

Edel Sofie Ulvåg Rikardsen

# Lagrangian methods and density estimation for advection-diffusion problems

Master's thesis in Applied Physics and Mathematics

Supervisor: Tor Nordam

February 2021



Edel Sofie Ulvåg Rikardsen

# **Lagrangian methods and density estimation for advection-diffusion problems**

Master's thesis in Applied Physics and Mathematics

Supervisor: Tor Nordam

February 2021

Norwegian University of Science and Technology

Faculty of Natural Sciences

Department of Physics



Norwegian University of  
Science and Technology



TFY4900 Physics, Master's Thesis

# Lagrangian methods and density estimation for advection-diffusion problems

Edel Sofie Ulvåg Rikardsen, February 2021  
Norwegian University of Science and Technology  
Faculty of Natural Sciences  
Department of Physics

**Supervisor:**  
Tor Nordam, Department of Physics

# Preface

This thesis was conducted at the Department of Physics at the Norwegian University of Science and Technology (NTNU) in Trondheim. It has been written to fulfill the graduation requirements of the 5 years Master's Degree Programme Applied Physics and Mathematics. I have been engaged in researching and writing this thesis from September 2020 to February 2021.

I would like to thank my supervisor, Tor Nordam, for guidance, support and expertise that greatly assisted this work. I offer my sincere appreciation for the learning opportunity provided by him.

# Abstract

In this thesis, we investigate two methods for numerical transport in the periodic double gyre system presented in Shadden et al. (2005), by solving an advection-diffusion problem using both an Eulerian and a Lagrangian formulation. The two-dimensional advection-diffusion equation is solved directly using the Crank-Nicolson finite difference method on a uniform grid, a so-called Eulerian fluid method. The diffusion coefficient is set to be constant in all systems, and to avoid oscillating solutions of the Crank-Nicolson scheme related to advection-dominated systems; we are forced to apply a relatively high diffusivity, with maximum cell Péclet numbers in the range  $Pe_{cell} \in \{0.16, 3.24\}$ . We present the Lagrangian particle method by deriving and solving the stochastic differential equation whose Fokker-Planck equation is equivalent to the advection-diffusion equation for smooth velocity and diffusion functions. By first applying Monte-Carlo techniques to achieve a discrete set of solutions, we estimate the probability density by applying a kernel density estimator. We demonstrate that the Lagrangian method produces the same results as the Eulerian fluid method for the double gyre system with constant diffusion. We define the optimal kernel bandwidth by minimizing the integrated squared error relative to a high-resolution Eulerian solution. The optimal bandwidth is found to decrease with the number of Lagrangian particles, and the relationship is estimated using the function form  $\Delta_{optimal} \sim 1/N^b$ . The parameter  $b$  is estimated to be  $0.133 \pm 0.070$  throughout the integration time of  $T = 10$ , which also was the period of the time-varying velocity field. The optimal bandwidth is found to have a clear increasing trend with diffusivity, as expected. The optimal bandwidth is finally found to have an ambiguous relation relative to time, and it is discussed whether the behavior might be related to the periodicity of the flow field. It is suggested that future work ought to include higher resolution Eulerian grids, lower diffusion constants, and system boundaries that have a smaller effect on the solution.

# Sammendrag

Denne oppgaven undersøker numerisk transport i to dimensjoner i et periodisk dobbel-vortex strømningsfelt som presentert i Shadden et al. (2005) med konstant diffusjon, gjennom å benytte både eulerske og lagrangske metoder. Adveksjon-diffusjonligningen er løst direkte gjennom å bruke den endelige differansemetoden Crank-Nicolson på en uniform grid, referert til som en eulersk fluid metode. Bruk av sentrale endelige differansemetoder for å løse adveksjonsdominerte systemer involverer ofte problemer relatert til oscillerende løsninger. For å unngå negative verdier i løsningen blir det benyttet høyere diffusjon for å senke det maksimale Péclet-tallet per gridcelle, som blir brukt i området  $\{0.16, 3.24\}$ . Den lagrangske partikkelmetoden er presentert gjennom å utlede en stokastisk differensialligning hvis Fokker-Planck ligning er lik adveksjon-diffusjonligningen dersom både diffusjonen og strømningsfeltet er glatt. Det er benyttet en Monte Carlo teknikk for å løse den stokastiske differensialligningen gjentatte ganger, hvor løsningene kan sees på som et stokastisk utvalg fra distribusjonen. Kernel density estimering (Kernel-tettheter) er benyttet for å estimere og konstruere fordelingen utvalget kommer fra. Partikkelmetoden gir likt resultat som den direkte eulerske metoden for adveksjon-diffusjonligningen i dobbel-vortex systemet med konstant diffusjon. En optimal båndbredde på kernel-funksjonen er definert som den båndbredden som minimerer integrert kvadratisk feil relativt til en eulerk løsning med høy oppløsning. Det er funnet at den optimale båndbredden er synkende for økende antall partikler, og forholdet mellom båndbredde og partikler er estimert ved kurvetilpasning til funksjonen  $\Delta_{optimal} \sim 1/N^b$ . Parameteren  $b$  er estimert til  $0.133 \pm 0.070$  gjennom hele integrasjonstiden  $T = 10$  (som også er perioden til strømningsfeltet). Den optimale båndbredden viser en klar økning som funksjon av diffusjon, men viser tvetydighet som funksjon av tid, men med en økende trend gjennom simuleringstiden. Det diskuteres hvorvidt resultatet kan være påvirket av strømningsfeltets periodiske oppførsel. Det trekkes frem at fremtidig arbeid bør inkludere høyere oppløsning av den eulerske gridden og lavere diffusjon, for dermed å senke Péclet-tall per celle. Samtidig bør andre grensebetingelser vurderes, eventuelt også undersøke større simuleringområder for å hindre den sterke påvirkning fra grensene.

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	EULERIAN AND LAGRANGIAN TRANSPORT MODELS . . . . .	1
1.1.1	Analytical double gyre flow field . . . . .	2
1.2	OBJECTIVES . . . . .	5
1.3	OUTLINE . . . . .	5
<b>2</b>	<b>THE ADVECTION-DIFFUSION PROBLEM</b>	<b>6</b>
2.1	TRANSPORT PHENOMENA . . . . .	6
2.1.1	The Advection-Diffusion Equation . . . . .	7
2.1.2	Eulerian formulation of solving advection-diffusion problems . . . . .	8
2.2	STOCHASTIC DIFFERENTIAL EQUATIONS . . . . .	9
2.2.1	Introduction to stochastic processes . . . . .	9
2.2.2	Brownian Motion and the Wiener Process . . . . .	11
2.2.3	Stochastic differential equations . . . . .	13
2.2.4	The Fokker-Planck Equation . . . . .	16
2.3	TIME DISCRETE APPROXIMATIONS . . . . .	18
2.3.1	Numerical schemes for solving Itô stochastic differential equations . . . . .	18
2.3.2	Accuracy and convergence in numerical simulations . . . . .	20
2.4	MONTE CARLO SIMULATIONS . . . . .	22
2.4.1	Introduction to the Monte Carlo Simulation Method . . . . .	23
2.4.2	Convergence and accuracy . . . . .	23
2.4.3	Kernel Density Estimation . . . . .	27
2.5	EQUIVALENCE OF EULERIAN AND LAGRANGIAN FORMULATIONS . . . . .	32
2.5.1	"Moderately smooth functions"-criteria and when it fails . . . . .	35
<b>3</b>	<b>CONCENTRATION FIELD IN DIFFUSIVE DOUBLE GYRE FLOW</b>	<b>37</b>
3.1	FLUID METHOD: NUMERICAL IMPLEMENTATION . . . . .	39
3.1.1	Domain discretization . . . . .	39
3.1.2	Crank-Nicolson scheme . . . . .	41
3.1.3	Constraints of numerical model . . . . .	45
3.2	FLUID METHOD: NUMERICAL RESULTS . . . . .	47
3.2.1	Convergence of Crank-Nicolson . . . . .	47
3.2.2	Conservation of mass . . . . .	47
3.2.3	Concentration solution . . . . .	50
3.3	PARTICLE METHOD: NUMERICAL IMPLEMENTATION . . . . .	50
3.3.1	Euler-Maruyama scheme . . . . .	51
3.3.2	The Wiener process . . . . .	52
3.3.3	Monte Carlo simulations . . . . .	53
3.4	PARTICLE METHOD: NUMERICAL RESULTS . . . . .	53
3.4.1	Strong and Weak Convergence . . . . .	53
3.4.2	Estimated concentration solution . . . . .	54

<b>4</b>	<b>BANDWIDTH SELECTION IN KERNEL DENSITY ESTIMATION</b>	<b>56</b>
4.1	IMPLEMENTATION . . . . .	56
4.1.1	Kernel Density Estimator . . . . .	56
4.1.2	Double Gyre System . . . . .	56
4.2	OPTIMAL BANDWIDTH IN DOUBLE GYRE SYSTEM . . . . .	57
4.2.1	Optimal bandwidth with regards to ISE . . . . .	58
4.2.2	Optimal bandwidth as a function of diffusion and time . . . . .	61
4.2.3	Optimal bandwidth as a function of Lagrangian particles . . . . .	62
<b>5</b>	<b>DISCUSSION AND CONCLUSION</b>	<b>66</b>
5.1	DISCUSSION . . . . .	66
5.1.1	Future Work . . . . .	67
5.2	CONCLUSION . . . . .	68
	<b>APPENDIX</b>	<b>I</b>
A	COMPLIMENTARY THEORY AND DERIVATIONS . . . . .	I
A.1	Derivation of the Advection-Diffusion Equation . . . . .	I
A.2	Finite Difference Methods . . . . .	IV
A.3	Crank-Nicolson Scheme . . . . .	VIII
A.4	Taylor's Theorem . . . . .	X
A.5	The Fokker-Planck equation . . . . .	X
B	COMPLIMENTARY NUMERICAL WORK . . . . .	XI
B.1	Equivalence of Eulerian and Lagrangian formulations in 2D . . . . .	XI
C	NOTES ON NUMERICAL IMPLEMENTATION . . . . .	XIV
C.1	KDE with periodic boundary conditions . . . . .	XIV
C.2	Crank-Nicolson scheme 2D . . . . .	XV
D	ADDITIONAL FIGURES AND TABLES . . . . .	XVIII

# Definitions and Notation

## Vectors and scalars

For vectors we use bold types and square brackets, e.g.,  $\mathbf{u} = [u, v, w]$ . Unit vectors are additionally marked with hats, e.g.,  $\hat{\mathbf{x}}$ . Vector components are also expressed using unit vectors, e.g.,  $\mathbf{u} = u\hat{\mathbf{x}} + v\hat{\mathbf{y}} + w\hat{\mathbf{z}}$ . Scalars are given in nonbold.

## Notation for differentiation

For partial derivatives we will be using either of the notations  $\partial_x = \frac{\partial}{\partial x}$ , and  $\partial_{xx} = \frac{\partial^2}{\partial x^2}$ , whenever convenient. For derivatives of univariate functions we also use Lagrange's notation, e.g.  $f'(x)$ . Higher derivatives are indicated using additional prime marks, and even higher order derivatives with superscripts of numerals in parentheses, e.g.  $f^{(n)}(x)$ .

## Other mathematical expressions

Values averaged over space are indicated with a bar, e.g.,  $\bar{a}$ . The norm is written  $|a|$ . The normal distribution is indicated by  $\mathcal{N}(\mu, \sigma^2)$ , with a mean  $\mu$  and variance  $\sigma^2$ . The symbol  $\sim$  is used to indicate proportionality, e.g.,  $f \sim (\Delta x)^2$ .

$x_i$  and  $y_j$  are the values of the coordinates  $x$  and  $y$  at the nodes  $i$  and  $j$ . Subscripts are for points in space, and the superscripts are for points in time, e.g.,  $f_i^n$ .

