a flow grid with 700 elements and a transport one with 493, the latter is based on a flow mesh with 11200 triangles and a transport one with 8868. They are depicted respectively in Figure 4.14a and 4.15a.

As already discussed in Section 4.1, the *CI* algorithm allows to reproduce a conservative field and maintain at the same time the structure of the velocity, without altering the distribution, despite the change in the mesh topology. It is important to focus on this aspect, because it is the main reason, to which Darcy's law could not be solved on the adaptive grid. Flow equation is an elliptic problem, therefore both downstream and upstream informations are necessary, whereupon a local value alone does not take on any meaning. Assuming for example, a point of maximum velocity, if it was surrounded by a slow zone, the flux would be prevented across it. This leads to the necessity to have a global reference solution of the flow distribution, finding preferential channels and low velocity regions. This is only possible upon the initial grid, where permeability field is defined, while on other mesh the flow structure would be altered.
According to this concept, interpolation is the sole alternative to reproduce the

velocity map from one adaptive mesh to another.
Now the objective is to test the consistency and stability in presence of an anisotropic transport grid, where shape, orientation and size of triangles are completely different either inside the mesh itself and from the original flow grid.

We start with the coarse case. Figure 4.16a depicts the original velocity norm distribution in logarithmic scale, obtained as result of Darcy's numerical solution on the regular grid. In the central region ($0.03 \leq y \leq 0.11$), the left-hand side depicts a slow zone, while the right-hand side is the fast one, with a preferential channel along the eastern boundary. We compare it with the interpolated velocity module, performed by *NCI* and through the new algorithm *CI*. We observe in Figure 4.17, that in the middle region, $0.02 \leq y \leq 0.08$, module and shape of the contour lines are almost identical and consistent with the original velocity norm. However in the top-left side, there is a mild difference, in particular, results of *NCI* detects a low velocity region, along the western boundary, around $y \simeq 0.1$, which is not present in the original distribution.
On the contrary, the improvement accomplished with the *CI* algorithm is clear if we focus on the flux balance map in logarithmic scale, depicted on Figure 4.18. Continuity constraint is respected and no artificial generation of fluid mass occurs. The distribution is quite homogeneous in the whole domain, in fact shape, size and orientation of triangles do not affect the balance. The same considerations can not be inferred for the map, generated by *NCI*, where the range of values is influenced by the mesh topology.

Analogous concepts incur for the case of fine meshes in Figure 4.15. Velocity norm in log-scale does not offer any particular differences either in the value and shape of the contour lines. The approximation is much better with respect to the coarse case, because of the increased spatial refinement. However, the relevant differences still occur in the flux balance map, especially in the order of magnitude, which give an idea of the accuracy, involved the flux balance. Results are printed in Figure 4.20b.

From the test cases analysed, it is possible to derive some features of both interpolation methods. We enlist the limits of the actual strategy, yielded by *NCI*:

1. it does not consider the heterogeneity of the system, because, although the edge of triangle intersects different permeability regions, the flux is computed, taking into account only one point, so one permeability value;

2. it does not respect the fluid mass balance;

3. the distribution of the log-flux balance map is affected by the mesh topology.

According to Section 1.4, *NCI* is computed, taking into account only the middle point of the edge. It is a sort of average value, which suffers particularly when the element size becomes coarser and coarser, because depending on the local permeability, it could underestimate or overestimate the interpolation. On the contrary, the *CI* algorithm is built on a rigorous process of intersection, in order to find all the segments $S_i$, which composes the edge $L_i$. In this way the resulting flux comes

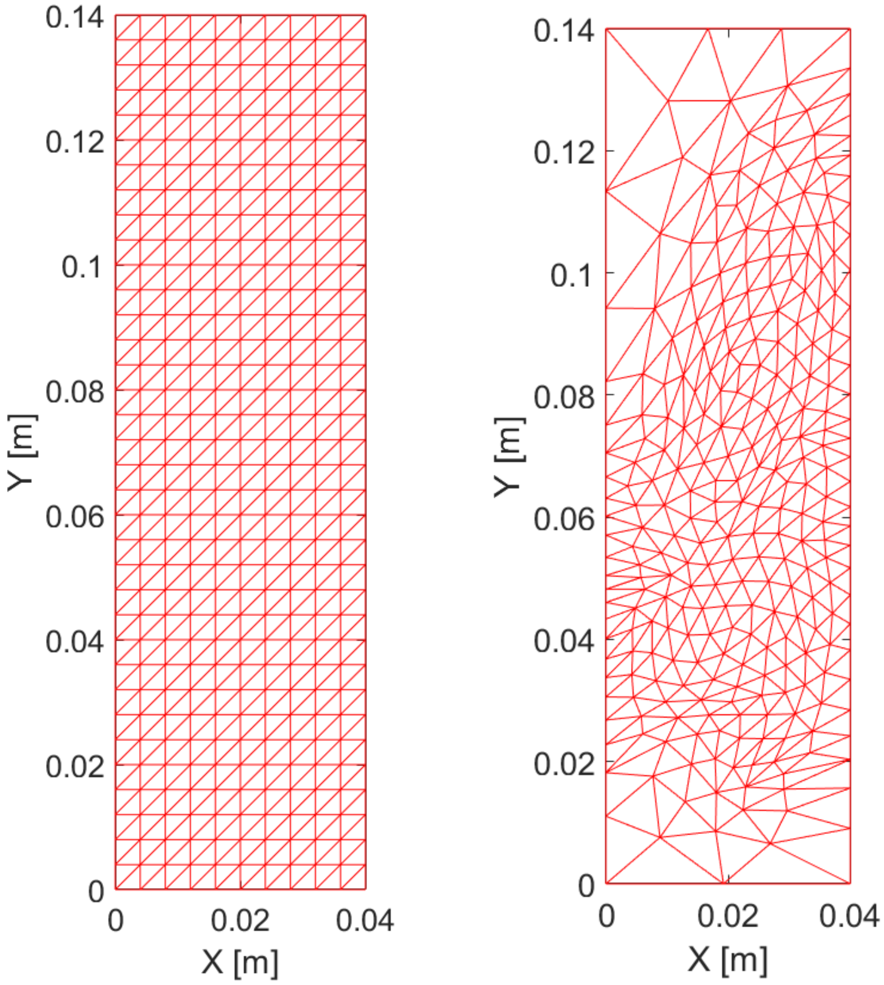
from $N_c$ contributions, with $N_c$ the number of segments $S_i$, each one with a different permeability value. Some are low, other are high, but the main advantage is that $q_i$ is computed as a correct summation.
As snowball effect, the strategy adopted by *NCI* does not fulfil fluid mass balance. This is evident from the log-flux balance maps previously shown. Our approach, instead is either locally and globally conservative. Locally, because each flux $q_{i,j}$ (associate to a segment $S_i$, which composes the edge $L_j$ of $K$) maintains the balance on the triangle $T_i$ of the flow grid (see Section 4.1 for further details). Globally, because the summation of $q_j$ is null, meaning that the continuity constraint holds.
Another common feature of the log-flux balance maps, generated by *NCI*, is the anisotropic distribution, namely they are influenced by mesh topology. Finer regions are more accurate, in terms of order of magnitude of flux balance than coarser one. It is still a consequence of the single point-wise strategy. The lower the area of triangles in transport grid, the higher the probability they are included in a single element of the flow grid. In this case both interpolation methods are acceptable and consistent with the permeability field, however it is a particular case, which does not make *NCI* general, *i.e.* the solution is affected by the geometrical features of the considered element.

As last parameter to investigate, Figure 4.21 depicts the computational time against the number of elements of transport grid, in logarithmic scale.
The *CI* code implemented is able to solve the numerical issues, deriving from the non-conservative interpolation method *NCI*, however the computational time is significant, especially when the mesh cardinality increases (see Table 4.2).

| **Number of elements** | **CPU [s]** |
|---|---|
| 70 | 5.35 |
| 493 | 40.18 |
| 8868 | $2.54{\cdot}10^3$ |

**Table 4.2:** CPU against number of elements

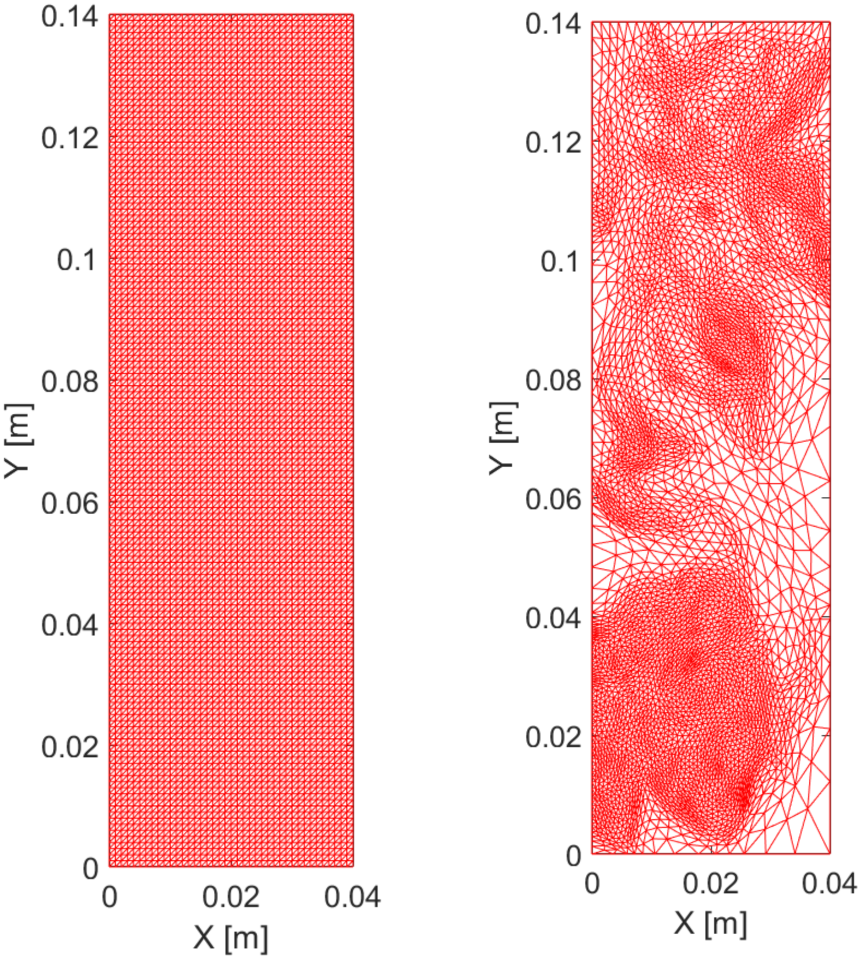
Adaptive setting obliges us to perform a flux interpolation, to project the initial flow map on the transport grid, at each time step. At this point, the conservative algorithm is quite efficient from a heuristic view, but it is not the optimal choice, due to the high cost, in terms of computational time. The code inevitably needs to be optimized under this aspect, in fact up to now, coupling it with the adaptive procedure makes this strategy unfeasible and the user would prefer to employ other types of code, based on a fix discretization.