## Matrices and column vectors

Matrices in general are written in the font  $\mathbb{A}$ . Some matrices however, when implemented numerically in this thesis, are written in the font  $\mathcal{A}$ . These are

$\mathcal{C}$ and $\mathcal{C}_{j,i}$ :	Concentration matrix
$\mathcal{U}$ and $\mathcal{U}_{j,i}$ :	Velocity matrix (matrix version of velocity field $u$ )
$\mathcal{V}$ and $\mathcal{V}_{j,i}$ :	Velocity matrix (matrix version of velocity field $v$ )
$\mathcal{D}$ and $\mathcal{D}_{j,i}$ :	Diffusion coefficient matrix

$\mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{D}$  are all row-ordered column vectors of the same fields.

## Acronyms

PDE	Partial Differential Equation
ODE	Ordinary Differential Equation
SDE	Stochastic Differential Equation
KDE	Kernel Density Estimation
ISE	Integrated Squared Error

# List of Figures

1.1.1	Double gyre flow field. . . . .	3
2.1.1	Illustrating a diffusive system. . . . .	8
2.2.1	Realizations of the Wiener process. . . . .	11
2.2.2	From discrete random walk to the Wiener process. . . . .	12
2.2.3	Different resolutions of the Wiener process. . . . .	13
2.2.4	Euler's method for ODEs. . . . .	15
2.2.5	Geometric Brownian motion. . . . .	17
2.4.1	Example of Monte-Carlo simulation technique. . . . .	24
2.4.2	Example of sample mean convergence. . . . .	25
2.4.3	Illustrating bin location choices in histograms. . . . .	28
2.4.4	Illustrating bin width choices in histograms. . . . .	28
2.4.5	Example for comparing histograms and kernel density estimates. . . . .	29
2.4.6	Illustrating the concept of Gaussian kernel density estimation. . . . .	30
2.4.7	Common kernels. . . . .	31
2.4.8	KDE dependency on bandwidth and sample size. . . . .	33
2.5.1	Comparing Lagrangian and Eulerian solutions: 1D uniform diffusion. . . . .	34
2.5.2	Comparing Lagrangian and Eulerian solutions: 1D Gaussian diffusion. . . . .	34
2.5.3	Non-equivalent solutions: 1D box-shaped diffusion. . . . .	36
2.5.4	Equivalent solutions: 1D smooth box-shaped diffusion. . . . .	36
3.1.1	Illustration of implemented grid structure. . . . .	40
3.1.2	Plot of the outline of a five-diagonal Crank-Nicolson scheme matrix. . . . .	43
3.1.3	Illustration of example system grid. . . . .	43
3.1.4	Illustration of periodic boundary condition in the Crank-Nicolson scheme. . . . .	44
3.1.5	Illustration of the implemented Crank-Nicolson scheme. . . . .	44
3.2.1	Log-log plot of convergence in time of the Crank-Nicolson scheme. . . . .	48
3.2.2	Log-log plot of the convergence in space of the Crank-Nicolson scheme. . . . .	48
3.2.3	Plot of deviations in mass using the fluid method. . . . .	49
3.2.4	Contour plot of the fluid method solution, in a system of $D_0 = 0.0005$ . . . . .	50
3.3.1	Implementations of the Wiener process' adaptive resolution. . . . .	52
3.4.1	Log-log plot of strong convergence in the Euler-Maruyama scheme. . . . .	54
3.4.2	Log-log plot of weak convergence in the Euler-Maruyama scheme. . . . .	55
3.4.3	Contour plot of the particle method solution for $D_0 = 0.001$ . . . . .	55
4.2.1	Plot of ISE as a.f.o. bandwidth, for a system with constant diffusion $D_0 = 0.0005$ . . . . .	59
4.2.2	Plot of ISE as a.f.o. bandwidth, for a system with constant diffusion $D_0 = 0.001$ . . . . .	59
4.2.3	Plot of ISE as a.f.o. bandwidth, for a system with constant diffusion $D_0 = 0.005$ . . . . .	60
4.2.4	Plot of ISE as a.f.o. bandwidth, for a system with constant diffusion $D_0 = 0.01$ . . . . .	60
4.2.5	Plot of the optimal bandwidth as a.f.o. diffusion. . . . .	61
4.2.6	Plot of optimal bandwidth as a.f.o. time. . . . .	62
4.2.7	Plot of the optimal bandwidth as a.f.o. $N$ , with curve fit ( $D_0 = 0.0005$ ). . . . .	63

4.2.8	Plot of the optimal bandwidth as a.f.o. $N$ , with curve fit ( $D_0 = 0.001$ ).	63
4.2.9	Plot of the optimal bandwidth as a.f.o. $N$ , with curve fit ( $D_0 = 0.005$ ).	64
4.2.10	Plot of the optimal bandwidth as a.f.o. $N$ , with curve fit ( $D_0 = 0.01$ ).	64
A.1	Illustration of advection, diffusion, and reaction.	I
A.2	Illustration of a 3D control volume.	II
A.3	Illustration of advection and diffusion.	III
A.4	Illustration of finite differences: backward, forward and central.	VI
A.5	Illustration of the stencil in the 1D Crank-Nicolson scheme.	VII
A.6	Illustration of the stencil in the 2D Crank-Nicolson scheme.	VIII
B.1	Contour plot of the 2D box diffusion problem, $D_b$ and $D_s$ .	XII
B.2	Contour plot of solution to 2D box-shaped diffusion problem.	XIII
B.3	Contour plot of solution to 2D smooth box-shaped diffusion problem.	XIII
C.1	Plot showing periodic BCs in KDEpy implementations.	XIV
C.2	Illustration of the concept of convex closure.	XIV
C.3	Illustration of Crank-Nicolson matrices $\mathbb{L}$ and $\mathbb{R}$ .	XVII
D.1	Double gyre system perturbation $\epsilon$ .	XVIII
D.2	Contour plot of the fluid method solution for $D_0 = 0.0005$ and $D_0 = 0.001$ .	XIX
D.3	Contour plot of the fluid method solution for $D_0 = 0.005$ and $D_0 = 0.01$ .	XX
D.5	Fluid method solution plotted for $D_0 = 0.01$ , at all $T$ and all grid sizes.	XXII
D.6	Fluid method solution plotted for $D_0 = 0.005$ , at all $T$ and all grid sizes.	XXIII
D.7	Fluid method solution plotted for $D_0 = 0.001$ , at all $T$ and all grid sizes.	XXIV
D.8	Fluid method solution plotted for $D_0 = 0.0005$ , at all $T$ and all grid sizes.	XXV

# List of Tables

3.0.1	Table of the implemented temporal discretization. . . . .	38
3.0.2	Table of parameters for Gaussian initial condition. . . . .	39
3.1.1	Table of parameter dependency in the advection-diffusion problem. . . . .	46
3.3.1	Table of simulation setup for the Lagrangian particle method. . . . .	53
4.2.1	Table of optimization parameters from curve fit $\Delta_{optimal} \sim 1/N^b$ . . . . .	65
B.1	Table of parameters for implementing 2D box-shaped diffusion. . . . .	XI
D.1	Table of optimal bandwidths and ISE for $N = 100000$ . . . . .	XVIII

# Chapter 1

## INTRODUCTION

The major global challenges of environmental pollution are a result of rapid industrialization and urbanization, continuing exploitation of natural resources, and escalating production of industrial wastes and consumer goods, bringing about unfavorable alterations to the environment. By directly or indirectly affecting the natural energy cycle and radiation levels, and both the chemical and physical constitution of organisms, environmental pollution attracts our attention due to its severe long-term consequences. The unfortunate effects are evident in all parts of the environment, from loss of biodiversity and vegetation, to the excessive amount of hazardous substances in the atmosphere and oceans. Solving some of the problems related to environmental pollution, or attempting to avoid the more severe consequences, requires accurate knowledge about how pollution is transported, settled, and degraded over time. Accurate transport models can be great tools for studying the physical and chemical processes in the atmosphere and ocean, predicting future changes in the climate, and evaluating the effects of different mitigation efforts.

Numerical transport models are mainly based on either a Lagrangian or an Eulerian formulation of fluid kinematics, or a combination of the two complementary descriptions. The Eulerian approach is based on describing fluid motion in a reference frame that is fixed in space. In the Lagrangian description, the observer follows along with the observed matter, describing the motion from the point of view of the time-dependent position of the fluid "parcel." The two formulations result in two separate mathematical descriptions of fluid motion, and also numerical frameworks for modelling transport. The more direct and mathematically complex method is the Eulerian, for which advection-diffusion problems typically are solved using, e.g., finite difference methods to approximate solutions to partial differential equations on the fixed Eulerian grid. This is the approach used in most ocean circulation models, and is also commonly applied to pollution transport problems such as air pollution (Christensen, 1997), global ocean microplastic transport (Mountford and Morales Maqueda, 2019), or oil spill scenarios (Meier and Höglund, 2013). Although it is the most conventional method for modeling transport problems, the Eulerian method has certain disadvantages. For many applications, the less computationally demanding Lagrangian method can be used, by looking at the problem from another perspective, literally. The method is used in many of the same problems, including air pollution (Stohl et al., 2002), marine transport of debris and plastic (Lebreton et al., 2012; Wichmann et al., 2019; Onink et al., 2019), and oil spills (French-McCay, 2004; De Dominicis et al., 2013).

### 1.1 EULERIAN AND LAGRANGIAN TRANSPORT MODELS

The Eulerian and Lagrangian methods have different approaches to solving transport problems, but both attempt to solve them as advection-diffusion problem. A Lagrangian formulation focuses

on an individual particle's trajectory, while the Eulerian formulation describes the transport of physical quantities in terms of their concentration fields.

The physical phenomena where particles, energy, or other physical quantities are transported inside a physical system due to advection and diffusion processes is described by the Advection-Diffusion Equation,

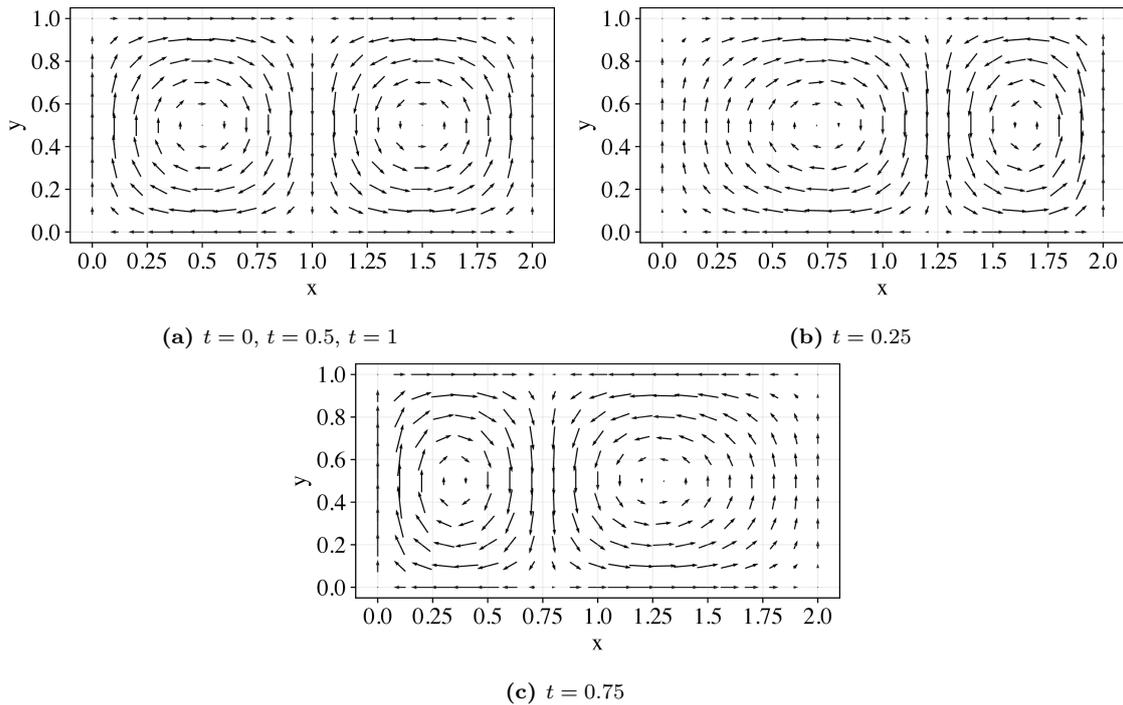
$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}C) = \nabla \cdot (D \nabla C), \quad (1.1.1)$$

where  $C$  is the concentration field of the physical quantity of interest, and  $D$  is the diffusivity. The background flow field  $\mathbf{u}$  is typically predefined and given by an Eulerian model or an analytical field. Because the advection-diffusion equation can be interpreted as a Fokker-Planck Equation, for which the concentration field is considered the probability density function of particles, an alternative method can be derived for solving it. In applied meteorology and oceanography particularly, the so-called Lagrangian particle method presents an elegant alternative to the Eulerian method. In the case of, e.g., a pollutant transported in a fluid flow, it is possible to derive a stochastic differential equation that models the motion of a single pollutant particle, say by a random walk model. When the stochastic differential equation is solved for an ensemble of particles, e.g., by using discrete stochastic schemes for the particle dynamics, then the distribution of particles evolves in a manner consistent with the advection-diffusion equation (1.1.1). The Lagrangian particle method solution is a discrete set of particles, and the construction of the density field implied by the particle positions is called "gathering". There are many ways to construct the density field (see, e.g., Lynch et al. (2014)), such as using histograms or kernel density estimation. The particle method provides a much more intuitive physical interpretation of the transport problem, dealing with discrete elements and not with the continuous fields as in the Eulerian fluid method.

There are advantages and disadvantages to both methods. E.g., oil spills are traditionally simulated as a large number of particles following the Lagrangian formulation, with the benefit of resolving processes on multiple scales. However, the solution does depend heavily on the number of virtual particles initially released. The Eulerian method resolves the entire field as a whole, without needing to consider the specific initial amount of particles, but is at risk of missing small-scale processes due to the grid's fixed resolution. Additionally, the method treats all grid cells equally, resulting in just as high computational effort for simulating areas with low concentrations as areas of high concentrations. There are ways to go about the problems related to the grid by implementing adaptive unstructured grids. Still, this will result in a relatively computationally demanding method. The Lagrangian particle method offers specific advantages because it focuses on individual particle trajectories. The method presents the opportunity to give each virtual particle individual properties (see van Sebille et al. (2018) and references therein), such as, e.g., buoyancy, age, or sinking velocities describing everything from fluid elements to organisms. Lagrangian particle trajectories can be investigated in reverse time as well (e.g., Sebille et al. (2015); Prants et al. (2017)), a technique particularly important in problems where we are concerned with where tracers have been or have come from, rather than where they are headed for, such as, e.g., the source of an oil spill. All in all, the Lagrangian method is versatile, explicitly conserves concentration, and reasonably easy to implement in transport applications whenever the velocity field is given. This thesis will investigate the Lagrangian particle method and how it compares to the Eulerian method, using tools such as histograms and kernel density estimations to compare the resolved distributions. This will be done for an analytical flow field to be presented in the following.

### 1.1.1 Analytical double gyre flow field

Both Lagrangian and Eulerian modeling of environmental transport require a known flow field. The field can be known through measurements but is often a result of an Eulerian model (van Sebille et al., 2018), or a combination of both. A third option is to use analytical flow fields as



**Figure 1.1.1:** Double-gyre flow field for parameters  $A = 0.1$ ,  $\omega = 2\pi$ , and  $\epsilon = 0.25$ .

approximations to real environmental flows or merely as a tool in, e.g., developing computational methods.

There are many ways to either measure fluid flow or model the flow using Eulerian models, which give the flow data at specific locations (measuring) or on a particular grid (modeling). To find the flow field at any other point or at any other time than what you have measured or modelled, you have to resort to techniques such as interpolation, to approximate the flow field values. However, depending on the resolution of the field, such an approximation might average over important flow patterns or eddies, and result in uncertainty. Measuring is time-consuming, resource-intensive, and unpredictable (and unknown) factors can influence the data. Even though the history of surface current measurements can be traced far back (see, e.g., Stewart (2008)), the measurement accuracy is questionable (Maximenko et al., 2012), making modeling the preferable choice. Modeling is a better option because we can achieve high-resolution fields for large areas, but it also gives the opportunity to predict future behavior and forecasting. For applications in environmental fluid mechanics, transport modeling usually requires that the velocity field is pre-calculated, commonly from an Eulerian model, before applying either Eulerian or Lagrangian models for a variety of transport applications.

Using analytical fields rather than fields from Eulerian models, gives access to the velocity value at any point in both time and space, and does not require large data sets. Of course, analytical fields never truly describe a real environmental system, but they can represent simplified models of said systems. Analytical fields can be of great use in testing numerical models before applying the "findings" to a real-world case study, using real data generated by a high-quality numerical model. In some cases, analytical fields do model real flows relatively well, e.g., simple vortices are found in the general ocean on multiple scales and have a simple analytical expression. Of course, the analytical fields do not describe the real flow exactly, nor will they ever represent a complete system. Still, they are particularly helpful for smaller closed domains and certain scales.

A simple yet interesting analytical flow field is the double-gyre flow given in Shadden et al. (2005).

This field will be given much attention in the numerical part of this thesis, and is therefore discussed and mathematically presented fully in this introductory section. The double-gyre flow is a simplified dynamical model, consisting of two counter-rotating vortices that alternately expand and contract periodically. The velocity field represents the flow of an incompressible fluid; however, the flow is not an approximated solution to the Navier-Stokes equation. The time-periodic velocity field is defined in the  $xy$ -plane, in the region  $x \in [0, 2]$  and  $y \in [0, 1]$ , and is given by

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \pi A \begin{bmatrix} -\sin(\pi f(x)) \cos(\pi y) \\ \cos(\pi f(x)) \sin(\pi y) \partial_x f \end{bmatrix} \quad (1.1.2)$$

where

$$f(x, t) = a(t)x^2 + b(t)x, \quad (1.1.3)$$

$$a(t) = \epsilon \sin(\omega t),$$

$$b(t) = 1 - 2\epsilon \sin(\omega t),$$

where  $A$  determines the magnitude of the velocity vectors. The flow is time-independent for  $\epsilon = 0$ , and will look identical to Fig. 1.1.1a at all times, with equal-sized gyres. By turning on the time-dependency, i.e., letting  $\epsilon > 0$ , the central line separating the vortices will oscillate left and right as time passes with a frequency  $\omega/2\pi$ . As the oscillations start, the gyres alternately expand and contract periodically, as shown in Fig. 1.1.1. How far this periodical motion shifts the separation line from the original separation point on the x-axis depends on the value of  $\epsilon$ . It can be shown (Shadden et al., 2005, p. 291, Eq. 77) that  $\epsilon$  is the amplitude of the oscillating motion. Thus, for very small values of  $\epsilon$ , the system can be considered a perturbation from the time-independent case of  $\epsilon = 0$ . Figure D.1 in Appendix D.1 visualizes the effect of a change in  $\epsilon$ , showing the gyres at their maximum and minimum sizes during one period.

### Lagrangian Coherent Structures: Mixing, turbulence and chaotic flows

The standard approach of predicting trajectories, by running numerical models and use resulting velocity fields to forecast trajectories, is relatively sensitive to the choices of spatial and temporal initial conditions (Peacock and Haller, 2013). Compensating by computing a larger number of solutions for the same scenario might give a better idea, but it can also typically produce large distributions of advected particles that completely hide the *key organizing structures* of the flow (Peacock and Haller, 2013). These structures referred to as the "hidden skeleton of fluid flow" (Peacock and Haller, 2013) are the underlying structures creating the tracer patterns with coherent features we find in nature. The existence of the emerging Lagrangian coherent structures (LCSs) (Haller and Yuan, 2000; Shadden et al., 2005; Peacock and Haller, 2013; Haller, 2015) in turbulent fluid flow is clear from observations, but it is harder to both define what they are and describe them physically. The double-gyre system presented above (Shadden et al., 2005) provides a simple means of investigating further the LCSs as parts of the turbulent flow. It is also a very visually descriptive flow of the workings of LCSs, as strong attracting LCSs ensure thin filaments of concentration that have been stretched along the structures. The resulting chaotic stirring in the flow leads to enhanced diffusive mixing via repeated stretching and folding of fluid elements (Pratt et al., 2015). The double gyre flow is a prime tool for investigating these structures, and although finding the LCSs is outside of the scope of this thesis, we will be able to observe the dynamical behavior as part of the advection patterns emerging, giving a different picture of the flow than the field itself.

## 1.2 OBJECTIVES

This thesis aims at describing concepts behind Lagrangian particle methods and Eulerian fluid methods for solving advection-diffusion problems, and provide practical insight into numerical schemes in both formulations, using their performance in a specific double gyre system as a case study.

### Objectives

The one-dimensional advection-diffusion equation is a well-investigated numerical problem, and our wish is to shed some light on the same setup in two dimensions. We intend to investigate a diffusive system with a double gyre flow, in the hopes of achieving a better understanding of constraints in numerical advection-diffusion problems in general, discovering the pitfalls of numerical calculations, and get a better feel of the two-dimensional aspects of modeling transport problems.

Our goal is to introduce the transport methods through an overview of theoretical aspects, supported by simple yet instructional numerical examples. We intend to get familiar with the mathematical framework, proficient references, and numerical schemes theoretically and methodically. Through this work, it should be shown that the two formulations give equivalent solutions given that the right criteria are fulfilled; criteria which should be explicitly expressed and demonstrated. The Lagrangian particle method gives a discrete set of numerical particles as solution, for which we should construct a probability density function using kernel density estimation. We wish to find the optimal kernel bandwidth, and investigate how it depends on system variations such as diffusion, time, and number of Lagrangian particles used in construction of the density distribution.

## 1.3 OUTLINE

The thesis is organized in five chapters, roughly divided in three parts:

Introduction and theory:	Chapters 1, 2
Numerical work:	Chapters 3, 4
Discussion and conclusion:	Chapter 5

Chapter 1 gives an introduction to the background and motivation behind the thesis, through a description of the two complementary formulations of fluid motion; Chapter 2 starts with an introduction to environmental transport, and describes the conventional method for solving the advection-diffusion equation in the Eulerian formulation, through the use of finite difference methods. The main focus of the chapter is investigating the theory and techniques behind the Lagrangian formulation; an introduction to stochastic differential equations (SDEs), their numerical solutions via time discrete approximations and Monte-Carlo simulations, all with a focus on how the advection-diffusion equation can be solved through the Lagrangian particle method; Chapter 3 presents the numerical implementations and results of solving the two-dimensional advection-diffusion equation using both formulations; Chapter 4 compares the two methods, and describes the optimal estimation parameter for minimizing the difference between the solutions; Chapter 5 offers a discussion on different aspects of the simulation results and a conclusion.

## Chapter 2

# FORMULATING THE ADVECTION-DIFFUSION PROBLEM

Using an Eulerian method is the conventional way to solve transport problems, but it is not the only way. In this chapter, we introduce transport phenomena through the Eulerian formulation (Section 2.1), before we direct all our attention to the Lagrangian formulation through the theory of stochastic differential equations and their numerical solutions (Sections 2.2-2.4). The two formulations are compared using simple and illustrative numerical examples to emphasize their connection (Section 2.5).