**(a)** Regular flow grid

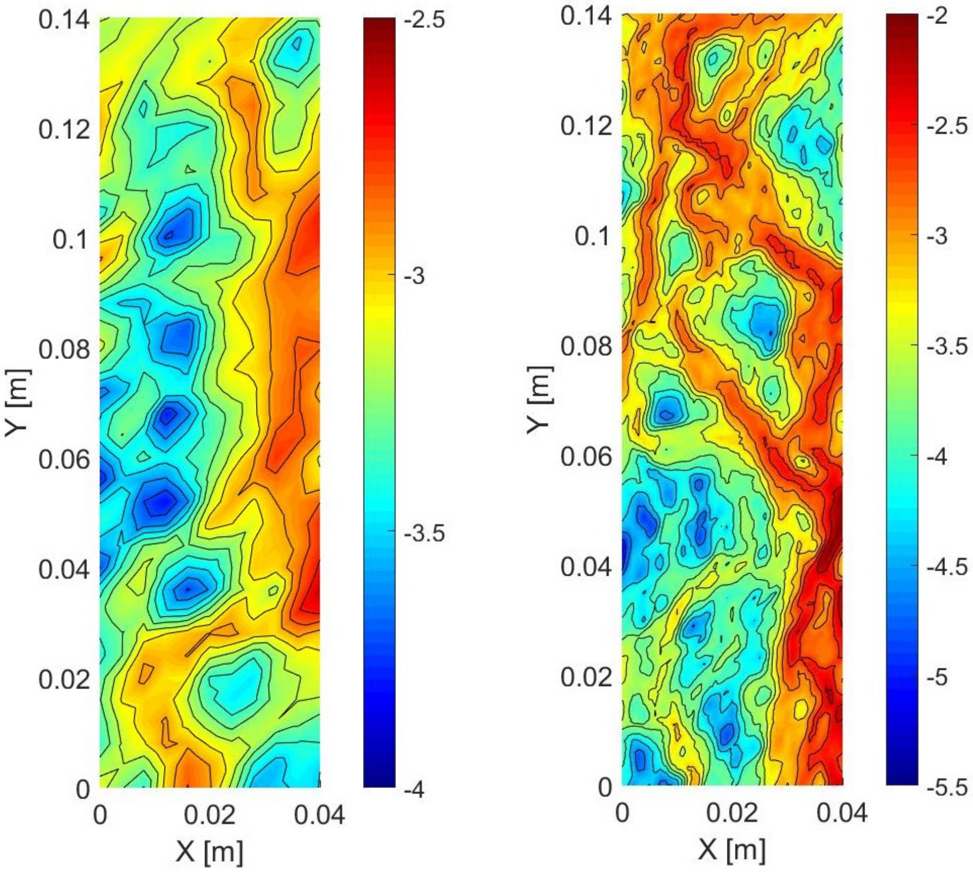
**(b)** Adaptive transport grid

**Figure 4.14:** Coarse mesh



**(a)** Regular flow grid **(b)** Adaptive transport grid

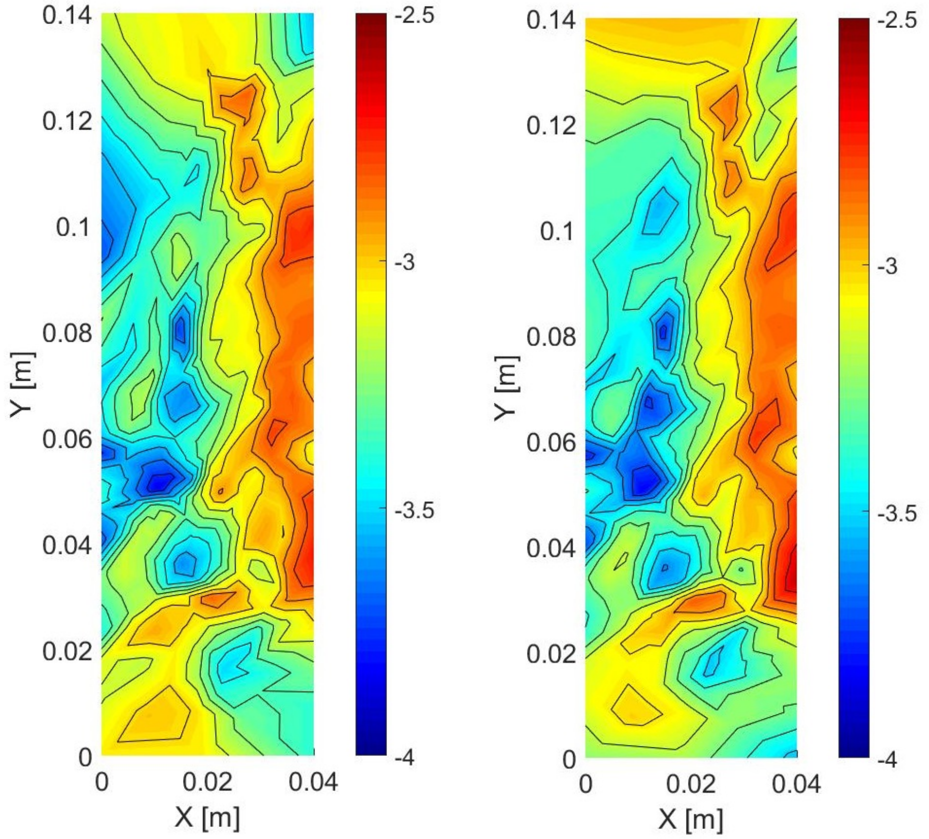
**Figure 4.15:** Fine mesh



**(a)** Coarse regular grid

**(b)** Fine regular grid

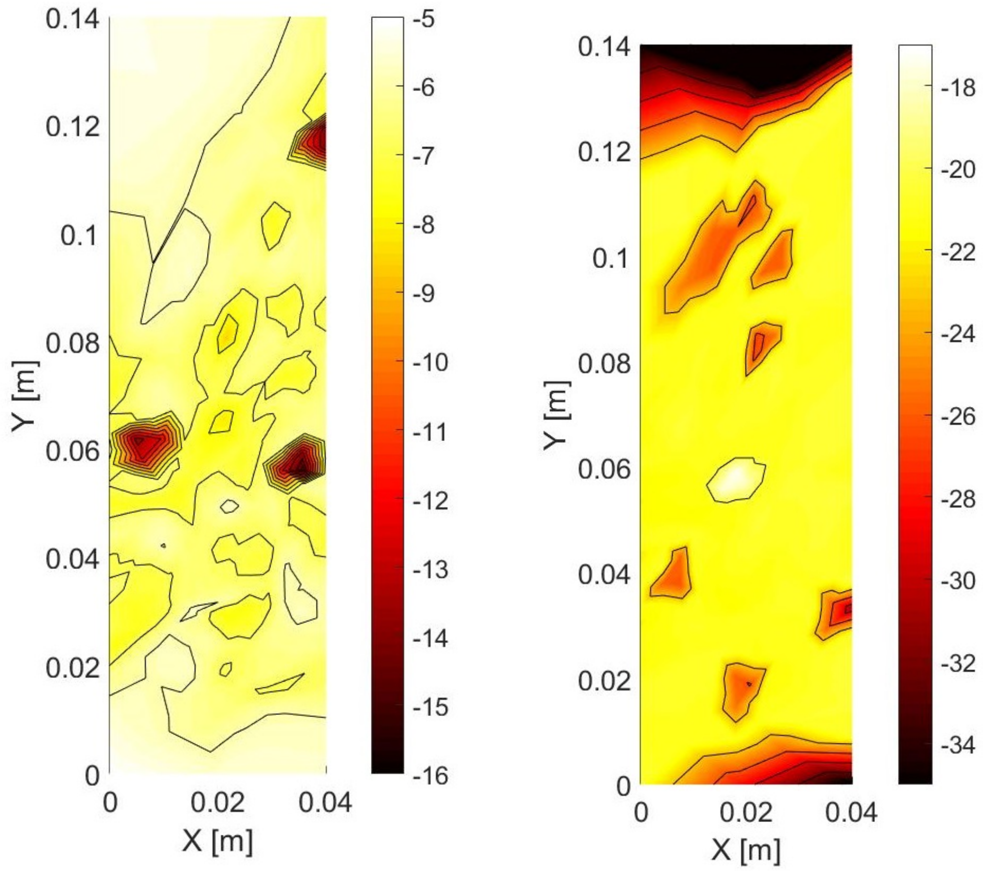
**Figure 4.16:** Log-velocity norm map, computed on regular flow grid as direct numerical solution of Darcy's law



**(a)** Log-velocity map computed through *NCI*

**(b)** Log-velocity map computed through *CI*

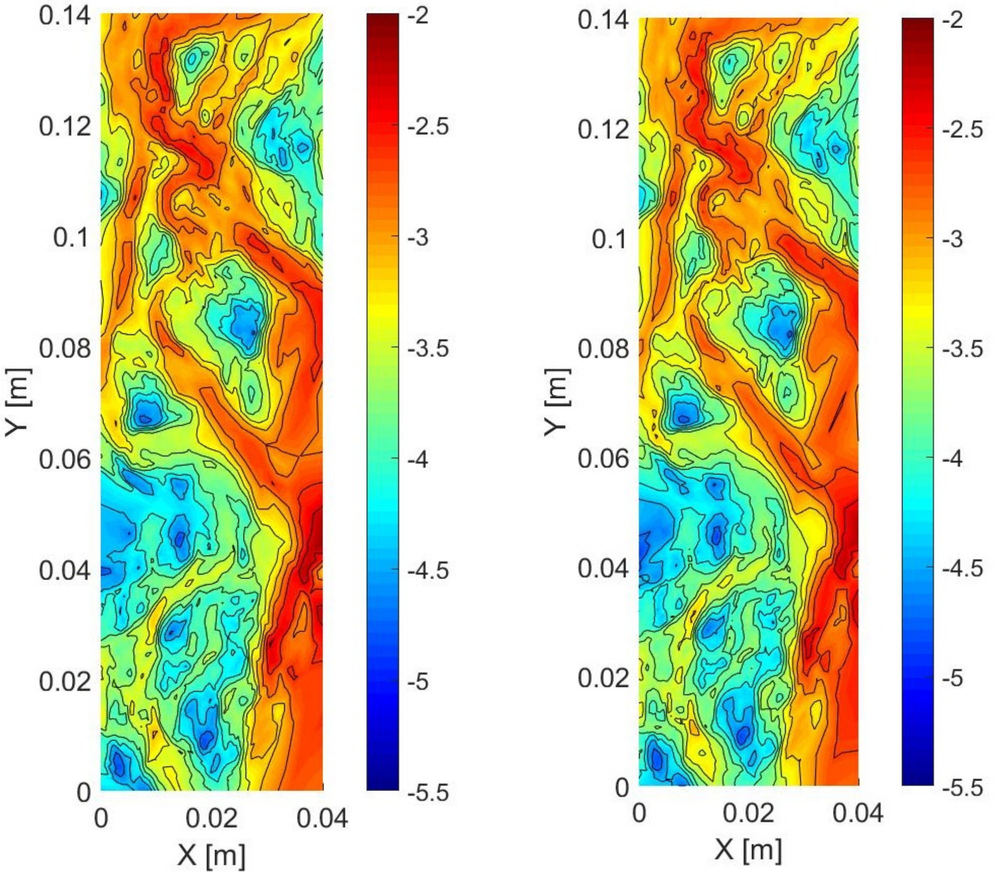
**Figure 4.17:** Comparison upon log-velocity norm map, *coarse case*