### 2.1 TRANSPORT PHENOMENA

Environmental fluid mechanics is the study of fluid mechanical processes that affect the fate and transport of physical quantities through the hydrosphere and atmosphere, describing the naturally occurring fluid flows of air and water and the quantities carried by these. The unified study of *transport phenomena* includes the transport of mass, momentum, and heat, as they have closely related mathematical frameworks, and frequently occur simultaneously in environmental problems (Bird, 2002). This section gives a short introduction to environmental transport and shows how advection-diffusion problems can be treated in the Eulerian formulation.

In the Eulerian framework, environmental fluid mechanics is the study of the fluid mechanical processes that change concentrations; that be of mass, momentum, or heat. The concentration is, put in simple terms of mass, a measure of the amount of substance within a mixture. Mathematically, we express the concentration,  $C$ , of a species  $i$  in a mixture as

$$C = \frac{m_i}{V}, \quad (2.1.1)$$

where  $m_i$  is the total mass of the species  $i$ , and  $V$  is the volume of the mixture (can be length or area, in one and two dimensions respectively). The processes that change the concentration can be classified into one of two main categories: *transport* and *transformation*, altering the concentration through transport by physical means and through reactions within the mixture.

## Scales of environmental fluid motion

Environmental fluid mechanics ranges from the study of the atmosphere, the oceans, lakes, and rivers - to tiny creeks, and airflows around buildings. The motions present occur in a correspondingly wide range of scales; from small surface ripples to large circulation systems the size of continents. The scales of relevance range from millimeters to thousands of kilometers, and from seconds to years (Gualtieri et al., 2012). Regional-scale dynamics can be described up to the limit of 100 km (Socolofsky and Jirka, 2002). It can be shown that above this scale, the Coriolis effect due to the Earth's rotation must be accounted for, and thus we are stepping into the realm of geophysical fluid dynamics. It differs from environmental fluid mechanics by taking into account stratification and rotation, rather than stratification and turbulence (Gualtieri et al., 2012). The wide range of scales is the cause of, in large part, the difficulties associated with understanding fluid motion in the environment. The dominating processes describing the motion in a system vary with the system's scale. See Gualtieri et al. (2012) for typical velocity, time, and length scales for the most common fluid processes and systems.

### 2.1.1 The Advection-Diffusion Equation

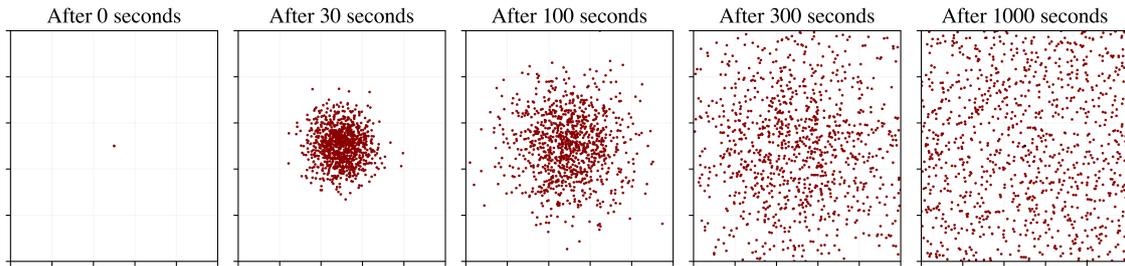
Transport refers to the process of moving quantities by physical means; mass, momentum, or heat is transported from one physical region of the system to another. The transport is a result of an influence on the quantity by the composition (concentration gradient), temperature, or velocity gradients. This movement of properties can be described by the Transport Equation – or from a mathematical point of view – the Convection-Diffusion Equation. The transport equation describes how a scalar physical quantity is transported in space, and can be derived straightforwardly from the general continuity equation,

$$\partial_t \rho + \nabla \cdot \mathbf{j} = S. \quad (2.1.2)$$

The continuity equation (2.1.2) states that the rate of change for a scalar quantity  $\rho$  in an infinitesimal control volume is given by the divergence of the total flux  $\mathbf{j}$  and the net volumetric source  $S$ , describing any generation or consumption inside the control volume.

In nature, the transport processes are a combination of advection and diffusion (Socolofsky and Jirka, 2002). While most of us have an intuitive understanding of transport through advection, the contribution to transport through the action of random motions is harder to envision. As advection moves the center of mass along the general flow direction, the particles will at the same time be spread to a larger (less concentrated) region due to diffusion. If a diffusive system is left untouched, it will end up in an equilibrium state of uniform concentration. Figure 2.1.1 illustrates how particles in a diffusive system (no advection) end up uniformly distributed in the domain, after having been transported to areas of less concentration. These are the two inherent properties of diffusion: the transport is from regions of high concentration to low concentration, and it is "random in nature" (Socolofsky and Jirka, 2002). Diffusion is the macroscopic description of the process that results from the random motion originating from microscopic (perfectly elastic) collisions in material. Both the material that diffuses and the medium in which the diffusion occurs can be in any of the three physical states; solid, gas, or liquid. The diffusion rate depends on the materials; for instance, gas will diffuse relatively fast in another gas relative to in a solid. Diffusion can also have origin in irregular turbulent velocity fluctuations, and is then referred to as turbulent diffusion. While molecular mixing is important in the smallest scales, turbulent diffusion is the dominant mechanism in a large range of time and length scales.

Returning to a more general and mathematical approach, we find the flux contributions that appear in the transport equation. Say that the scalar field described by the continuity equation (2.1.2) is the concentration field of some physical quantity. Thus  $\rho$  might represent, e.g., species concentration for mass transfer or temperature for heat transfer. The flux measures the amount of the quantity or substance of interest that will flow through a unit area of the control volume's



**Figure 2.1.1:** An example of the time development of 1000 particles in a diffusive system (constant diffusion).

surface during a unit time interval. For simplicity, let  $\rho$  stand for the concentration of a chemical species present in a fluid, and rename it  $C$ . Firstly, the diffusive flux is given by Fick's first law (Csanady, 1973, p. 5) as

$$\mathbf{j}_{\text{diffusion}} = -D\nabla C, \quad (2.1.3)$$

where the factor of proportionality,  $D$ , is called the diffusion coefficient. For molecular diffusion, the diffusivity  $D$  is a property of both the material transported and the medium it is transported in. Second, the mean fluid flow has the associated advective flux given by

$$\mathbf{j}_{\text{advection}} = \mathbf{u}C, \quad (2.1.4)$$

where  $\mathbf{u}$  is the flow velocity of the medium. In Appendix A.1 we have derived the transport equation in one dimension, mainly following Hundsdorfer (2003), before describing the possible incorporation of transformation into the system as a final step. Here, we disregard the transformation processes, and merely present the one-dimensional advection equation found in Appendix A.1,

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} = D \frac{\partial^2 C}{\partial x^2}, \quad (2.1.5)$$

specifically for a system with constant flow velocity  $u$ , constant diffusivity  $D$ .

### 2.1.2 Eulerian formulation of solving advection-diffusion problems

Chapter 1 introduces the direct approach of the Eulerian method for solving advection-diffusion problems, both for calculating the flow itself but also for the transport of physical quantities. Typically, the so-called *fluid method* for solving transport phenomena involves solving partial differential equations such as the advection-diffusion equation (2.1.5) on a fixed grid, e.g., using finite difference methods<sup>1</sup>. One such numerical scheme is Crank-Nicolson<sup>2</sup>; an implicit second order method in time, known for its unconditional numerical stability. Considering a typical one-dimensional second order partial differential equation, written in a general form,

$$\frac{\partial C}{\partial t} = F \left( x, t, C, \frac{\partial C}{\partial x}, \frac{\partial^2 C}{\partial x^2} \right), \quad (2.1.6)$$

with discretization  $C(i\Delta x, n\Delta t) = C_i^n$ , for which the function  $F = F_i^n$  is evaluated for the values  $i$ ,  $n$ , and  $C_i^n$ , the Crank-Nicolson scheme is given by

<sup>1</sup>See Appendix A.2 **Finite Difference Methods** for an introduction.

<sup>2</sup>See Appendix A.3: **Crank-Nicolson Scheme** for an introduction.

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = \frac{1}{2} \left[ F_i^{n+1} \left( x, t, C, \frac{\partial C}{\partial x}, \frac{\partial^2 C}{\partial x^2} \right) + F_i^n \left( x, t, C, \frac{\partial C}{\partial x}, \frac{\partial^2 C}{\partial x^2} \right) \right]. \quad (2.1.7)$$

Applied to the advection-diffusion equation (2.1.5), the scheme results in a system of algebraic equations, which in matrix form can be written as the problem

$$\mathbb{L}\mathbf{C}^{n+1} = \mathbb{R}\mathbf{C}^n. \quad (2.1.8)$$

The system is solved iteratively, with multi-diagonal sparse matrices  $\mathbb{L}$  and  $\mathbb{R}$ , for the concentration field  $\mathbf{C}^{n+1}$ . The concentration solution can be found through direct matrix inversion

$$\mathbf{C}^{n+1} = \mathbb{L}^{-1}\mathbb{R}\mathbf{C}^n, \quad (2.1.9)$$

and works well for obtaining a solution for a relatively small domain. However, matrix inversion is computationally expensive, and it must be noted that the size of the matrices  $\mathbb{L}$  and  $\mathbb{R}$  grow with the square of the size of the vector  $\mathbf{C}$ . Computational efforts can be reduced by choosing an iterative method (see, e.g., Barrett et al. (1994)) for solving the system. Consider the linear system  $\mathbb{L}\mathbf{C} = \mathbf{b}$  at a time  $n + 1$ , for  $\mathbf{b} = \mathbb{R}\mathbf{C}^n$ ; The general objective of iterative methods is constructing a sequence of improving approximate solutions  $\{\mathbf{C}^{(k)}\}_{k=1}^{\infty}$ , that converge to the fixed vector  $\mathbf{C}^*$  which is the solution of the linear system. An example of one such method is the Biconjugate Gradient Stabilized Method (see, e.g., Barrett et al. (1994, page 27)), specifically developed for solving non-symmetric linear systems such as the Crank-Nicolson scheme applied to the advection-diffusion equation.

## 2.2 STOCHASTIC DIFFERENTIAL EQUATIONS

Stochastic differential equations (SDEs) are used to model various phenomena in nature, such as population dynamics and turbulence, as well as in industry and even finance. We are particularly interested in the Fokker-Planck equation in this section, a family of partial differential equations (PDEs) that can be used in numerous fields, as it describes the time evolution of a probability density function. The equation has a close relation to the phenomenon of Brownian motion of particles, described by the Wiener process – an important stochastic process with the Markov property. This section as a whole gives the framework and theoretical background for using the Fokker-Planck equation corresponding to the Advection-Diffusion Equation, in other words, introducing the Lagrangian formulation for solving advection-diffusion problems.

### 2.2.1 Introduction to stochastic processes

A stochastic process means that one has observations of a system at certain times, and that the outcome of these observations, the values observed, is at each time a random variable (Blomberg, 2007). The evolution of numerous quantities can be described as stochastic processes, such as, e.g., daily closing prices of an index or the number of phone calls occurring in a day. For every observation made, there is a certain probability to get a certain outcome. This probability will frequently depend on previous observations, but it can also be completely independent of previous outcomes. This short introductory section will review necessary definitions for understanding stochastic processes; random variables that change with time. Basic references for this are e.g. van Kampen (2007) and Brémaud (2020).

#### Definition of stochastic processes and sample paths

A sequence of random variables  $X_{t_0}, X_{t_1}, X_{t_2}, \dots$  can be thought to describe the evolution of a probabilistic system over discrete instants of time  $t_0 < t_1 < t_2 < \dots$ . The collection of random

variables is indexed by some index set, often time  $t$  (and we will refer to it as a time set from now on), and  $X_t \in \mathbb{R}^d$  or any subset of it. We then say that it is a *d-dimensional stochastic process*, where the totality of its joint distribution functions,

$$F_{X_{t_{i_1}}, \dots, X_{t_{i_j}}}(x_{i_1}, \dots, x_{i_j}) = P(X_{t_{i_1}} \leq x_{i_1}, \dots, X_{t_{i_j}} \leq x_{i_j}), \quad (2.2.1)$$

is called the *probability law* (see, e.g., Kloeden (1992) and Platen (2010)).

Stochastic processes are defined for a time set  $\mathcal{T}$ , which can be both discrete and continuous. Discrete time stochastic processes are defined on a set of discrete time points  $t_0, t_1, t_2, \dots$  where  $t_0 < t_1 < t_2 < \dots$ . While continuous time stochastic processes can have both bounded time sets such as the interval  $[0, 1]$ , or be defined on an unbounded set such as  $\mathcal{T} \in [0, \infty)$ . Say that we freeze the outcome  $\omega = \{X_{t_0}, X_{t_1}, \dots\}$  of a discrete time stochastic process, then the time trajectory  $X_t(\omega)$  forms what is called a *sample path* of the process. In other words, the sample path describes the time evolution of a physical variable  $X$  for the specific outcome  $\omega$  (Levy, 2020).

### Time variability of a stochastic process

For both continuous and discrete time sets  $\mathcal{T}$ , it is useful to differentiate stochastic processes depending on their distinct temporal relationships. For a time  $t \geq 0$ , the mean and variance, first and second moment respectively, is given by

$$\mu(t) = E[X_t], \quad \sigma^2(t) = Var(X_t) = E[(X_t - \mu(t))^2]. \quad (2.2.2)$$

The expectation value is the central tendency of the observations. It is the sum of the products of observations and their probabilities of occurrence, i.e., a weighted average. The variance describes the spread of the observations from the average value. And lastly, we mention the *covariance*, which is a measure of the joint variability of two random variables (Levy, 2020),

$$Cov(X_s, X_t) = E[(X_s - \mu(s))(X_t - \mu(t))]. \quad (2.2.3)$$

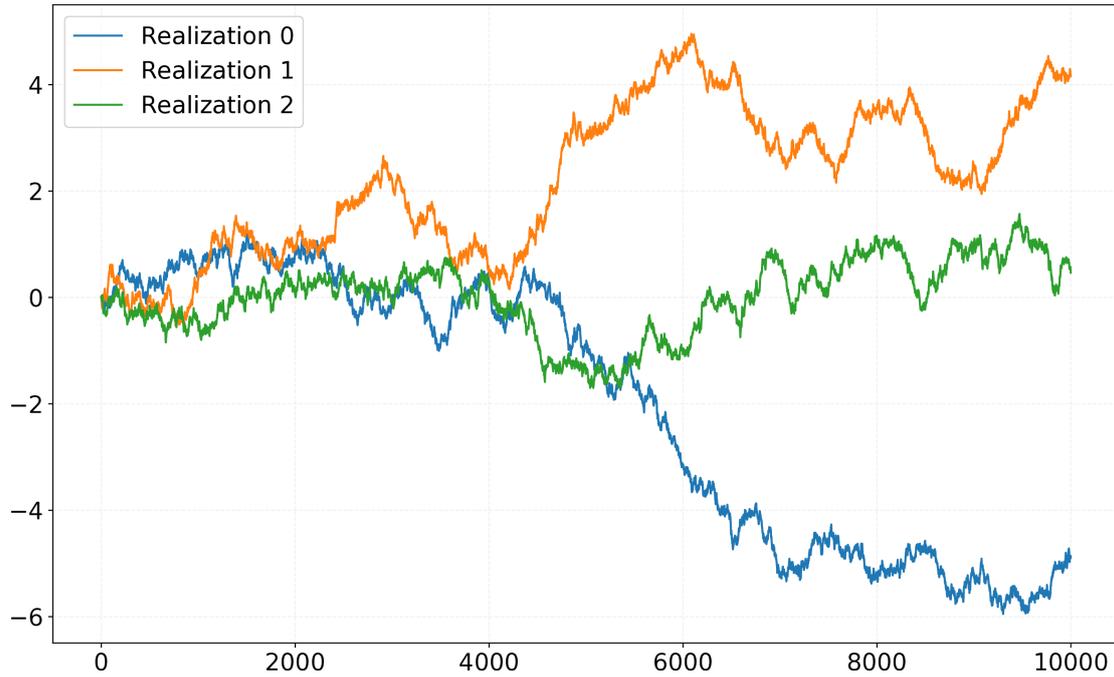
for  $s, t \in \mathcal{T}$ . The relationships presented; the expectation, variance, and covariance provide important and distinguishable information about the time variability of stochastic processes.

### The Markov property

A Markov process  $\{X_t\}$  (e.g., Pinsky (2011, p. 79), Blomberg (2007, p. 179)) is a stochastic process with the property that, given the value of  $X_t$ , the values of  $X_s$  for  $s > t$  are not influenced by the values of  $X_u$  for  $u < t$ . Or in words; the next value of the process only depends on the known present value, and not on any of the past outcomes of the process. When a process has this property, we say that it is *Markov* or has the *Markov property*. For a process with time index set  $\mathcal{T} = (0, 1, 2, \dots)$ , and whose state space is a finite or countable set, the Markov property is given by

$$P(X_{n+1} = j | X_0 = i_0, \dots, X_{n-1} = i_{n-1}, X_n = i) = P(X_{n+1} = j | X_n = i), \quad (2.2.4)$$

for all time points  $n$  and all states  $i_0, \dots, i_{n-1}, i, j$ . Eq. (2.2.4) describes how the conditional probability distribution of the future state  $X_{n+1}$  (conditional on both present and all past values of the process) depends only on the present state. This memorylessness is found in stochastic processes such as the Poisson process, or of bigger interest to us – the Wiener process – that we will have a closer look at in the next section.



**Figure 2.2.1:** Realizations of  $W_t$ , the Wiener process. The Wiener process increments,  $\Delta W = W_{t_{n+1}} - W_{t_n}$ , are Gaussian with mean  $\mu = 0$  and variance  $\sigma^2 = t_{n+1} - t_n$ , here given for 10000 time steps of size  $\Delta t = 10^{-3}$ .

## 2.2.2 Brownian Motion and the Wiener Process

Through the observation of the irregular motion of pollen particles suspended in water, a special kind of physical process was discovered. Today it is known as *Brownian motion*, after botanist Robert Brown who first described the phenomenon (see, e.g., Zwanzig (2001); Blomberg (2007); Petroni (2020)). The erratic motion he observed was later explained by Albert Einstein by the random collisions with the molecules of the liquid. The phenomenon is highly observable in the natural world, not only found in pollen particles suspended in water, but any particle suspended in a fluid. The mathematical framework describing the motion includes the concept of a stochastic process that represents the position of a particle, a process named the *Wiener process*. The physical process and the mathematical process are both frequently referred to as Brownian motion, although somewhat misleading.

The Wiener process represents a mathematical idealization of the Brownian motion of suspended particles in a fluid (Levy, 2020). It is an important Markov process, used heavily in the construction of stochastic models. Figure 2.2.1 shows three realizations of the process, visualizing the erratic, random movement. The Wiener process  $W_t$  is characterized by the following basic properties:

$$\underline{\text{The Wiener Process}} \tag{2.2.5}$$

### 2.2.5.i – Independence:

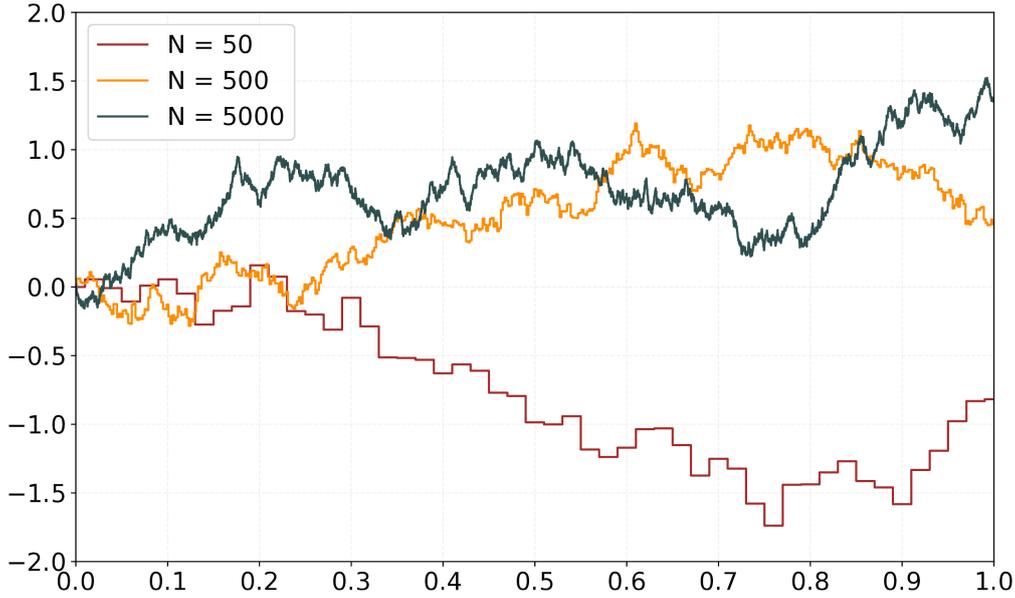
$W_t - W_s$  is independent of  $\{W_\tau\}_{\tau \leq s} \forall 0 \leq s \leq t$

### 2.2.5.ii – Stationarity:

The statistical distribution of  $W_{t+s} - W_s$  is independent of  $s$  (and therefore equivalent in distribution to  $W_t$ .)

### 2.2.5.iii – Gaussianity:

$W_t$  is a Gaussian Process with mean and covariance  
 $E[W_t] = 0$  and  $E[W_t \cdot W_s] = \min(t, s)$ .



**Figure 2.2.2:** For increasing values of  $N$ , three realizations of the function  $W_t^N$  given by Eq. (2.2.8).

*2.2.5.iv – Continuity:*

With probability 1,  $W_t$  viewed as a function of  $t$  is continuous (i.e. the set of discontinuities has measure zero).

Property *2.2.5.iv* is crucial and its most striking feature: the sample paths of a Wiener process are continuous almost surely, but are also nowhere differentiable almost surely. The constantly changing direction causes the sample paths of the process to be fractal curves (Levy, 2020).