**(a)** Log-flux balance map computed through *NCI*

**(b)** Log-flux balance map computed through *CI*

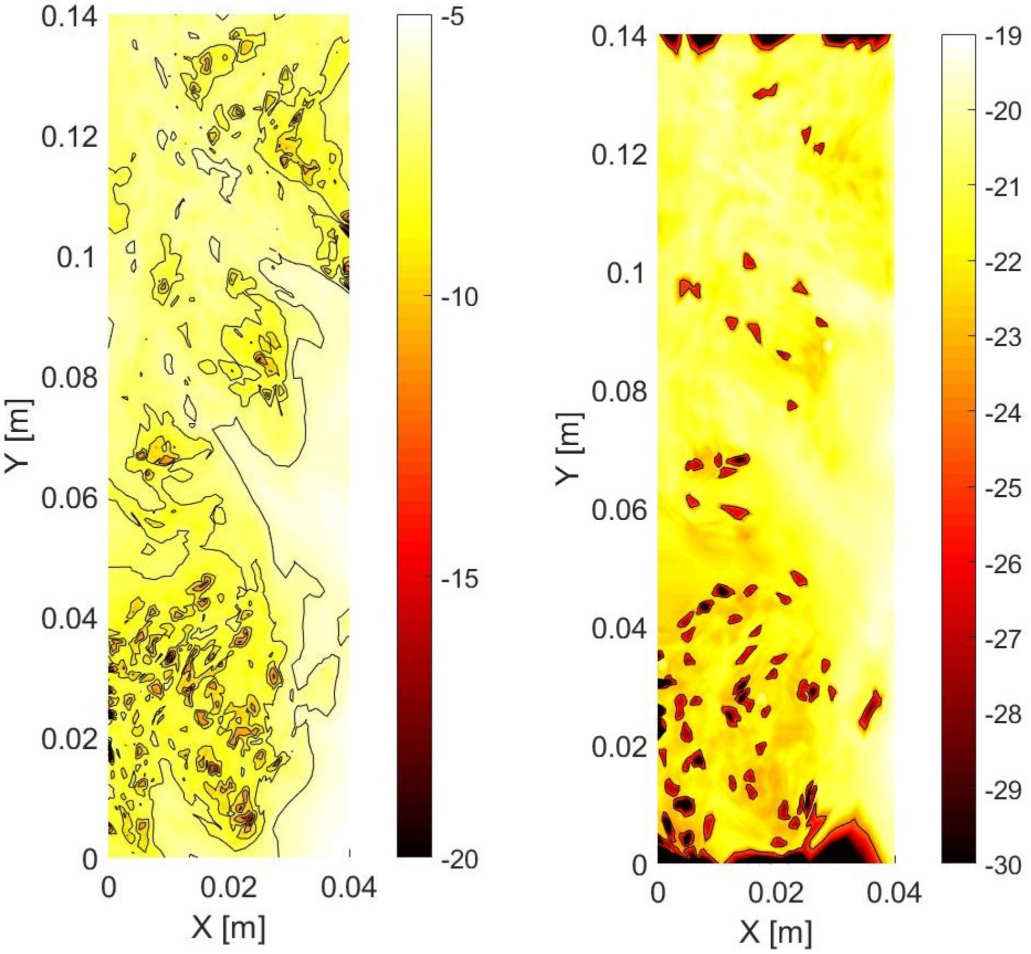
**Figure 4.18:** Comparison upon log-flux balance map, *coarse case*



**(a)** Log-velocity map computed through *NCI*

**(b)** Log-velocity map computed through *CI*

**Figure 4.19:** Comparison upon log-velocity norm map, *fine case*



**(a)** Log-flux balance map computed through *NCI*

**(b)** Log-flux balance map computed through *CI*

**Figure 4.20:** Comparison upon log-flux balance map, *fine case*

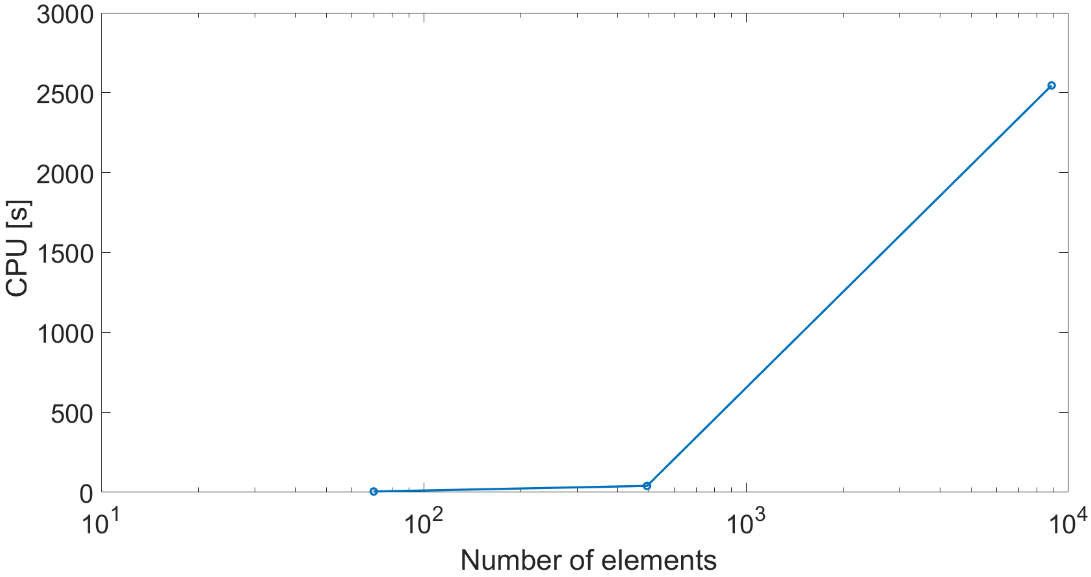


**Figure 4.21:** Computational time against number of elements

# Conclusion

In the context of petroleum engineering, the study of subsurface transport in porous media became particularly relevant and it is a key component in the design and optimization of oil and gas production operations or in the planning and management of geological carbon sequestration. A deep theoretical and analytical knowledge is a fundamental step to treat such themes, however the critical challenge is the development of numerical methodologies, rendering suitable approximations of the space-time dynamics of concentrations field in highly heterogeneous media.
In this thesis work, we assess the behaviour of a space-time adaptation technique (labelled as *Adaptive*), grounded on an anisotropic a posteriori error estimator, when it is applied to the numerical resolution of solute transport in both homogeneous and heterogeneous domain. The latter is quantified in terms of random Gaussian permeability field at constant variance.
In order to ensure confidence on the results, we carry out a benchmarking process, relying on two codes, *Traces* and *Fix*. Both are implemented via fixed space-time strategy in a Finite Element Framework (as well as *Adaptive*). *Traces* is built on Discontinuous Galerkin scheme , which allows to limit numerical error and oscillations, while *Fix* develops the same Galerkin approximation of *Adaptive.* The choice of the codes and their peculiarities enables us to carry out a deep and complete analysis of flow and transport problem. We develop a double vision of code benchmarking, involving both the discretization scheme (Streamline Upwind versus Discontinuos Galerkin) and the space-time procedure (fix versus adaptive strategy). Several scenarios of flow (homogeneous and heterogeneous domain) and transport (advective or diffusive dominated) problem are investigated with the aim to figure out strength and weaknesses of each solution strategy.
From the simulations performed, we can come to the following conclusions:

1. First test case is based on a homogeneous domain with a transport process, dominated by dispersion. It reveals all codes are accurate in outlining spatial concentration distribution and local evolution along time. However, raising grid Péclet number, by varying dispersivity values, leads to a transport governed by advection, which generates numerical diffusion in the concentration map. In different measure, all codes suffer from this instability. *Adaptive* and *Traces* show a significant ability in limiting oscillations, while the Streamline Upwind technique, employed in *Fix*, demonstrates to be unreliable.

2. Second test is based on solute transport, which takes place in heterogeneous system. Solutions provided by *Fix* and *Traces* are almost identical. Grid adaptation manages to optimize triangles shape, size and orientation upon

the concentration front evolution. It allows to reduce the number of elements, necessary for an accurate description of the solute transport at hand, however it is not able to handle with the complex structure of the velocity field, due to the imposition of high heterogeneity. We note discrepancies in the spatial concentration distribution and oscillations in the local evolution, especially in the low velocity region. Projection of advective field is a mandatory, but problematic step, due to the complex structure of velocity field and the significant change in the mesh topology between flow and transport grid. Projection is actually performed through *NCI* (non-conservative interpolation) algorithm via linear interpolation, however results show us oscillations in the concentration evolution and shortage in the flux balance. From preliminary test, we evaluate that numerical inaccuracies tend to decrease, when the error in the flux balance decreases as well.

3. To solve the issues, related to previous point, we implement *CI* (conservative interpolation) algorithm. It is the main novelty of this thesis work. It succeeds in overcoming the above limits, in fact flux balance is completely respected, the distribution of divergence map is uniform, which ensures no matches with the mesh topology. Furthermore, the interpolated convective field is consistent with the original Darcy's problem. We can assure the fluid mass redistribution is done, accordingly with the heterogeneity structure.
   The scheme introduced is the key original feature of this study. It manages in handling with meshes of different order of magnitude, in terms of cardinality, eventhoug the increase of computational cost is significant.

**Future works**

Future works of this thesis should be concerned on the optimization of the algorithm, in terms of computational cost. Coupling it with the adaptive strategy is the main objective to achieve, in order to observe the effects of a conservative approach on the concentration distribution. An increase of the complexity level could involve the study at different grid Péclet range, to deal with a process dominated by both diffusion/dispersion and convection. Finally it should be envisioned to include chemical reaction term inside the ADE and have a more realistic description of the solute transport.

# Appendix A

# Appendix A

## A.1 Traces Code

*Traces* is a computer programme for the simulation of flow and reactive transport in saturated porous media. It employs finite volume formulation to ensure exact local mass balance, handle high parameter discontinuities between adjacent elements and treat full tensor without approximation. Following [6], mathematical model describing the flow in the porous material, System (1.14), is solved by the mixed hybrid finite element (label as MHFEM). Transport equation, Equation (A.6), is split in two parts, where the advective term is solved by discontinuous Galerkin finite element (label as DGFEM) and the rest by mixed hybrid method. We consider 2D polygonal domain, triangle mesh discretization and take as reference the following nomenclature, previously described in Section 1.2:

- $T$: generic triangle in the mesh $\mathcal{T}_h$;
- $E_i$: edge $i$, (with $i = 1, 2, 3$) of generic triangle $T$;
- $N_t$: number of triangle elements in $\mathcal{T}_h$;
- $N_e$: number of edges in $\mathcal{T}_h$;
- $N_f$: number of edges of $T$, therefore $N_f = 3$.

### Raviart-Thomas space

In the Raviart-Thomas space of lowest order ($RT0$), the elementary fluxes at the element $T$ level are defined by:

$$\boldsymbol{\psi}^T = \sum_{i=1}^{N_f} \boldsymbol{\varpi}_i^T \Psi_i^T \tag{A.1}$$

where $N_f$ is the number of edges of the element $T$ and $\Psi_i^T$ are the fluxes through the edge $i$. The vectorial basis function $\boldsymbol{\varpi}_i^T$ for element $T$ is defined by:

$$\int_{Ei} \boldsymbol{\varpi}_i^T \cdot \mathbf{n}_j = \begin{cases} 1 \text{ if i=j} \\ 0 \text{ if i}\neq\text{j} \end{cases} \tag{A.2}$$

where $\mathbf{n}_j$ is the normal outwardly oriented vector and $j$ the edge $j$ of $T$.
Due to Equation (A.2), the properties of $\boldsymbol{\varpi}_i^T$ fulfil:

$$\int_T \nabla \cdot \boldsymbol{\varpi}_i^T \, dT = \frac{1}{|T|} \tag{A.3}$$

where $|T|$ is the area of the element $T$. In addition:

$$\boldsymbol{\varpi}_i^T \cdot \mathbf{n}_j = \begin{cases} 1/|E_i| \text{ if j=i} \\ 0 \text{ if j}\neq\text{i} \end{cases} \tag{A.4}$$

where $|E_i|$ is the length of the edge $i$.

### A.1.1 Discretization

Consider the flow and transport equations, respectively reported in Equation (A.5) and (A.6):

$$\partial_t P_H - \nabla \cdot (\boldsymbol{K_H} \nabla P_H) = 0 \tag{A.5}$$

$$\partial_t c - \nabla \cdot (\mathbf{D} \nabla c - \mathbf{u} c) = 0 \tag{A.6}$$

Despite the different formulation with respect to the original Problem (1.14), Equation (A.5) is perfectly equivalent, in fact temporal derivative of hydraulic head $P_H$, already introduced in Section 1.1.1, is null, because of assumption of stationariness.
For sake of simplicity, both problems could be generalized to a common form, expressed in the system of equations (A.7):

$$\begin{cases} S \partial_t \Theta + \nabla \cdot \boldsymbol{\psi} + S \lambda \Theta = f \\ \boldsymbol{\psi} = -\boldsymbol{\Upsilon} \nabla \Theta \end{cases} \tag{A.7}$$

where $S$ is the storage coefficient (flow) or the retardation factor (transport), $\Theta$ the state variable (head or concentration), $\lambda$ a kinetic term, $f$ a sink/source term, $\boldsymbol{\Upsilon}$ is a tensor (hydraulic conductivity or dispersion) and $\boldsymbol{\psi}$ the related fluxes. First equation of the System (A.7) states for the water or solute mass balance, while the second one for Darcy's or Fick's law. Discretization over the element $T$ leads to:

$$\begin{aligned} \int_T \left( S \frac{\partial \Theta}{\partial t} + \nabla \cdot \boldsymbol{\psi} + \lambda \Theta - f \right) dT = |T| \overline{S}^T \frac{\partial \overline{\Theta}^T}{\partial t} + \int_T \nabla \cdot \left( \sum_{i=1}^{Nf} \boldsymbol{\varpi}_i^T \Psi_i^T \right) dT \\ + |T| \overline{S}^T \overline{\lambda}^T \overline{\Theta}^T - \overline{\sigma}^T = |T| \overline{S}^T \frac{\partial \overline{\Theta}^T}{\partial t} + \sum_{i=1}^{Nf} \Psi_i^T + |T| \overline{S}^T \overline{\lambda}^T \overline{\Theta}^T - \overline{\sigma}^T = 0 \end{aligned} \tag{A.8}$$

where $\overline{\Theta}^T$ is the average value of the state variable over element $T$, $\overline{S}^T$ is the approximation of $S$, assumed to be constant over the element $T$, $\Psi_i^T$ are the fluxes through the $i^{th}$ edge (or face) of $T$, $\overline{\sigma}^T$ is the source or sink term, associated with the element $T$ (average value fo $f$ over $T$) and $N_t$ the number of elements in the grid.

The variational form of constitutive law, Equation (A.7), using Green's theorem, leads to:

$$\begin{gathered}\int_T \left(\left(\Upsilon^T\right)^{-1} \boldsymbol{\psi}^T\right) \cdot \boldsymbol{\varpi}_i^T = -\int_T \nabla\Theta \cdot \boldsymbol{\varpi}_i^T = \\ \int_T \Theta\nabla \cdot \boldsymbol{\varpi}_i^T - \sum_{i=1}^{Nf} \int_{Ei} \Theta\boldsymbol{\varpi}_i^T \cdot \mathbf{n}_i \quad i = 1, .., N_f(=3)\end{gathered} \tag{A.9}$$

where $\Upsilon^T$ is the approximation of $\boldsymbol{\Upsilon}$, assumed to be constant over the element $T$. Note that in Equation (A.9) the subscript $i$ is referred $i^{th}$ edge of triangle $T$. Using properties (A.4) and (A.3) of the vectorial basis function $\boldsymbol{\varpi}_i^T$, it leads:

$$\int_T \left(\left(\Upsilon^T\right)^{-1} \boldsymbol{\psi}^T\right) \cdot \boldsymbol{\varpi}_i^T = \frac{1}{|T|} \int_T \Theta - \frac{1}{|E_i|} \int_{Ei} \Theta = \overline{\Theta}^T - \overline{\Theta}_i^T \tag{A.10}$$

where $\overline{\Theta}^T$ is the average of the state variable $\Theta$ over the element $T$, and $\overline{\Theta}_i^T$ its average over $i^{th}$ edge. Previous equation can be rewritten as:

$$\sum_{j=1}^{Nf} B_{ij}^T \Psi_j^T = \overline{\Theta}^T - \overline{\Theta}_i^T \quad i = 1, ..., N_f \tag{A.11}$$

where

$$B_{ij}^T = \int_T \boldsymbol{\varpi}_i^T \left(\Upsilon^T\right)^{-1} \boldsymbol{\varpi}_j^T \tag{A.12}$$

and therefore,

$$\Psi_i^T = \sum_{j=1}^{Nf} \left(B_{ij}^T\right)^{-1} \left(\overline{\Theta}^T - \overline{\Theta}_i^T\right) = \alpha_i^T \overline{\Theta}^T - \sum_{j=1}^{Nf} \left(B_{ij}^T\right)^{-1} \overline{\Theta}_j^T \tag{A.13}$$

with

$$\alpha_i^T = \sum_{j=1}^{Nf} \left(B_{ij}^T\right)^{-1} \tag{A.14}$$

To mass balance and constitutive flux laws, continuity of the state variable $\Theta$ and fluxes between two adjacent elements $T$ and $T'$ are added. These continuities are ensured by:

$$\begin{cases} \overline{\Theta}_i^T = \overline{\Theta}_k^{T'} \\ \Psi_i^T + \Psi_k^{T'} = 0 \end{cases} \tag{A.15}$$

### A.1.2 Hybrid scheme

The hybridization consists in writing one equation per element edge or face. The following steps are used:

1. the mass balance equation is rewritten using state variables only;
2. the average element variable is expressed as a function of the average edge values;

3. the flux continuity equation is rewritten using the average edge variables only. This flux continuity equation is slightly modified at Neumann boundaries.

Using Equation (A.13) leads to:

$$\sum_{i=1}^{Nf} \Psi_i^T = \alpha^T \overline{\Theta}^T - \sum_{i=1}^{Nf} \alpha_i^T \overline{\Theta}_i^T \quad \text{with } \alpha^T = \sum_{i=1}^{Nf} \alpha_i^T \tag{A.16}$$

mass conservation law is rewritten in:

$$|T|\overline{S}^T \frac{\partial \overline{\Theta}^T}{\partial t} + \alpha^T \overline{\Theta}^T - \sum_{i=1}^{Nf} \alpha_i^T \overline{\Theta}_i^T + |T|\overline{S}^T \overline{\lambda}^T \overline{\Theta}^T - \overline{\sigma}^T = 0 \tag{A.17}$$

Time derivatives are approximated by a standard implicit scheme, except for the degradation term where:

$$\overline{\Theta}^T = \vartheta \overline{\Theta}^{T,n+1} + (1-\vartheta)\overline{\Theta}^{T,n} \quad 0 \le \vartheta \le 1 \tag{A.18}$$

From Equation (A.17), the average state variable is equal to:

$$\overline{\Theta}^{T,n+1} = \sum_{i=1}^{Nf} \gamma_i^T \overline{\Theta}_i^{T,n+1} + F^T \tag{A.19}$$

with

$$\gamma_i^T = \frac{\Delta t\, \alpha_i^T}{\left(|T|\overline{S}^T + \alpha^T \Delta t + |T|\overline{S}^T \overline{\lambda}^T \vartheta \Delta t\right)} \tag{A.20}$$

and

$$F^T = \frac{|T|\overline{S}^T \overline{\Theta}^{T,n} + \overline{\sigma}^T \Delta t - |T|\overline{S}^T \lambda^T (1-\vartheta)\Delta t}{|T|\overline{S}^T + \alpha^T \Delta t + |T|\overline{S}^T \overline{\lambda}^T \vartheta \Delta t} \tag{A.21}$$

The flux continuity equation, System (A.15), is modified using (A.13):

$$\alpha_i^T \overline{\Theta}^T - \sum_{j=1}^{Nf} \left(B_{ij}^T\right)^{-1} \overline{\Theta}_j^T + \alpha_k^{T'} \overline{\Theta}^{T'} - \sum_{j=1}^{Nf} \left(B_{kj}^{T'}\right)^{-1} \overline{\Theta}_j^{T'} = 0 \tag{A.22}$$

The element average value of the state variable given by Equation (A.19) is replaced in (A.22):

$$\begin{aligned} &\alpha_i^T \left( \sum_{i=1}^{Nf} \gamma_i^T \overline{\Theta}_i^{T,n+1} + F^T \right) - \sum_{j=1}^{Nf} \left(B_{ij}^T\right)^{-1} \overline{\Theta}_j^T + \\ &\alpha_k^{T'} \left( \sum_{i=1}^{Nf} \gamma_i^{T'} \overline{\Theta}_i^{T',n+1} + F^{T'} \right) - \sum_{j=1}^{Nf} \left(B_{kj}^{T'}\right)^{-1} \overline{\Theta}_j^{T'} = 0 \end{aligned} \tag{A.23}$$

which leads to the following system equations:

$$\begin{aligned} \sum_{j=1}^{Nf} \left[ \alpha_i^T \gamma_j^T - \left(B_{ij}^T\right)^{-1} \right] \overline{\Theta}_j^{T,n+1} + \sum_{j=1}^{Nf} \left[ \alpha_k^{T'} \gamma_j^{T'} - \left(B_{ij}^{T'}\right)^{-1} \right] \overline{\Theta}_j^{T',n+1} \\ = -\alpha_i^T F^T - \alpha_k^{T'} F^{T'} \end{aligned} \tag{A.24}$$

At Neuman boundaries, denote $\Psi_N$ as the normal flux, Equation (A.15) is modified in:

$$\begin{cases} \Psi_i^T + \Psi_N = \alpha_i^T \overline{\Theta}^T - \sum_{j=1}^{Nf} \left(B_{ij}^T\right)^{-1} \overline{\Theta}_j^T + \Psi_N \\ \sum_{j=1}^{Nf} \left[ \alpha_i^T \gamma_j^T - \left(B_{ij}^T\right)^{-1} \right] \overline{\Theta}_j^{T,n+1} = -\alpha_i^T F^T - \Psi_N \end{cases} \tag{A.25}$$

### A.1.3 Discontinuous Galerkin Finite Element Method

DGFEM is applied to the hyperbolic part of the transport equation, allowing to solve the advective field without oscillations and with very limited numerical diffusion.

$$\frac{\partial c}{\partial t} + \mathbf{u}\cdot\nabla c = 0 \tag{A.26}$$

rewritten in:

$$\frac{\partial c}{\partial t} = -\nabla\cdot(\mathbf{u}c) + c(\nabla\cdot\mathbf{u}) \tag{A.27}$$

Equation (A.27) is multiplied by a test function $\zeta$ and Green's theorem is used:

$$\int_T \frac{\partial c}{\partial t}\zeta\, dT + \int_T \mathbf{u}c\cdot\nabla\zeta\, dT - \sum_{i=1}^{Nf}\int_{Ei}\zeta c\mathbf{u}\cdot\mathbf{n}\, dE_i = \int_T c(\nabla\cdot\mathbf{u})\zeta\, dT \tag{A.28}$$

Due to the discontinuity of the concentration across the element edge, the convective flux is approximated by solving a 1D Riemann problem. The convective flux is equal to:

$$\begin{cases}\mathbf{u}c = \mathbf{u}c^{out} \text{ if } \mathbf{u}\cdot\mathbf{n} < 0\\ \mathbf{u}c = \mathbf{u}c^{in} \text{ if } \mathbf{u}\cdot\mathbf{n} \geq 0\end{cases} \tag{A.29}$$

The concentration is approximated by:

$$c^T(x,t) = \sum_{k=1}^{Nf}\omega_k^T(x)c_k^T(t) = \sum_{k=1}^{Ne}\omega_k^T(x)c_k^{T,in}(t) \tag{A.30}$$

where $c_k^T$ is the concentration at node $k$ of element $T$, $\omega_k^T$ are the Galerkin finite element test functions and $N_f$ is the number of edges of element $T$.
Using Equation (A.30) and (A.29) in (A.28) and replacing $\zeta$ by the test function $\omega_k^T$, with $k = 1, ..., N_f$ lead to:

$$\begin{gathered}\sum_{k=1}^{N_f}\frac{\partial c_k^T}{\partial t}\int_T \omega_k^T\omega_i^T\, dT = \sum_{k=1}^{N_f}[-c_k^T\int_T \omega_k^T\mathbf{u}\cdot\nabla\omega_i^T\, dT +\\ \sum_{j=1}^{N_f} c_k^{T,in/out}\int_{Ej}\omega_k^T\omega_i^T\mathbf{u}\cdot\mathbf{n}\, dE_j + c_k^T\int_T \omega_k^T(\nabla\cdot\mathbf{u})\omega_i^T\, dT]\end{gathered} \tag{A.31}$$

After inverting the local mass matrix (left hand side of Equation (A.31)), this system can be written in the following matrix form:

$$\frac{d\mathbf{c}^T}{dt} = M(\mathbf{c}^{T,in}, \mathbf{c}^{T,out}) \tag{A.32}$$

where $\mathbf{c}^T$ is a vector of dimension $N_f$, containing the cell unknowns $c_k^T$ and $M$ represents the components of the right hand side of Equation (A.31), multiplied by the inverse of the mass matrix. Note that $c_k^T$ and $c_k^{T,in}$ are the same.
Time discretization is second order accurate and based on a simplified explicit Runge-Kutta method. The following three steps algorithm is used:

1. Computation of an intermediate approximation $\mathbf{c}^{T,n+1/2}$ by

   $\mathbf{c}^{T,n+1/2} = \mathbf{c}^{T,n} + \Delta t/2 \;\; M(\mathbf{c}^{T,in,n}, \mathbf{c}^{T,in,n})$
   Note only the inner value of the nodal concentrations are used.