To get a better feel of the stochastic process, we will look at a short construction by scaling a random walk in one dimension, as suggested by, e.g., Vanden-Eijnden (2006). We start by letting  $\{\xi_k\}_{k \in \mathbb{N}}$  be a sequence of independent identically distributed (i.i.d.) discrete random variables with equal probability  $1/2$  for the values  $\pm 1$ . In essence, there is an equal probability of stepping to the left and step to the right. Thus, we have expectation and variance of the i.i.d. variables given by  $E[\xi_k] = 0$  and  $E[\xi_k^2] = 1$ . Next, we can define the position of a random walk  $X_n$  as a function of time  $n$ ,

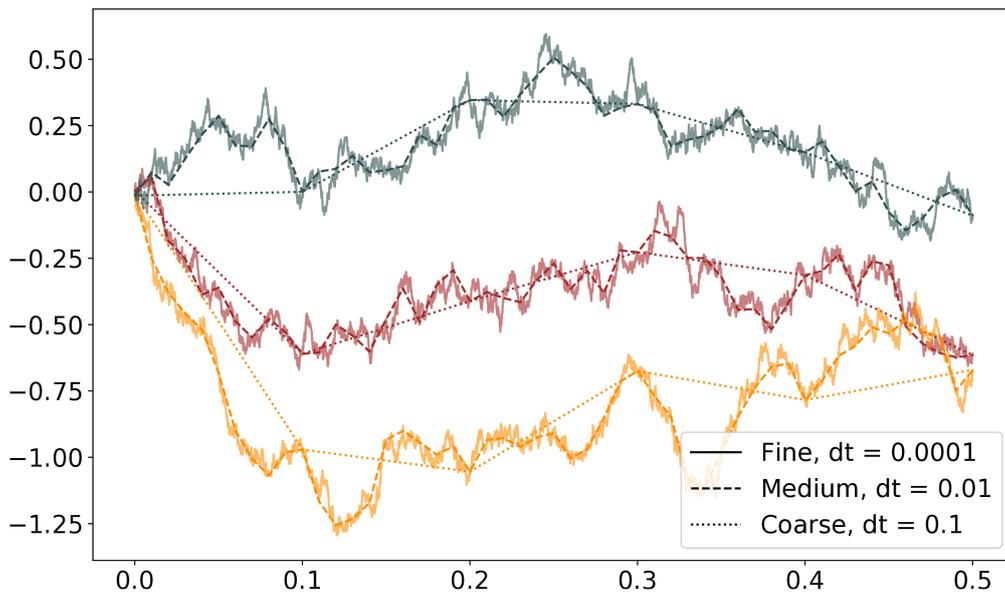
$$X_n = \sum_{k=1}^n \xi_k \text{ for } P(\xi_k = \pm 1) = \frac{1}{2}, \forall n \in \mathbb{N}. \quad (2.2.6)$$

By scaling and by applying the central limit theorem (e.g., Kadanoff (2000, Chapter 3.6)), we find that the distribution of  $X_N/\sqrt{N}$  converges to a standard normal distribution:

$$\frac{X_N - N \cdot \mu_{\xi_k}}{\sigma_{\xi_k} \cdot \sqrt{N}} = \frac{X_N}{\sqrt{N}} \xrightarrow{N \rightarrow \infty} \mathcal{N}(0, 1). \quad (2.2.7)$$

We can then define the piecewise constant random function  $W_t^N$  on  $t \in [0, 1]$  by letting

$$W_t^N = X_{\lfloor Nt \rfloor} / \sqrt{N}, \quad (2.2.8)$$



**Figure 2.2.3:** Three realizations of the Wiener process, each with three levels of resolution.

where  $\lfloor Nt \rfloor$  indicates the floor function. Hence, we have gone from a discrete random walk to a continuous step function. Finally, it can be shown that as  $N \rightarrow \infty$ ,  $W_t^N$  converges to the stochastic process  $W_t$  – the Wiener process:

$$W_t^N \xrightarrow[N \rightarrow \infty]{} W_t. \quad (2.2.9)$$

Figure 2.2.2 shows some realizations for three different discretizations of  $W_t^N$ , and see how with increasing  $N$ , the process starts to look like the Wiener Process, as suggested by Eq. (2.2.9).

We will look at some stochastic processes where the Wiener process contributes with random fluctuations. The Wiener process and its direct transformations, including geometric Brownian motion and the Ornstein-Uhlenbeck process, are the most frequently used diffusion models in applications (Platen, 2010). For more flexibility, one can control the resolution of the processes. In other words, use time steps of a variety of sizes for the same exact process. From a realization of a high-resolution process, increments can be added to create a process with a coarser resolution. In other words, if the finest resolution process has increments

$$\{0, \Delta W_1, \Delta W_2, \Delta W_3, \Delta W_4, \Delta W_5, \Delta W_6, \dots\} \text{ at times } [t_0, t_1, t_2, \dots, t_5, t_6, \dots]$$

then the coarser resolution process will be described by the combined increments

$$\{0, (\Delta W_1 + \Delta W_2), (\Delta W_3 + \Delta W_4), (\Delta W_5 + \Delta W_6), \dots\} \text{ at times } [t_0, t_2, t_4, t_6, \dots].$$

Figure 2.2.3 shows the concept of changing the resolution of a particular realization of a process, starting from the one with the finest resolution.

### 2.2.3 Stochastic differential equations

Differential equations with one or more terms containing stochastic processes are called stochastic differential equations (see, e.g., Kloeden (1992); Øksendal (2003)). Their solutions are also stochastic

processes. The most common form of stochastic differential equations (SDEs) is ordinary differential equations (ODEs) with additional perturbations in the form of a white noise term. As with stochastic processes in general, SDEs model various systems of variability and fluctuations, such as thermal fluctuations or stock prices.

In the following, we will distinguish stochastic models from deterministic through an introduction containing ODEs. We will then move on to Itô calculus and the Itô process that will introduce us to stochastic differential equations. We should mention that there are two formulations of stochastic calculus, but we shall only concern ourselves with Itô Calculus in this thesis.

### Stochastic and deterministic models

Stochastic processes can be contrasted with deterministic processes. Contrary to stochastic models, a deterministic model will describe precisely how the system will evolve; any number of realizations will result in the same outcome. In a stochastic model however, where the evolution of the system is at least partially random, each realization will give non-identical results. Both types of models build on solving differential equations as time evolves. We start by looking at a deterministic model, and will take a closer look at stochastic differential equations in the following sections.

We start out by looking at an ordinary differential equation,

$$\frac{dZ_t}{dt} = a(Z_t, t) \quad (2.2.10)$$

or in the differential form,

$$dZ_t = a(Z_t, t)dt, \quad (2.2.11)$$

that can be seen as a degenerate form of a stochastic differential equation, as  $a$  is yet to be defined (Kloeden, 1992, Chp. 3.1). More accurately, we write Eq. (2.2.11) as an integral equation,

$$Z_t = Z_0 + \int_0^t a(Z_s, s)ds, \quad (2.2.12)$$

where  $Z_t = Z(t; Z_0, t_0)$  is a solution satisfying the initial condition  $Z_{t_0} = Z_0$ . Assumptions are made on  $a$  (e.g., Lipschitz continuity in Kloeden (1992, p. 76)) to ensure the existence of a unique solution  $Z(t; Z_0, t_0)$  for each initial condition, and the solutions are related by the evolutionary property

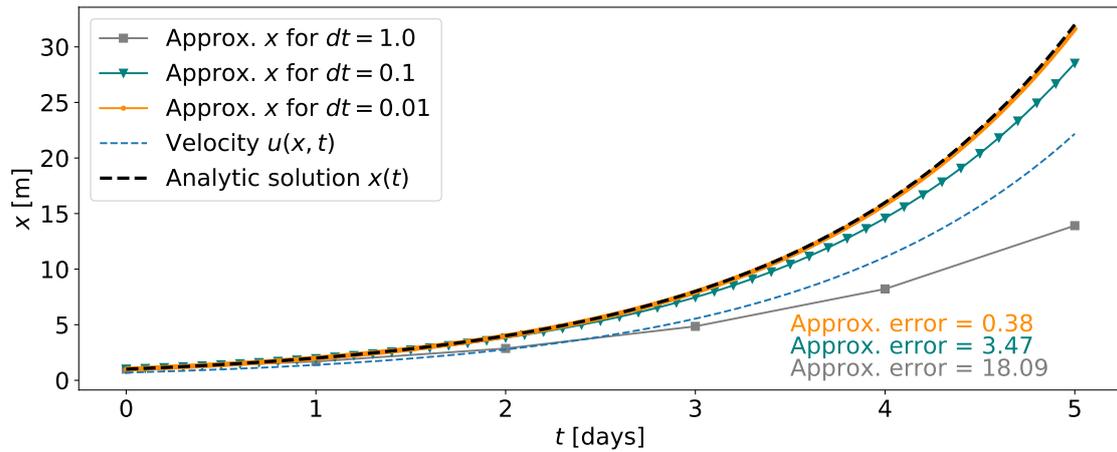
$$Z(t; Z_0, t_0) = Z(t; Z(s; Z_0, t_0), s) \quad (2.2.13)$$

for all  $t_0 \leq s \leq t$ . Of course, this deterministic process  $Z_t$  does not show any form of randomness; the future is determined completely by the present.

To put this in perspective, let us look at the simple example of a deterministic model of a particle in one dimension, being advected passively through a velocity field. The trajectory of the particle is defined by the ordinary differential equation (ODE)

$$\dot{x} = u(x, t), \quad (2.2.14)$$

where  $u$  is the velocity at time  $t$  and position  $x$ . To ensure a unique solution the initial condition must be provided  $x(t_0) = x_0$ . The initial value problem can be solved numerically by using numerical integration of the differential equation. We rewrite the equation as an integral equation,



**Figure 2.2.4:** Euler method for different time steps  $\Delta t$  approximating the solution of  $\dot{x} = u(x, t)$  for  $u(x, t) = x \log(2)$ , and initial condition  $x(0) = 1$ . The example field is from a numfys.net notebook called *Euler's method*. The analytic solution is  $x(t) = 2^t$ .

$$x_t = x_0 + \int_0^t u(x, s) ds, \quad (2.2.15)$$

it can be solved using the Forward Euler method (see, e.g., Sauer (2014, p. 284)), with a discrete fixed time step  $\Delta t$ . That is, for  $u = u(x, t)$ , the approximation of the solution of the ODE (2.2.14) can be written

$$x(t + \Delta t) = x(t) + u\Delta t. \quad (2.2.16)$$

Since there are no random terms in the deterministic model, the trajectory is determined and will not change from realization to realization. Only the value of the time step  $\Delta t$  can change the time trajectory of  $x$ . An example of a time-discrete approximation using the Euler method is shown in Figure 2.2.4, along with the analytic solution.

However, for most realistic models, we must add some kind of noise term either additionally or to the velocity field itself. This can of course be done by redefining the coefficient  $a$  in Eq. (2.2.10), to be stochastic, in other words going from a deterministic to a stochastic model. We shall now explain differential equations in which one or more of the terms is a stochastic process. We will see how stochastic differential equations have solutions that themselves are stochastic processes, and thus we return to the concept of sample paths as discussed in subsection 2.2.1.

### Itô stochastic differential equations

The inclusion of random effects in differential equations leads to two distinct classes of equations, one for which the solution has differentiable sample paths and the class that gives non-differentiable sample paths. The latter is the one we will focus on in this chapter, and will be the basis for the numerical work in Chapter 3 and Chapter 4. The first and simplest of the two classes of equations includes random effects by introducing random coefficients, random initial value, or a combination of the two. These are solved in the same manner as the ODE presented in the previous chapter, "sample path by sample path" (Kloeden, 1992). This results in sample paths of the solution process that are differentiable functions, which the second class lack. The forcing in the second class of SDEs is through an irregular stochastic process such as Brownian motion, and can be studied through the eyes of either *Itô calculus* or *Stratonovich stochastics*, two dominating versions of stochastic calculus (e.g., Øksendal (2003); Kloeden (1992)). As the title suggests, we shall look at the second class of SDEs in the framework of Itô calculus.