2. Calculation of $\widetilde{\mathbf{c}}^{T,n+1}$ by solving the Riemann problem

   $\widetilde{\mathbf{c}}^{T,n+1} = \mathbf{c}^{T,n} + \Delta t M(\mathbf{c}^{T,in,n+1/2}, \mathbf{c}^{T,out,n+1/2})$

3. The solution is stabilized with the multidimensional slope limiting operator L, which avoid that oscillations raise, for further details see [6]
   $\mathbf{c}^{T,n+1} = L(\widetilde{\mathbf{c}}^{T,n+1})$

Consider for the element $T$ the following notations:

- $\overline{c}^T$ is the average of $c$ over $T$.
- $min(i)$ is the minimum of the $\overline{\widetilde{c}^{T,n+1}}$ of all the elements having $i$ has a node.
- $max(i)$ is the minimum of $\overline{\widetilde{c}^{T,n+1}}$ of all the elements having $i$ for a node.
- $min(T)$ is the minimum of $\overline{\widetilde{c}^{T,n+1}}$ of all the elements, having a common node with the element $T$.
- $max(T)$ is the maximum of the $\overline{\widetilde{c}^{T,n+1}}$ of all the elements having a common node with the element $T$.

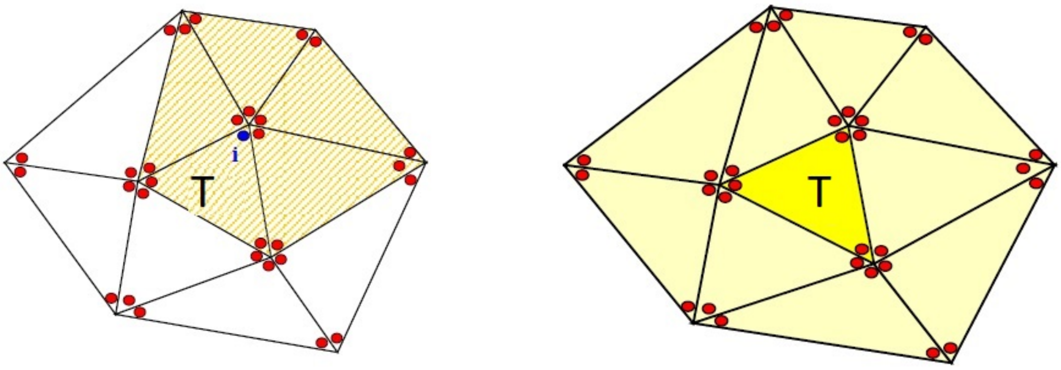


**Figure A.1:** Element T, node I and their neighborhood, from [6]

The multidimensional slope limiting process satisfies the following conditions:

$$\overline{c^T} = \overline{\widetilde{c}^T} \quad \text{in order to preserve mass balance.} \tag{A.33}$$

In order to avoid oscillations:

$$min(i) \leq \overline{\widetilde{c}^{T,n+1}} \leq max(i) \tag{A.34}$$

if $\overline{\tilde{c}^{T,n+1}}$ is a local maximum or minimum, then

$$c_i^{T,n+1} = \overline{c^T} \quad i = 1, ..., N_f \tag{A.35}$$

If $\overline{\tilde{c}^{T,n+1}}$ is not a local maximum or minimum, the concentration at node $i$ of element $T$, $c_i^{T,n+1}$, is not uniquely defined. Therefore, $c_i^{T,n+1}$ is imposed to be as close as possible to $\tilde{c}^{T,n+1}$ by minimizing the following objective function J:

$$J(p) = \sum_{i=1}^{Nf} \parallel c_i^{T,n+1} - \tilde{c}_i^{T,n+1} \parallel^2 \tag{A.36}$$

where $p$ represents the $N_f$ concentrations of the inner node of element $T$. The method used to minimize (A.36) with the constraint given by (A.34) and (A.33) is the saddle point method.
We remind the previous theoretical presentation, involving numerical method of *Traces* is completely taken from [6].

## A.2 Space Time Adaptation

In this Section, space-time grid adaptation procedure is implemented, in order to efficiently improve the accuracy of numerical solution. The method is said *adaptive*, because the process depends on previous results at all stages and it has to be interpreted as an enhancement of the classic finite element algorithm, described in Section 1.2.
An automatically adaptive grid is a very advisable tool in modelling real scale applications, in fact, often, the most relevant processes, driving the evolution of the variable, take place in different regions throughout the domain, therefore a homogeneous distribution could lead to useless computational cost.
Adaptation procedure is not a new concept in scientific literature, however the novelty of this work is to make spatial and time distribution grounded on the definition of an *anisotropic a posteriori* error estimator for the global discretization error.
About the choice of an *anisotropic* estimator, the main objective is to include geometrical information, regarding orientation and size, about a generic triangle $T$ in mesh $\mathcal{T}_h$. This may tailor to the discretization of ADE, in particular to concentration evolution within the domain. Furthermore, according to [14], using rather elongated elements may help in reducing the number of degrees of freedom necessary for a given solution accuracy.
About the choice of *a posteriori* estimator, it is important to remind that an analytical solution is not available for the Problem (1.20) at hand, therefore neither the error could be calculated. However, a posteriori estimator only relies on the discrete function and it could qualitatively detect which elements in the mesh give a larger error contribution. This could be perfectly coupled with a refinement, just selecting those triangles with a significant error and subdividing them, reducing the mesh size $h$.

The Section deals with a first presentation of anisotropic setting, introducing the main variables, thereafter the focus is shifted towards the definition of the spatial and time estimators. Finally, the adaptative procedure is explained and the algorithm, which has been implemented in [19], is described in all its steps. Notice that all the theoretical framework is taken from scientific literature.

### A.2.1 Anisotropic setting

Following [3], it is possible to derive the anisotropic information by introducing the standard invertible affine map $F_T : \hat{T} \to T$, which transforms the equilateral triangle $\hat{T}$, with vertices in coordinates:
(-$\sqrt{3}/2, -1/2$),($\sqrt{3}/2, -1/2$) and $(0, 0)$ and initially inscribed in a circle with unit radius centred at $(0, 0)$, into a generic $T$. The map changes the circle into an ellipse circumscribing $T$, as illustrated in Figure A.2.

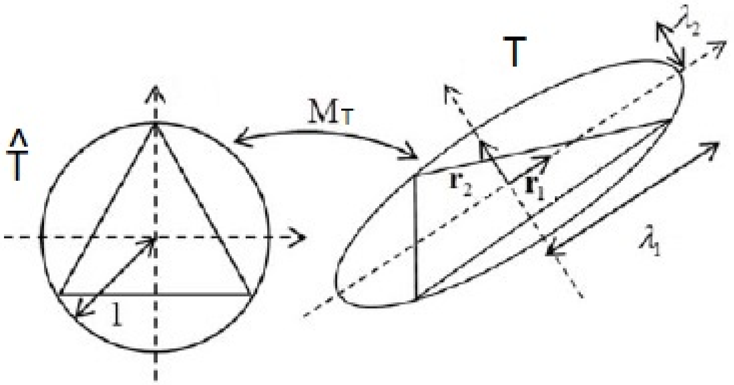


**Figure A.2:** Geometrical interpretation of the map $F_T$ from [3]

It is possible to provide the following definition:

$$\mathbf{x} = F_T(\hat{\mathbf{x}}) = \boldsymbol{M_T}\hat{\mathbf{x}} + \boldsymbol{f_T} \quad \forall \mathbf{x} = (x_1, x_2)^T \in T \tag{A.37}$$

with $\hat{\mathbf{x}} = (\hat{x_1}, \hat{x_2})^T \in \hat{T}, \boldsymbol{M_T} \in \mathbb{R}^{2\times2}$ the Jacobian associated with the map $F_T$ and $\boldsymbol{f_T} \in \mathbb{R}^2$ a shift vector. It is possible to provide a decomposition of $\boldsymbol{M_T}$ as:

$$\boldsymbol{M_T} = \boldsymbol{B_T}\boldsymbol{Z_T} \tag{A.38}$$

with $\boldsymbol{B_T} \in \mathbb{R}^{2\times2}$ a symmetric positive definite matrix and $\boldsymbol{Z_T} \in \mathbb{R}^{2\times2}$ an orthogonal matrix. We consider the following spectral decomposition of $\boldsymbol{B_T}$:

$$\boldsymbol{B_T} = \boldsymbol{R_T^T}\boldsymbol{\Lambda_T}\boldsymbol{R_T} \tag{A.39}$$

where

$$\boldsymbol{R_T} = \begin{bmatrix} \boldsymbol{r_{11}} & \boldsymbol{r_{21}} \\ \boldsymbol{r_{12}} & \boldsymbol{r_{22}} \end{bmatrix} \qquad \boldsymbol{\Lambda_T} = \begin{bmatrix} \lambda_{1,K} & 0 \\ 0 & \lambda_{2,K} \end{bmatrix}$$

in which $\boldsymbol{R_T} \in \mathbb{R}^{2\times2}$ is the matrix containing the eigenvectors of $\boldsymbol{B_T}$ and $\boldsymbol{\Lambda_T} \in \mathbb{R}^{2\times2}$ is the diagonal matrix of the corresponding eigenvalues, where it is assumed $\lambda_{1,T} \geq \lambda_{2,T}$. In particular, the right eigenvalues are expressed as $\boldsymbol{r_{1,T}} = (r_{11}, r_{12})$

and $\boldsymbol{r_{2,T}} = (r_{21}, r_{22})$. Notice shape, size and orientation of $T$ are fully described by $\boldsymbol{r_{i,T}}$ and $\lambda_{i,T}$ for $i = 1, 2$. In particular, $\boldsymbol{r_{1,T}}$ and $\boldsymbol{r_{2,T}}$ identify the directions of the two semi-axes of the ellipse circumscribing $T$, while $\lambda_{1,T}$ and $\lambda_{2,T}$ correspond to the measure of the semi-axes. All these information are represented in Figure A.2. Deformation of the triangle $T$ is provided by the aspect ratio $s_T = \lambda_{1,T}/\lambda_{2,T} \geq 1$ that provides an information about the deformation of the triangle $T$.

### A.2.2 Recovery-based error estimator

As previously introduced, mesh adaptivity is governed by the error estimator $\eta_{ht}$

$$\eta_{ht} = \eta_h + \eta_t \tag{A.40}$$

which includes a compound $\eta_h$ for spatial discretization and one, $\eta_t$, for time. Following [10], the two error estimates can be computed separately, entailing two steps:

- local reconstruction of the solution gradient, upon averaging or re-interpolating it;
- estimation of the discretization error, obtaining by computing the $L^2$-norm of the difference between the recovered and the discrete gradient.

#### Spatial error estimator

Consider a finite element approximation $z_h(t)$ of a generic scalar variable $z(t)$. Discretization error is defined as:

$$e_h^z(t) = z(t) - z_h(t) \tag{A.41}$$

while its $H^1$-seminorm is defined as:

$$|e_h^z(t)|^2_{H^1} = \int_\Omega |\nabla z(t) - \nabla z_h(t)|^2 \, d\Omega \tag{A.42}$$

However, the seminorm can not be calculated, since the exact gradient of the solution is unknown. Relying on an isotropic view, the general idea is to replace the exact gradient with a recovered one $P(\nabla z_h(t))$:

$$|e_h^z(t)|^2_{H^1} \simeq \int_\Omega |P(\nabla z_h(t)) - \nabla z_h(t)|^2 \, d\Omega = \eta(e_h^z(t)) \tag{A.43}$$

Define $\Delta_T$ as the patch of triangles, sharing at least one vertex with the given triangle $T$, the anisotropic error estimator should differ from the isotropic one, by involving geometrical quantities, indeed indentifying the size, shape and orientation of element $T$.

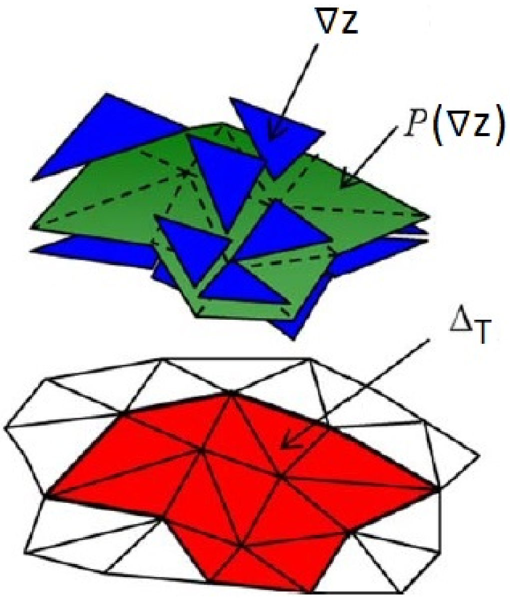


**Figure A.3:** Spatial recovered gradient procedure from [3]. Patch corresponds to the red area; numerical gradients are the blue pieces; recovered gradients are the green ones.

The local error estimator is obtained:

$$\begin{aligned}\eta^A_{T,h}(z_h(t))^2 = \frac{1}{\lambda_{1,T}\lambda_{2,T}} \int_{\Delta_T} \{\lambda^2_{1,T}\left[\mathbf{r}_{1,T} \cdot (P(\nabla z_h(t)) - \nabla z_h(t))\right]^2 + \\ \lambda^2_{2,T}\left[\mathbf{r}_{2,T} \cdot (P(\nabla z_h(t)) - \nabla z_h(t))\right]^2\} \, d\Delta_T \quad \forall T \in \mathcal{T}_h\end{aligned} \tag{A.44}$$

The factor $1/(\lambda_{1,T}\lambda_{2,T})$ in Equation (A.44) ensures the consistency with respect to isotropic case, in fact imposing $\lambda_{1,T} = \lambda_{2,T}$, the isotropic estimator is obtained. Following [31], the procedure to find the recovered gradient is reported below:

$$P_{\Delta_T}(\nabla z_h)(\mathbf{x}, t) = \frac{1}{|\Delta_T|} \sum_{T_i \in \Delta_T} |T| \nabla z_h(t) \quad \text{with } \mathbf{x} \in T_i \tag{A.45}$$

where $T_i$ represents a generic triangle, which belongs to the patch $\Delta_T$. Equation (A.45) represents the area-weighted average over the whole patch $\Delta_T$ of the gradients of the discrete solution and then such value is assigned to the single element $T$, as shown in Figure A.3. Thereafter for any $T_i \in \Delta_T$ with $T_i \neq T$, $P_{\Delta_{T_i}}(\nabla z_h)$ is assumed constant over the whole patch $\Delta_{T_i}$.

### Time error estimator

In this part an overview of the time discretization method is provided. Following [11], time step $\Delta t^n$, which defines the length of the interval $I_n$ between $t^n$ and $t^{n+1}$ should be determined. The prediction is carried out by defining a local time error estimator, instead of the global one. $\Delta t$ can not be predicted a priori, since the total number and length are not constant values, in fact both are defined by the adaptive procedure itself. Consider a general scalar variable $z$ and its finite element approximation $z_h$. The $H^1-$seminorm of the time discretization error on the interval $I_n$ is:

$$|e_h^z(\mathbf{x})|^2_{H^1} = \int \left| \frac{\partial z(\mathbf{x})}{\partial t} - \frac{\partial z_h(\mathbf{x})}{\partial t} \right|^2 dt \tag{A.46}$$

Firstly, dealing with the approximated derivative, at a generic time level $t^n$, $z_h$ at all previous time $t^i$, with $i = 0, ..., n-i$ is known. Since the solution on previous step has already been computed, it is possible to replace the derivatives term in Equation (A.46), by substituting $z_h$ with the straight line interpolating $z_h$ at $t^{n-1}$ and $t^n$ as showed in Figure A.5.
Therefore:

$$\frac{\partial z_h(\mathbf{x})}{\partial t} \simeq \frac{z_h^n - z_h^{n-1}}{\Delta t^{n-1}} \quad \text{with } z_h^n = z_h(\mathbf{x}, t^n) \tag{A.47}$$

Secondly, the exact derivative should be substituted and to do this, the solution $z(\mathbf{x})$ is replaced with a recovered value $z^R(\mathbf{x})$, which has been selected as the parabola interpolating the values $z_h^{n-2}, z_h^{n-1}, z_h^n$, in correspondence of time step $t^{n-2}, t^{n-1}, t^n$, as shown in Figure A.5.

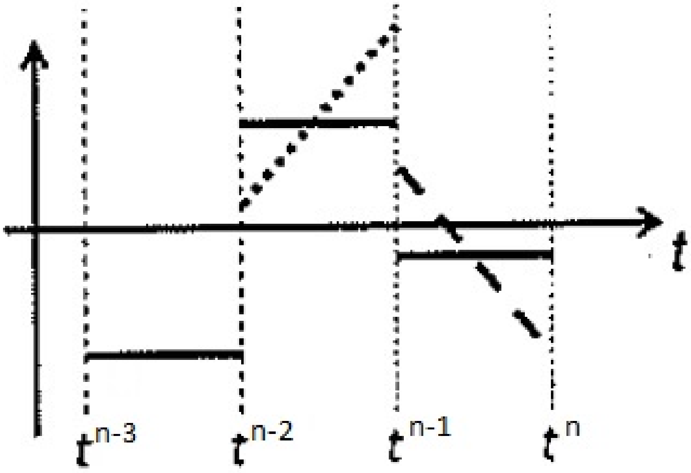


**Figure A.4:** Time derivatives from [11]; $\partial_t z^R$ (dotted and dashed line), $\partial_t z_h$ (continuous line).

Choosing a parabola to identify the recovered solution leads to a piecewise linear recovered gradient. According to [29], a more detailed formulation of $H^1-$seminorm of the time discretization error in Equation (A.46) could be estimated, by approximating via local time recovery-based error estimator $\eta_{n-1,t}(z_h(\mathbf{x}))$ as:

$$|e_h^z(\mathbf{x})|^2_{H^1} \simeq \pi \int \left| \frac{\partial z^R(\mathbf{x})}{\partial t} - \frac{z_h^n - z_h^{n-1}}{\Delta t^{n-1}} \right|^2 dt = \eta_{n-1,t}(z_h(\mathbf{x}))^2 \tag{A.48}$$

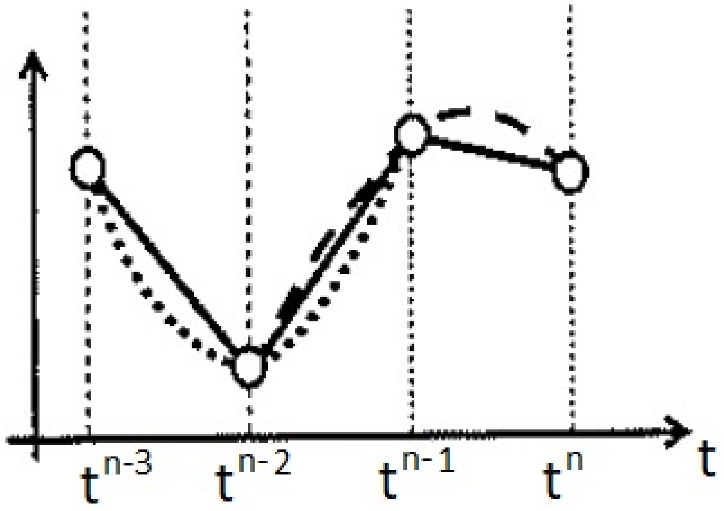


**Figure A.5:** Time gradient recovery from [11]. Recovered solution (dotted and dashed line) and linear interpolate values (continuous line).

with $\pi$ a suitable time scale factor of the problem. In hand, in order to make the estimator dimensionless, generally it coincides with the previous time step. $\eta_{n-1,t}(z_h(\mathbf{x}))$ is evaluated at each vertex of the triangle $T$, however a single averaged value for the interval $I^n$ is needed.

$$\eta_{n-1,T}(z_h)^2 = \frac{1}{3}\sum_{B_i \in T} \eta_{n-1,T}(z_h(B_i))^2 \tag{A.49}$$

where $B_i$ with $i = 1, 2, 3$ represents the vertex of the triangle $T$. Shifting from local to global level, once these values have been computed is possible to provide an area-weighted average:

$$\eta_{n-1}(z_h)^2 = \frac{1}{\sum_{T\in\mathcal{T}_h}|T|}\sum_{T\in\mathcal{T}_h}|T|\eta_{n-1,T}(z_h)^2 \tag{A.50}$$

Equation (A.50) normalises the time error with respect to the domain dimension. In view of the time adaptive scheme in Section A.2.3, the global time error estimator is introduced:

$$\eta_t(e_h^z)^2 = \sum_{n=1}^{Nstep} \eta_{n-1,t}(z_h)^2 \tag{A.51}$$

where $N_{step}$ identifies the total number of time intervals, but it is still unknown.

### A.2.3 Adaptation procedure

Until now the algorithm combines the discretization scheme of ADE with the information, provided by the error estimators, to automatically adapt the mesh $\mathcal{T}_h$ and time domain $[0, t_{end}]$. However, the aim is to guarantee the error below a certain value, therefore, following [10], the accuracy target needs to be specified:

$$\tau = \tau_{\Delta t} + \tau_h \tag{A.52}$$

The former drives the mesh adaptation, while the latter the time advancement.

### Spatial mesh adaptivity

Following [14], the adaptive mesh is built starting from a metric, induced by the anisotropic error estimator, labelled as $\eta_h^A(e_h^z(t))$, so the number of mesh elements is minimized and the tolerance $\tau_h$ is guaranteed on $\eta_h^A(e_h^z(t))$ via an error equidistribution criterion. Before going on with the description, the concept of *metric* is introduced. It is a symmetric positive definite tensor field, $\mathcal{M} : \Omega \to \mathbb{R}^{2\times 2}$, such that:

$$\mathcal{M}(\mathbf{x}) = \tilde{\boldsymbol{R}}^T(\mathbf{x})\tilde{\boldsymbol{\Lambda}}^{-2}(\mathbf{x})\tilde{\boldsymbol{R}}(\mathbf{x}) \quad \text{for any } \mathbf{x} \in \Omega \tag{A.53}$$

with $\tilde{\boldsymbol{\Lambda}}(\mathbf{x}) = \text{diag}(\tilde{\lambda}_1(\mathbf{x}), \tilde{\lambda}_2(\mathbf{x}))$ and $\tilde{\boldsymbol{R}}^T(\mathbf{x}) = [\tilde{\mathbf{r}}_1(\mathbf{x}), \tilde{\mathbf{r}}_2(\mathbf{x})]$ a positive diagonal and orthogonal tensor. It is possible to associate a piecewise constant metric $\mathcal{M}$ with a background grid $\mathcal{T}_h$, by exploiting the anisotropic spatial error estimator, previously defined in Equation (A.44), therefore to evaluate it on $\mathcal{T}_h$. The computation of the new elemental metric, which is indicated by the symbol $\mathcal{M}_T^{n+1}$, where $n+1$ stands for the successive time step, is performed on the background grid $\mathcal{T}_h^n$ by:

1. applying an error equidistribution criterion;
2. solving a local constrained optimization problem on each element $T$ of $\mathcal{T}_h^n$.