### The Itô Process

An Itô process is a type of stochastic process  $X = \{X_t, t \geq 0\}$  described by Japanese mathematician Kiyoshi Itô, and here presented using theory mainly from Kloeden (1992). The Itô process has the form

$$X_t = X_0 + \int_0^t a(X_s)ds + \int_0^t b(X_s)dW_s \quad (2.2.17)$$

for  $t \geq 0$ . The process consists of an initial value  $X_0 = x_0$ , which may be random. The second term of the process is a slowly varying component called the drift, which is a *Lebesgue integral* of a process over time. The second integral is an *Itô stochastic integral* with respect to the Wiener process  $W = \{W_t, t \geq 0\}$  (see Section 2.2.2), referred to as the diffusion. This is a rapidly varying continuous random component that we shall investigate further. We rewrite the integral equation (2.2.17) into a differential equation

$$dX_t = a(X_t)dt + b(X_t)dW_t. \quad (2.2.18)$$

When the Itô process is written in terms of differentials, it is called an Itô stochastic differential equation (SDE). In general, a  $d$ -dimensional SDE can be written (Platen, 2010)

$$d\mathbf{X}_t = \mathbf{a}(\mathbf{X}_t)dt + \mathbf{b}(\mathbf{X}_t)d\mathbf{W}_t, \quad (2.2.19)$$

with initial value  $\mathbf{X}_0 \in \mathfrak{R}^d$ , and the  $d$ -dimensional Wiener process  $\mathbf{W}$ . An elementary example of a solution of the integral equation (2.2.17), or equivalently the SDE in Eq. (2.2.18), is one for

$$X_t = \mu t + \sigma W_t, \quad (2.2.20)$$

for constants  $\mu$  and  $\sigma$ . With constant coefficients  $a(X_t) = \mu$  and  $b(X_t) = \sigma$ , we can directly integrate the SDE to find the process of *Brownian motion* with drift  $\mu$  and diffusion coefficient  $\sigma$ . Another example is the standard *geometric Brownian motion*, or *exponential Brownian motion*, for which  $a(X_t) = \mu X_t$  and  $b(X_t) = \sigma X_t$ . The stochastic process  $X_t$  is said to follow a geometric Brownian motion if it satisfies the SDE

$$dX_t = \mu X_t dt + \sigma X_t dW_t. \quad (2.2.21)$$

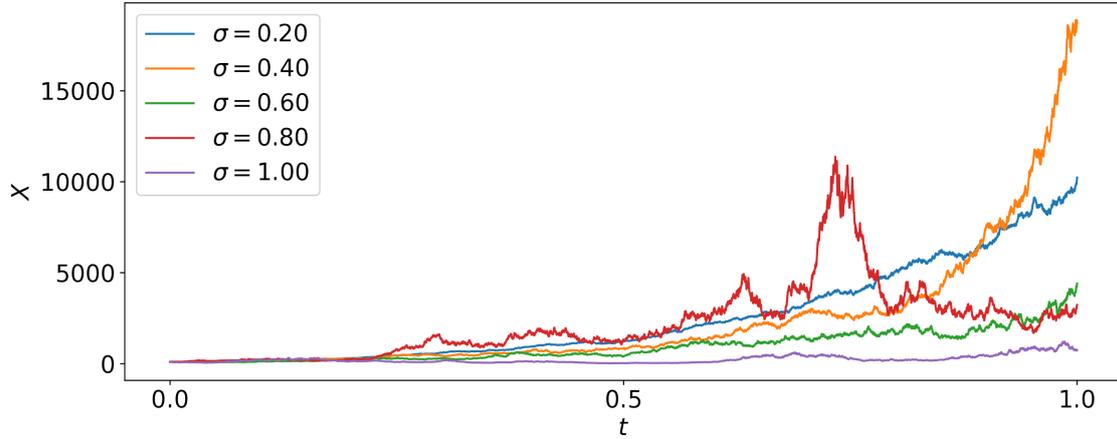
It has the solution (in Itô calculus)

$$X_t = X_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right), \quad (2.2.22)$$

Shown in Fig. 2.2.5 for a range of values  $\sigma$ .

### 2.2.4 The Fokker-Planck Equation

The Fokker-Planck equation, or Kolmogorov forward equation, is a second-order partial differential equation that describes the time evolution of a probability density function (see, e.g., Haken (2004); Platen (2010); Risken (1996)). It is an equation of many names and many applications, arising in a number of different fields (Frank, 2020). E.g., solid-state physics, chemical physics, circuit theory, and quantum optics (Tatari et al., 2007). The Fokker-Planck equation is used to describe everything from fluctuations in the intensity of laser light (Risken et al., 1966), to the stochastic behavior of exchange rates (Ivanova et al., 2003). The equation was first introduced to describe the Brownian motion of particles. For a derivation of the Fokker-Planck equation, see for instance Risken (1996, Chp. 4); Haken (2004, Chp. 4).



**Figure 2.2.5:** Realizations of Geometric Brownian Motion for different values of  $\sigma$ ,  $X_0 = 100$ .

In this thesis, we are interested in the class of stochastic differential equations (SDEs) of the form

$$dX_t = X_t + a(X_t)dt + b(X_t)dW_t, \quad (2.2.23)$$

as presented in the sections above. Assume that the process  $X_t$  has a drift term  $a(\cdot)$  and a diffusion term  $b(\cdot)$  that are "moderately smooth functions" (Kloeden, 1992, p. 37). Following Theorem 1 from Gichman (1972, p. 102)<sup>3</sup>, the explicit criterion on the smoothness of the coefficients is – in simplified terms – that:

$$[...] \text{ the partial derivatives } \frac{\partial a}{\partial x}, \quad \frac{\partial b}{\partial x}, \quad \text{and} \quad \frac{\partial^2 b}{\partial x^2} \quad \text{exist.} \quad (2.2.24)$$

Then, then the corresponding *transition probability density*  $p = p(x, t|x_0, t_0)$  satisfies the Fokker-Planck equation

$$\frac{\partial p}{\partial t} = \frac{1}{2} \frac{\partial^2}{\partial x^2} \{b(x, t)^2 p\} - \frac{\partial}{\partial x} \{a(x, t)p\}. \quad (2.2.25)$$

Here  $p$  describes the transition probability density from an initial position  $x_0$  and time  $t_0$ , to a position  $x$  at a later time  $t$ .

We could, e.g., let the Itô process  $X_t$  describe a particle position with a deterministic term that represents the advection and a random term that represents diffusion. Then the Fokker-Planck equation describes in this case, given that the smoothness criteria on the coefficients (2.2.24) are fulfilled, the time development of the probability distribution for particle position.

### The Fokker-Planck equation equivalent to the advection-diffusion equation

We now look at an example of a stochastic process whose Fokker-Planck equation is the advection-diffusion equation. This example will tie a connection between the Eulerian and Lagrangian formulation, which is a large part of the scope of this thesis. We start by rewriting the Fokker-Planck equation (2.2.25) into

$$\frac{\partial p}{\partial t} = \frac{\partial}{\partial x} \left( \frac{b^2}{2} \frac{\partial p}{\partial x} \right) - \frac{\partial}{\partial x} \left[ \left( a - \frac{1}{2} \frac{\partial}{\partial x} b^2 \right) p \right], \quad (2.2.26)$$

<sup>3</sup>The full theorem is given in Appendix A.5.

and compare this to the advection-diffusion equation (see Appendix A.1, Eq. A.5). We state it here again for clarity,

$$\frac{\partial p}{\partial t} = \frac{\partial}{\partial x} \left( D \frac{\partial p}{\partial x} \right) - \frac{\partial}{\partial x} (up), \quad (2.2.27)$$

for diffusion coefficient  $D$ , and fluid flow velocity  $u$ . We have replaced the concentration  $C$  with the normalized concentration to describe the probability distribution  $p$ .

By matching the terms in the two equations, we find the coefficients

$$\begin{aligned} D = b^2/2 & \quad \rightarrow \quad b = \sqrt{2D}, \\ u = a - \partial_x b^2/2 & \quad \rightarrow \quad a = u + \partial_x D. \end{aligned} \quad (2.2.28)$$

Thus, we have found the SDE

$$dX_t = (u + \partial_x D) dt + \sqrt{2D} dW_t \quad (2.2.29)$$

whose Fokker-Planck equation is the advection-diffusion equation. In other words, solving the SDE (2.2.29) for a large number of realizations is equivalent to solving the advection-diffusion equation (2.2.27), assuming coefficients satisfy the criteria (2.2.24). In the following section we shall use time-discrete approximations for solving SDEs, and will further discuss the equivalence of solving the PDE (2.2.27) and SDE (2.2.29) in Section 2.5: EQUIVALENCE OF EULERIAN AND LAGRANGIAN FORMULATIONS.

## 2.3 TIME DISCRETE APPROXIMATIONS

We have been introduced to stochastic differential equations (SDEs), such as the Fokker-Planck equation, in the previous section. We now want to look at the methods that exist for numerically solving these equations. Essentially, we will only consider the Itô stochastic process  $X = \{X_t, t_0 \leq t \leq T\}$  that satisfy the SDE

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t \quad (2.3.1)$$

on  $t_0 \leq t \leq T$  with the initial value  $X_{t_0} = X_0$ .

In the following theory on time discrete approximations of Itô processes, we use Kloeden (1992) as the main reference. We limit our discussion in this thesis to only time discrete approximations, and mainly consider the simplest one – the Euler-Maruyama approximation. We also look at the Milstein approximation, as well as discuss the convergence criteria and errors occurring in simulations.

### 2.3.1 Numerical schemes for solving Itô stochastic differential equations

Time discrete methods means that the Itô process is approximated for a given time discretization

$$t_0 = \tau_0 < \tau_1 < \cdots < \tau_n < \cdots < \tau_N = T, \quad (2.3.2)$$

of the time interval  $[t_0, T]$ . We only consider equidistant discretization times, such that the  $n$ th time interval  $\Delta t$  is equal for all increments;

$$\tau_{n+1} - \tau_n = \Delta_n = \Delta t = \frac{T - t_0}{N}. \quad (2.3.3)$$

The approximations, being time discrete, only give the approximated values at the discretization times (2.3.2). Any other required values, intermediate values, must be found through estimation based on the discrete values we have obtained, i.e., interpolation.

### The Euler-Maruyama Approximation

The simplest time discrete approximation of an Itô process is the Euler-Maruyama Approximation, or sometimes referred to as simply the Euler approximation or the Euler method. The approximation is a time-discrete process  $Y = \{Y(t), t_0 \leq t \leq T\}$  that satisfy the iterative scheme

$$Y_{n+1} = Y_n + a(\tau_n, Y_n)(\tau_{n+1} - \tau_n) + b(\tau_n, Y_n)(W_{\tau_{n+1}} - W_{\tau_n}), \quad (2.3.4)$$

for  $n = 0, 1, 2, \dots, N - 1$  with initial value  $Y_0 = X_0$ , and here given for an arbitrary time step size (Platen, 2010, p. 246). The process is defined on the time interval (2.3.2), and we write  $Y_n = Y(\tau_n)$  for the value of the time-discrete Euler-Maruyama approximation at the discretization time  $\tau_n$ . The last term of the approximation contains the difference between two preceding increments of the Wiener process, the stochastic process defined in more detail in Section 2.2.2. Considering that we will only look at equidistant time steps, we simplify the notation of the scheme to

$$Y_{n+1} = Y_n + a \Delta t + b \Delta W_n, \quad (2.3.5)$$

where  $\Delta W_n = W_{\tau_{n+1}} - W_{\tau_n}$

$$\Delta W_n = W_{\tau_{n+1}} - W_{\tau_n} \quad (2.3.6)$$

is an increment of the Wiener process over the subinterval  $\tau_n \leq t \leq \tau_{n+1}$ . More generally, in multiple dimensions,  $d = 1, 2, \dots$ , the  $k$ th component of the Euler-Maruyama scheme is given by

$$Y_{n+1}^k = Y_n^k + a^k \Delta t + b^k \Delta W^k, \quad (2.3.7)$$

for component  $k = 1, 2, \dots, d$ . We remind ourselves that the increments are independent identically distributed random variables,  $\Delta W_n \sim \mathcal{N}(0, \Delta t)$ , or in other words have the properties

$$E[\Delta W_n] = 0, \quad \text{and} \quad E[(\Delta W_n)^2] = \Delta t. \quad (2.3.8)$$

On a final side note about the scheme, we have that the stochastic scheme can easily be translated to the deterministic Euler scheme by letting the random term be zero, or essentially let the diffusion coefficient  $b \equiv 0$ . This result in an ordinary differential equation  $dX_t = a(t, X_t)dt$  that can be numerically integrated using the deterministic Euler scheme (2.2.16), as in the example in Figure 2.2.4.