In details, firts the ADE is solved on the background grid, then the unknown metric $\mathcal{M}_T^{n+1}$ is computed by imposing the desired accuracy $\tau_h$ and by equidistributing the error. In this way, local accuracy $\tau_T = \tau_h/\text{card}(\mathcal{T}_h)$ is also guaranteed. This procedure essentially leads to deal with local constrained optimization problems, which can be explicitly solved in [14], [32], [3]. The general idea is to minimize the mesh elements, by maximising the area of each triangle $T$. Reporting and analysing the whole minimisation problem is not a matter of this thesis work, for further details refer to the suggested scientific literature. Only the solution is reported. An optimal aspect ratio, label as $\tilde{s}_T^\eta$, can be derived, which also drives the definition of the optimal metric $\mathcal{M}^\eta$, completely identified by two separate values $\tilde{\lambda}_{1,T}^\eta$ and $\tilde{\lambda}_{2,T}^\eta$. The algorithm allows to minimize the number of elements, always maintaining a certain value of accuracy, defined by $\tau_h$
Once the optimal metric has been derived, the new adapted mesh ensues. However some constraints are necessary, in order to avoid a cluster of elements with small surface area. The following input parameters are set:

$p_{min}$ : minimum area value allowed for $T$. Therefore the actual triangle area will be $min\{\tilde{\lambda}_{1,T}^\eta\, \tilde{\lambda}_{2,T}^\eta,\, p_{min}\}$;

$N_{min}$, $N_{max}$ : maximum and minimum number of elements in $\mathcal{T}_h$. The initial number of elements is first found by means of optimal metric and if the number does not belong to the interval, a new metric is generated via a global and uniform scaling of the tensor $\mathcal{M}$;

$h_{min}$, $h_{max}$ : maximum and minimum triangle edges length.

### Time step adaptivity

Time step $\Delta t$ might be predicted, however the total number of intervals is an unknown and it could be determined only a posteriori. A local tolerance $\tau_t$ is needed to be imposed at each interval, instead of considering a global value. Roughly speaking, time step adaptation entails three actions:

1. computation of a recovered time derivative;
2. evaluation of the local time estimator;
3. prediction of the next time step size.

Time error estimator in Equation (A.50) can be rewritten as:

$$\eta_{n-1}(z_h)^2 = \Delta t^{n-1} \varepsilon_{n-1}(z_h)^2 \tag{A.54}$$

where

$$\varepsilon_{n-1}(z_h)^2 = \frac{1}{(\Delta t^{n-1})^2 \sum_{T \in \mathcal{T}_h^n} |T|} \sum_{T \in \mathcal{T}_h^n} |T| \eta_{n-1,T}(z_h)^2 \tag{A.55}$$

where $\mathcal{T}_h^n$ denotes the mesh at time instant $t^n$. Finally, it is possible to compute the actual time step:

$$\Delta t^n = \frac{\eta_{n-1}(z_h)}{\varepsilon_{n-1}(z_h)} \tag{A.56}$$

Further constraints are needed in order to control the time adaptation, therefore the following parameters have been set:

$\Delta t_{min}$ : minimum allowed time step for the adaptation procedure.

$\Delta t_{max}$ : maximum allowed time step for the adaptation procedure.

Both values have been chosen accordingly to the characteristic time scale of the given transport problem.

# Appendix B

# Appendix B

## B.1 Transport problem

Recall conservative strong formulation of ADE:

$$\frac{\partial c}{\partial t} - \nabla \cdot (\mathbf{D}\nabla c - \mathbf{u}c) = 0 \tag{B.1}$$

Consider functional space $\mathcal{Z}$ and test function $\zeta \in \mathcal{Z}$, described in Section 1.2.2. Weak formulation of previous equation reads:

$$\int_\Omega \left[ \frac{\partial c}{\partial t}\zeta - \zeta \nabla \cdot (\mathbf{D}\nabla c - \mathbf{u}c) \right] d\Omega = 0 \tag{B.2}$$

Apply Green theorem, it is obtained:

$$\begin{aligned} &\int_\Omega \left[ \zeta \nabla \cdot (\mathbf{D}\nabla c - \mathbf{u}c) \right] d\Omega \\ = &\int_\Gamma \zeta (\mathbf{D}\nabla c - \mathbf{u}c) \cdot \mathbf{n}\, d\Gamma - \int_\Omega (\mathbf{D}\nabla c - \mathbf{u}c) \cdot \nabla\zeta \, d\Omega \end{aligned} \tag{B.3}$$

Linear integral term could be decomposed into four parts, which are label as $\Gamma_i = \partial\Omega_i$, representing each border of the control volume. However only the north section is characterized by an outlet flow, as a consequence of impermeable walls impositions. Furthermore, since natural boundary condition holds, it ensues:

$$\int_\Gamma \zeta \cdot (\mathbf{D}\nabla c - \mathbf{u}c) \cdot \mathbf{n} d\Gamma = \int_{\Gamma_{north}} \zeta \mathbf{u} \cdot \mathbf{n}\, c\, d\Gamma \tag{B.4}$$

Galerkin discrete approximation immediately derives:
find $c_h = c_h(t) \in \mathcal{Z}_h$ such that for any $\zeta_h \in \mathcal{Z}_h$:

$$\sum_{T \in \mathcal{T}_h} \left[ \int_T \left( \frac{\partial c_h}{\partial t}\zeta_h + \nabla\zeta_h \cdot (\mathbf{D}_h \nabla c_h - \mathbf{u}_h c_h) \right) dT \right] = 0 \tag{B.5}$$

Final step is the time discretization, reported in Section 1.2.2.

## B.2 False Diffusion

Analytical demonstration, dealing with the origin of numerical diffusion is here described. We consider, for sake of simplicity, 1D ADE with uniform velocity field $u$ along $x$ direction and constant dispersion $D = \alpha_L \, u$, reported in Equation (B.6)

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} - u \frac{\partial c}{\partial x} \tag{B.6}$$

First space derivative (convective term) is commonly discretized through finite difference of the incremental ratio, therefore we consider two nodes, the first located at $x$ and the other at $x + \Delta x$, so:

$$\frac{\partial c}{\partial x} = \frac{c_{x+\Delta x} - c_x}{\Delta x} \tag{B.7}$$

however, we consider to apply Taylor series up to the second order

$$c_{x+\Delta x} = c_x + \frac{\partial c}{\partial x} \Delta x + \frac{\partial^2 c}{\partial x^2} \frac{\Delta x^2}{2} \tag{B.8}$$

Equation (B.6) could be rearranged, by declaring the term $\partial_x c$ from Equation (B.8). Collecting all the terms, in order to rebuild the new form of ADE equation, it is obtained:

$$\frac{\partial c}{\partial t} = \left( D + \frac{\Delta x}{2} u \right) \frac{\partial^2 c}{\partial x^2} - u \left( \frac{c_{x+\Delta x} - c_x}{\Delta x} \right) \tag{B.9}$$

Comparing Equation (B.6) and (B.9), it is noticed an extra term, $u\Delta x/2$, is present, which does not have any physical meaning, but it simply comes from the discretization of the advective component on the dispersive term. It is inferred in order to reduce the impact of the numerical (or false) diffusion, the following constraint has to be respected:

$$u \frac{\Delta x}{2} << D \tag{B.10}$$

which can be also rewritten as $Pe_h << 2$ or $\Delta x << \alpha_L$, if we assume that $D = u \, \alpha_L$, with $\alpha_L$ the longitudinal dispersivity.

# List of Figures

# List of Tables

# List of Symbols

### Notation

$\mathbf{u} = \{u_i\}$ with $\mathbf{u} \in \mathbb{R}^n$ and $i = 1, ..., n$

$\mathbf{D} = \{D_{ij}\}$ with $\mathbf{D} \in \mathbb{R}^{n \times m}$ and $i = 1, ..., n$ and $j = 1, ..., m$

$\nabla z = \sum_{i=1}^{n} \partial_{xi}\, z$ with $z : \Omega \mapsto \mathbb{R}$

$\nabla \cdot \mathbf{z} = \sum_{i=1}^{n} \partial_{xi}\, z_i$ with $z : \Omega \mapsto \mathbb{R}^n$

$\overline{z} = \sum_{i=1}^{n} z_i / n$

### Meshes and geometrical features

| | | |
|---|---|---|
| $\Omega$ | Domain on which a PDE is defined | |
| $\Omega_h$ | Approximated domain | |
| $\|\Omega\|$ | Volume of the domain $\Omega$ | $[m^3]$ |
| $l$ | Correlation length | $[m]$ |
| $\Gamma$ | Boundary of the domain, $\Gamma = \partial\Omega$ | |
| $h$ | Mesh size (*i.e.* diameter of triangle) | |
| $\mathcal{T}_h$ | Triangulation (*i.e.* the set of triangles) | |
| $\mathcal{T}_h^n$ | Triangulation at time instant $t^n$ | |
| $\mathcal{N}$ | Set of mesh vertices | |
| $T$ | Generic triangle in $\mathcal{T}_h$ | |
| $\|T\|$ | Triangle area | $[m^2]$ |
| $E_i$ | $i^{th}$ edge of triangle $T$ | |
| $\|E_i\|$ | Length of $E_i$ | $[m]$ |
| $V_i$ | $i^{th}$ vertex of triangle $T$ | |
| $N_t$ | Number of triangles in $\mathcal{T}_h$ | |

| | |
|---|---|
| $N_e$ | Number of edges in $\mathcal{T}_h$ |
| $N_v$ | Number of vertices in $\mathcal{T}_h$ |
| $N_f$ | Number of faces of $T$ (for triangle $N_f = 3$) |
| $\Delta_i$ | Patch of the elements sharing node $i$ |
| $\Delta_T$ | Patch of the elements sharing at least a vertex with $T$ |

### Thermo-physical fluid properties

| | | |
|---|---|---|
| $\rho$ | Fluid density | $[kg/m^3]$ |
| $\mu$ | Fluid viscosity | $[Pa\,s]$ |

### Flow and Transport variables

| | | |
|---|---|---|
| $\mathbf{g}$ | Gravitational acceleration | $[m/s^2]$ |
| $\mathbf{u}$ | Velocity vector | $[m/s]$ |
| $\mathbf{q}$ | Volumetric flow rate per unit area | $[m/s]$ |
| $P$ | Pressure | $[Pa]$ |
| $L$ | Characteristic lenght | $[m]$ |
| $k$ | Permeability | $[m^2]$ |
| $K_H$ | Hydraulic conductivity | $[m/s]$ |
| $\varphi$ | Porosity | |
| $A$ | Cross sectional area | $[m^2]$ |
| $S_w$ | Water saturation | |
| $c$ | Concentration | $[mol/m^3]$ |
| $c_0$ | Initial concentration | $[mol/m^3]$ |
| $c_D$ | Dirichlet concentration, imposed on a boundary | $[mol/m^3]$ |
| $M$ | Mass | $[kg\,or\,mol]$ |
| $\mathbf{D}$ | Dispersion-Diffusion tensor | $[m^2/s]$ |
| $\alpha_L$ | Longitudinal dispersivity | $[m]$ |
| $\alpha_T$ | Transverse dispersivity | $[m]$ |
| $D_m$ | Molecular diffusion | $[m^2/s]$ |

### Finite element space

| | |
|---|---|
| $H^k$ | Hilbert space of order $k$ |
| $\mathcal{V}$, $\mathcal{W}\|_\Omega$ | Finite dimensional spaces on $\Omega$ |
| $\mathcal{X}$, $\mathcal{Z}\|_\Omega$ | Finite dimensional spaces on $\Omega$ |
| $\mathcal{P}_r(\Omega)$ | Space of polynomials of degree up to and including $r$ on $\Omega$ |
| $(\mathcal{P}_q(\Omega))^n$ | Vectorial space of dimension $n$ |
| $z$ | Generic scalar function |
| $z_h$ | Finite element approximation of a generic scalar variable $z$ |
| $w_h$, $\zeta_h$, $\xi_h$ | Test functions |
| $e_h^z$ | Discretization error |
| $\varpi_i$ | Lagrangian basis function, associated to node $i$ |
| $\Phi_i$ | Lagrangian elementary function, associated to node $i$ |
| $z_i$ | $i^{th}$ degree of freedom |
| $Z$ | Set of $z_i$ degree of freedom |

### Adaptive theory

| | |
|---|---|
| $F_T$ | Standard invertible affine map |
| $\mathbf{r}$ | Eigenvector |
| $\lambda$ | Eigenvalue of the associated eigenvector |
| $\tilde{\lambda}^\eta_{1,T}$ | Optimal eigenvalue |
| $\tilde{\lambda}^\eta_{2,T}$ | Optimal eigenvalue |
| $\tilde{\mathbf{r}}^\eta_{1,T}$ | Optimal direction |
| $s_T$ | Aspect ratio |
| $\eta_h^{type}(e_h^z(t))$ | Spatial estimator of the discretization error, where $type = A/I$ represents anisotropic or isotropic case |
| $\eta_t$ | Time estimator of the discretization error |
| $\mathcal{M}$ | Metric |
| $\tau_h$ | Spatial tolerance of error estimator |
| $\tau_T$ | Local spatial tolerance |

| | |
|---|---|
| $\tau_t$ | Time tolerance of error estimator |
| $p_{min}$ | Minimum allowed area for triangle $T$ |
| $h_{min}$ | Shortest triangle edge |
| $h_{max}$ | Longest triangle edge |
| $N_{min}$ | Minimum number of elements in $\mathcal{T}_h$ |
| $N_{max}$ | Maximum number of elements in $\mathcal{T}_h$ |
| $\Delta t$ | Time step |
| $\Delta t_{max}$ | Maximum allowed time step |
| $\Delta t_{min}$ | Minimum allowed time step |

# Acronyms

***Freefem*** FreeFem++ FreeFem++ is a high level integrated development environment for numerically solving partial differential equations.

***Traces*** Traces Traces (Transport of RadioActive Elements in Subsurface) is a computer programme for simulation of flow and reactive transport in saturated porous media. It is written in FORTRAN 95 language and is independent of any specific platform.

**FEM** Finite Element Method

**FVM** Finite Volume Method

**PDE** Partial Differential Equation

**DGFEM** Discontinuos Galerkin Finite Element Method

**MHFEM** Mixed Hybrid Finite Element

**BTC** Breakthrough curve

# Bibliography

[1] Martin J. Blunt. *Reservoir Engineering.* 2010. ISBN: 0444416676 (cit. on pp. vii, 1, 6, 7, 9, 11, 12, 19).

[2] Jacob Bear. *Dynamics of Fluids in Porous Media.* Vol. 120. 2. 1972, pp. 162–163. ISBN: 0486656756. DOI: 10.1097/00010694-197508000-00022. URL: http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage{\&}an=00010694-197508000-00022{\%}5Cnhttp://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage{\&}an=00010694-197508000-00022{\%}5Cnhttp://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=int (cit. on pp. vii, 1, 9, 11, 19).

[3] Bahman Esfandiar et al. "Anisotropic mesh and time step adaptivity for solute transport modeling in porous media". In: *SEMA SIMAI Springer Series.* Vol. 5. 2015, pp. 231–260. ISBN: 9783319060521. DOI: 10.1007/978-3-319-06053-8_12 (cit. on pp. vii, 2, 18–21, 33, 122, 124, 127).

[4] Simona Perotto and Luca Formaggia. *New Challenges in Grid Generation and Adaptivity for Scientific Computing.* Springer Publishing Company, Incorporated, 2015. ISBN: 331906052X, 9783319060521 (cit. on pp. vii, 2).

[5] G. M. Porta, S. Perotto, and F. Ballio. "A space-time adaptation scheme for unsteady shallow water problems". In: *Mathematics and Computers in Simulation* 82.12 (2012), pp. 2929–2950. ISSN: 03784754. DOI: 10.1016/j.matcom.2011.06.004. URL: http://dx.doi.org/10.1016/j.matcom.2011.06.004 (cit. on pp. vii, 2, 21, 76).

[6] P Ackerer H. Hoteit. *Traces user's guide.* 2004 (cit. on pp. viii, 115, 120, 121).

[7] Irina Gaus et al. "Geochemical and solute transport modelling for CO2 storage , what to expect from it ? To cite this version : HAL Id : hal-00568197". In: *International Journal of Greenhouse Gas Control, Elsevier, 2008, 2 (4), pp.605-625* (2011) (cit. on p. 1).

[8] Benito M Chen-Charpentier. "Numerical Methods for Flow and Transport in Porous Media". In: (2014). DOI: 10.1007/978-94-017-1114-2 (cit. on p. 1).

[9] Aronne Dell'Oca et al. *Space-time mesh adaptation for solute transport in randomly heterogeneous porous media.* 2017. DOI: 10.1016/j.jconhyd.2017.07.001 (cit. on pp. 2, 9, 21, 33).

[10] S Micheletti and Simona Perotto. "Anisotropic mesh adaption for time-dependent problems". In: *International Journal for Numerical Methods in Fluids* 58.9 (2008), pp. 1009–1015. ISSN: 02712091. DOI: 10.1002/fld.1597 (cit. on pp. 2, 21, 123, 126).

[11] Bahman Esfandiar et al. “Impact of space-time mesh adaptation on solute transport modeling in porous media”. In: *Water Resources Research* 51.2 (2015), pp. 1315–1332. ISSN: 19447973. DOI: 10.1002/2014WR016569. arXiv: 2014WR016527 [10.1002] (cit. on pp. 2, 21, 33, 124–126).

[12] W Cao, W Huang, and R D Russell. “A moving mesh method based on the geometric conservation law”. In: *SIAM J. Sci. Comput.* 24.1 (2002), pp. 118–142. ISSN: 10648275. DOI: 10.1137/S1064827501384925 (cit. on p. 2).

[13] L. Formaggia and S. Perotto. “New anisotropic a priori error estimates”. In: *Numerische Mathematik* 89.4 (2001), pp. 641–667. ISSN: 0029599X. DOI: 10.1007/s002110100273 (cit. on pp. 2, 21).

[14] L Formaggia and S Perotto. “Anisotropic error estimates for elliptic problems”. In: *Numerische Mathematik* (2003), pp. 67–92 (cit. on pp. 2, 21, 121, 127).

[15] S Micheletti and S Perotto. “Anisotropic adaptation via Zienkiewicz-Zhu error estimator for 2D Elliptic problems”. In: (2013), pp. 645–653. ISSN: 14397358. DOI: 10.1007/978-3-642-33134-3. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/978-3-642-33134-3 (cit. on pp. 2, 21).

[16] C Scudeler, M Putti, and C Paniconi. “Advances in Water Resources Mass-conservative reconstruction of Galerkin velocity fields for transport simulations”. In: 94 (2016), pp. 470–485. DOI: 10.1016/j.advwatres.2016.06.011 (cit. on p. 2).

[17] Adrien Berchet, Anthony Beaudoin, and Serge Huberson. “Comptes Rendus Mecanique Divergence-free condition in transport simulation”. In: 344 (2016), pp. 642–648. DOI: 10.1016/j.crme.2016.02.007 (cit. on p. 2).

[18] Alfio Quarteroni. *Numerical models for differential problems*. MS&A. Milano: Springer, 2009. URL: https://cds.cern.ch/record/1639539 (cit. on pp. 5, 16, 18–20).

[19] Mazzalovo. “Space-time adaptation technique for solute transport heterogeneous porous media”. In: (2017) (cit. on pp. 6, 9, 10, 21, 33, 58, 59, 122).

[20] John D Anderson. *Computational Fluid Dynamics: The Basics with Applications*. 1995. DOI: 10.1017/CBO9780511780066. arXiv: arXiv:1011.1669v3. URL: http://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=intitle:Computational+fluid+dynamics.+The+basics+with+applications{\#}6{\%}5Cnhttp://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=intitle:Computational+Fluid+Dynamics:+The+Basics+with+Applications.+1995{\%}23 (cit. on p. 8).

[21] J Bear and Y Bachmat. “Macroscopic modelling of transport phenomena in porous media. 2: Applications to mass, momentum and energy transport”. In: *Transport in Porous Media* 1 (1986), pp. 241–269. ISSN: 01693913. DOI: 10.1007/BF00238182. URL: http://link.springer.com/article/10.1007/BF00238182 (cit. on p. 9).

[22] Kambiz Vafai. *Handbook of porous media*. second edi. ISBN: 9780824727475 (cit. on p. 11).

[23] Robert C. Kirby et al. “Automated Solution of Differential Equations by the Finite Element Method”. In: (2012), pp. 95–119. ISSN: 15710661. DOI: 10.1007/978-3-642-23099-8. URL: http://link.springer.com/chapter/10.1007/978-3-642-23099-8{_}3 (cit. on pp. 13, 16).

[24] P. L. George. *Automatic mesh generation and finite element computation*. 1996. DOI: 10.1016/S1570-8659(96)80003-2 (cit. on pp. 14, 15).

[25] Eric Sonnendr and Ahmed Ratnani. “Advanced Finite Element Methods”. In: (2016) (cit. on p. 16).

[26] Franco Brezzi and Michel Fortin. *Mixed and Hybrid Finite Element Methods*. 1991 (cit. on p. 18).

[27] Dmitri Kuzmin. *A Guide to Numerical Methods for Transport Equations*. 2010, pp. 5–8. ISBN: 978-3-642-11639-1. DOI: 10.1007/978-3-642-11640-7_3. URL: http://www.mathematik.uni-dortmund.de/{\%}7B{~}{\%}7Dkuzmin/Transport.pdf (cit. on p. 20).

[28] Alexander N Brooks and Thomas J R Hughes. “Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations”. In: *Computer Methods in Applied Mechanics and Engineering* 32.1-3 (1982), pp. 199–259. ISSN: 0374-2830. DOI: 10.1016/0045-7825(82)90071-8 (cit. on p. 20).

[29] P E Farrell, S Micheletti, and S Perotto. “An anisotropic Zienkiewicz-Zhu-type error estimator for 3D applications”. In: *International Journal for Numerical Methods in Engineering* 85.6 (2011), pp. 671–692. ISSN: 00295981. DOI: 10.1002/nme.2980. arXiv: 1010.1724 (cit. on pp. 21, 125).

[30] Giovanni Michele Porta, Simona Perotto, and F. Ballio. “Anisotropic mesh adaptation driven by a recovery-based error estimator for shallow water flow modelling”. In: *International Journal for Numerical Methods in Fluids* October 2007 (2008), pp. 601–629. ISSN: 02712091. DOI: 10.1002/fld. arXiv: fld.1 [DOI: 10.1002] (cit. on pp. 21, 24).

[31] S Perotto S Micheletti. *Anisotropic adaptation via a Zienkiewicz–Zhu error estimator for 2D elliptic problems*. 2010 (cit. on pp. 21, 124).

[32] Stefano Micheletti and Simona Perotto. “Reliability and efficiency of an anisotropic Zienkiewicz-Zhu error estimator”. In: *Computer Methods in Applied Mechanics and Engineering* 195.9-12 (2006), pp. 799–835. ISSN: 00457825. DOI: 10.1016/j.cma.2005.02.009 (cit. on pp. 21, 127).

[33] F Hecht. “BAMG : Bidimensional anisotropic mesh generator”. In: February (2015) (cit. on p. 22).

[34] F Hecht. “New development in freefem+”. In: *Journal of Numerical Mathematics* 20.3-4 (2012), pp. 251–265. ISSN: 15702820. DOI: 10.1515/jnum-2012-0013 (cit. on pp. 25, 26).

[35] Isaac Amidror. “Scattered data interpolation methods for electronic imaging systems : a survey”. In: 11.April (2002), pp. 157–176. DOI: 10.1117/1.1455013 (cit. on p. 26).

[36] R R Yadav and Dilip Kumar Jaiswal. "Two-dimensional analytical solutions for point source contaminants transport in semi-infinite homogeneous porous medium". In: 6.4 (2011), pp. 459–468 (cit. on p. 30).

[37] H Hoteit et al. "The maximum principle violations of the mixed-hybrid finite-element method applied to diffusion equations". In: *International Journal for Numerical Methods in Engineering* 55.12 (2002), pp. 1373–1390. ISSN: 00295981. DOI: 10.1002/nme.531 (cit. on pp. 31, 32).

[38] Ishmail B Celik et al. "Procedure for estimation and reporting of uncertainty due to discretization in {CFD} applications". In: *Journal of fluids {Engineering-Transactions} of the {ASME}* 130.7 (2008). DOI: 10.1115/1.2960953. URL: http://dx.doi.org/10.1115/1.2960953 (cit. on p. 58).