### The Milstein Approximation

The second approximation process we shall introduce is the Milstein Approximation. It is similar to the Euler-Maruyama approximation but has an additional term from the Wagner-Platen series (that we return to in Section 2.3.2). The recursive discrete-time approximation scheme is given by

$$Y_{n+1} = Y_n + a \Delta t + b \Delta W_n + \frac{1}{2} b b' \{(\Delta W_n)^2 - \Delta t\}. \quad (2.3.9)$$

The additional term contains the derivative of the diffusion coefficient,  $b' = \partial_x b$ , as well as a random contribution from the squared Wiener process increment,  $(\Delta W_n)^2$ .

An important observation is that not only for  $b \equiv 0$ , the scheme is equivalent to the Euler-Maruyama approximation, but also for any function  $b$  that gives  $\partial_x b \equiv 0$ . In other words, any constant diffusion coefficient will result in the two schemes being equivalent. Even a diffusion constant that varies in time,  $b = b(t)$ , will still satisfy  $\partial_x b = 0$ , and thus give the same approximation as the Euler-Maruyama scheme.

Finally, we also mention that in the multidimensional case,  $d = 1, 2, \dots$ , the  $k$ th component of the Milstein scheme is given by

$$Y_{n+1}^k = Y_n^k + a^k \Delta t + b^k \Delta W_n^k + \frac{1}{2} \left( \sum_{l=1}^d b^l \frac{\partial b^k}{\partial x^l} \right) \{(\Delta W_n^k)^2 - \Delta t\}, \quad (2.3.10)$$

for  $k = 1, 2, \dots, d$ .

### 2.3.2 Accuracy and convergence in numerical simulations

To simulate a trajectory, we first choose a numerical scheme and decide the time discretization. Following a recursive formula (in our case), we start with the initial value  $Y_0 = X_0$  at  $t_0$  and generate the next value recursively. One often assumes that with a fine enough time discretization, the resulting trajectory must have close resemblance to the Itô process. However, there will always be some errors in numerical solutions of stochastic differential equations (SDEs). Our goal will be to get resulting trajectories of the approximating process that *converge* towards the Itô process.

The errors can originate from both systematic and statistical sources as well as round-off errors, hence to numerically solve for accurate and converging solutions, the errors should be determined and minimized, at least to a level of acceptable accuracy. In that way, we can ensure that our results make sense and are actually approximating the process we are interested in. Choosing the approximating scheme of a process depends on which convergence criteria we wish to be most effective. This again depends on what property we are interested in approximating, whether it be a good pathwise approximation or some functional of the Itô process. Finding the real objective of the simulation is therefore important before choosing a numerical scheme to apply to the problem. The numerical schemes presented; the Euler-Maruyama approximation (2.3.5) and the Milstein approximation (2.3.9), are yet to be categorized by their convergence properties. We shall have a look at errors, convergence criteria, and the order of these two schemes in the following subsections.

#### Absolute error

One of the most basic ideas is that by the end of a simulation, the final position or state of the process should be as close to the real process as possible. Say that an Itô process at the final time  $T$  has the value  $X_T$ , then the absolute error in the approximation solution  $Y_N$  is defined as

$$\epsilon(\Delta t) = \mathbb{E}[|X_T - Y_N|] \leq \sqrt{\mathbb{E}[|X_T - Y_N|^2]}, \quad (2.3.11)$$

for  $T = t_0 + N\Delta t$  (Kloeden, 1992). A relatively low absolute error suggests a closeness of the sample paths of the actual Itô process  $X$  and the approximation  $Y$  at time  $T$ . However, we do not always know the Itô process that is the solution of the SDE we are trying to solve. In that case we do not have a solution  $X_T$  to calculate the absolute error (2.3.11) from. An alternative approach is assuming that the absolute error must be small as  $\Delta t \rightarrow 0$ . In other words, calculate a high-resolution approximation  $Y$  and assume that it is "close enough" to the true process solution  $X$ . Then, we calculate the absolute error relative to this sample path. The fine resolution solution must use a sufficiently small time step  $\Delta t_{min}$ , and we use the notation  $Y_{N'}$ , for the simulation using  $N' = T/\Delta t_{min}$  time steps to calculate the final time instant  $T$  (for  $t_0 = 0$ ). Thus, the approximating absolute error is written

$$\epsilon(\Delta t) = \mathbb{E}[|Y_{N'} - Y_N|], \quad (2.3.12)$$

with  $N = N(\Delta t)$  for a fixed  $T$ . For now, we will return to the true solution of the Itô process  $X_T$ , but knowing that we might have to interchange it for  $Y_{N'}$  wherever we lack a known solution.

### The Strong Convergence Criterion

The strong convergence of a numerical scheme is defined (Kloeden, 1992, p. xxiv) as follows:

*An approximating process  $Y$  converges in the strong sense with order  $\gamma \in (0, \infty]$  if there exists a finite constant  $K$  and a positive constant  $\delta_0$  such that*

$$\mathbb{E}[|X_T - Y_N|] \leq K(\Delta t)^\gamma \quad (2.3.13)$$

*for any time discretization with maximum step size  $\Delta t \in (0, \delta_0)$ .*

In essence, the criterion defined above says something about the closeness in path between the approximation and the Itô process, and in what order it is affected by the time step size.

### The Weak Convergence Criterion

The closeness in path of an approximation to an Itô process is not always the most relevant criteria for a good approximation. In some cases, we are more interested in the probability distribution of a random variable, than the actual pathwise behavior. In this case, we are interested in, e.g., the expectation value  $\mathbb{E}[X_T]$  or the second moment  $\mathbb{E}[(X_T)^2]$ , and thus we need a good approximation of the probability distribution of  $X_T$ .

In practical cases, our interest in the expectation might come from the desire to look at the system as a whole. Instead of looking at a specific particle moving in a specific path, we are interested in, e.g., the center of mass of the particles if many were to be transported in the same system. We define, as in (Kloeden, 1992, p. xxv), the weak convergence criterion to be:

*The time discrete approximation  $Y$  converges in the weak sense with order  $\beta \in (0, \infty]$  if for any polynomial  $g$  there exists a finite constant  $K$  and a positive constant  $\delta_0$  such that*

$$|\mathbb{E}[g(X_T)] - \mathbb{E}[g(Y_N)]| \leq K(\Delta t)^\beta, \quad (2.3.14)$$

*for any time discretization with maximum step size  $\Delta t \in (0, \delta_0)$ .*

Under the weak and strong convergence criteria we will further look at errors in solving SDEs in Section 2.4.2.

### Classifying numerical methods

The numerical schemes presented have not yet been classified with regards to their convergence. To do that, we go beyond just the formulas presented. They originate from and can be classified by comparing them to parts of the stochastic Taylor formula, or Wagner-Platen formula (see, e.g., (Platen, 2010, p. 187)). Following the Wagner-Platen formula, we find the following representation for the Itô process:

$$\begin{aligned} X_t = X_{t_0} + a(X_{t_0}) \int_{t_0}^t ds + b(X_{t_0}) \int_{t_0}^t dW_s \\ + b(X_{t_0})b'(X_{t_0}) \int_{t_0}^t \int_{t_0}^{s^2} dW_{s_1} dW_{s_2} + R. \end{aligned} \quad (2.3.15)$$

This stochastic Taylor expansion can be used to form time-discrete Taylor approximations when applied at successive time steps.  $R$  is the remainder consisting of higher-order multiple stochastic integrals involving the process itself, the coefficients, and their derivatives.

First of all, truncating the Wagner-Platen formula after the Itô integral gives us the Euler-Maruyama scheme (2.3.5). The Euler-Maruyama scheme converges in the strong sense<sup>4</sup> with  $\gamma = 0.5$ , and usually converges with weak order  $\beta = 1$ . The stochastic Taylor expansion (2.3.15) gives a strong convergence of 0.5 since the increments  $\Delta W_n$  of the Wiener process are of root mean square order  $(\Delta t)^{1/2}$ . In the deterministic case, for zero diffusion, the Euler approximation for ordinary differential equations (ODEs) converges with order 1.

Continuing with the Wagner-Platen formula (2.3.15), including the next term, we obtain the Milstein scheme (2.3.9). To show this, we can rewrite the double integral as

$$\int_{t_0}^t \int_{t_0}^{s_2} dW_{s_1} dW_{s_2} = \frac{1}{2} \{(\Delta W_n)^2 - \Delta t\}. \quad (2.3.16)$$

The Milstein approximation converges with strong order  $\gamma = 1$ , under the assumption that  $E[(X_0)^2] < \infty$ , and specific smoothness and continuity conditions on the coefficients<sup>5</sup>. In a weak sense, the Milstein approximation converges with order  $\beta = 1$ . We sum up the order of convergence for the two schemes, for a general case:

<u>Euler-Maruyama:</u>	<i>Strong convergence</i> $\sim (\Delta t)^{\frac{1}{2}}$	<i>Weak convergence</i> $\sim (\Delta t)^1$
<u>Milstein:</u>	<i>Strong convergence</i> $\sim (\Delta t)^1$	<i>Weak convergence</i> $\sim (\Delta t)^1$

The link between the two schemes in the case of a constant diffusion coefficient, briefly commented on in section 2.3.1, will also change the apparent order of a scheme. An example is when using the Euler-Maruyama scheme with a diffusion coefficient that satisfies  $\partial_x b = 0$ , in other words, a constant diffusion coefficient. The Euler-Maruyama scheme is then equivalent to the Milstein approximation, as the last term in (2.3.9) is zero. In other words, for a constant  $b$ , the Euler-Maruyama has the same order of strong convergence as the Milstein scheme;  $\gamma = 1$ . The same applies for diffusion of the form  $b(x, t) = b(t)$ .

The Euler-Maruyama scheme gives good numerical results for near-constant drift and diffusion coefficients  $a$  and  $b$ , and is often computationally more efficient as it has fewer terms and the same order of weak convergence (Platen, 2010). Of course, if the strong convergence is of interest the Milstein scheme is preferred, but the numerical stability of the method is worse than of the Euler-Maruyama scheme. A review of other approximation schemes of higher order can be found in, e.g., Kloeden (1992), Gräwe (2011), and Gräwe et al. (2012).

## 2.4 MONTE CARLO SIMULATIONS

Stochastic numerical methods and Monte Carlo methods (Sauer, 2014; Landau, 2009) use stochastic techniques (random numbers) to solve problems that might be deterministic in principle. Monte Carlo algorithms can be used to solve computational problems involving stochastic differential equations (SDEs), which scarcely have analytical solutions, by performing multiple simulations to find the probability of specific outcomes. It is an efficient numerical method, having evolved from a "last resort" solution to a leading methodology, much due to its simplicity and general applicability (Kroese et al., 2014). Even though some distinguish between a simulation, a Monte

<sup>4</sup>Under "*Lipschitz and bounded growth conditions on the coefficients  $a$  and  $b$* " (Kloeden, 1992, p. xxvii).

<sup>5</sup>Conditions are that  $a$  and  $b$  are twice continuously differentiable, and that  $a, a', b, b'$ , and  $b''$  satisfy a uniform Lipschitz condition following Kloeden (1992, p. xxviii).

Carlo (MC) method and an MC simulation<sup>6</sup>, we will in this thesis use Monte Carlo as a wide term to address the family of methods using repetition of simulations to discover long run outcomes of events (see Sawilowsky (2003)). We introduce the Monte Carlo method and associated errors in the following, and discuss the tradeoff between step size and sample size, as well as the probability density estimation from finite size samples from Monte Carlo simulations.

### 2.4.1 Introduction to the Monte Carlo Simulation Method

From our discussion about deterministic and stochastic models in subsection 2.2.3, we know that the result from running a random model is intrinsically random. While the deterministic model will have the same result every time it is run, the stochastic result can be considered a "sample" from the true distribution of the random variable. The Monte Carlo method solves a problem by repeatedly simulating the same stochastic model. The class of approaches that are described as Monte Carlo methods are mainly used in problems involving *optimization*, numerical integration (*estimation*), and generating draws from a probability distribution (*sampling*) (Kroese et al., 2014).

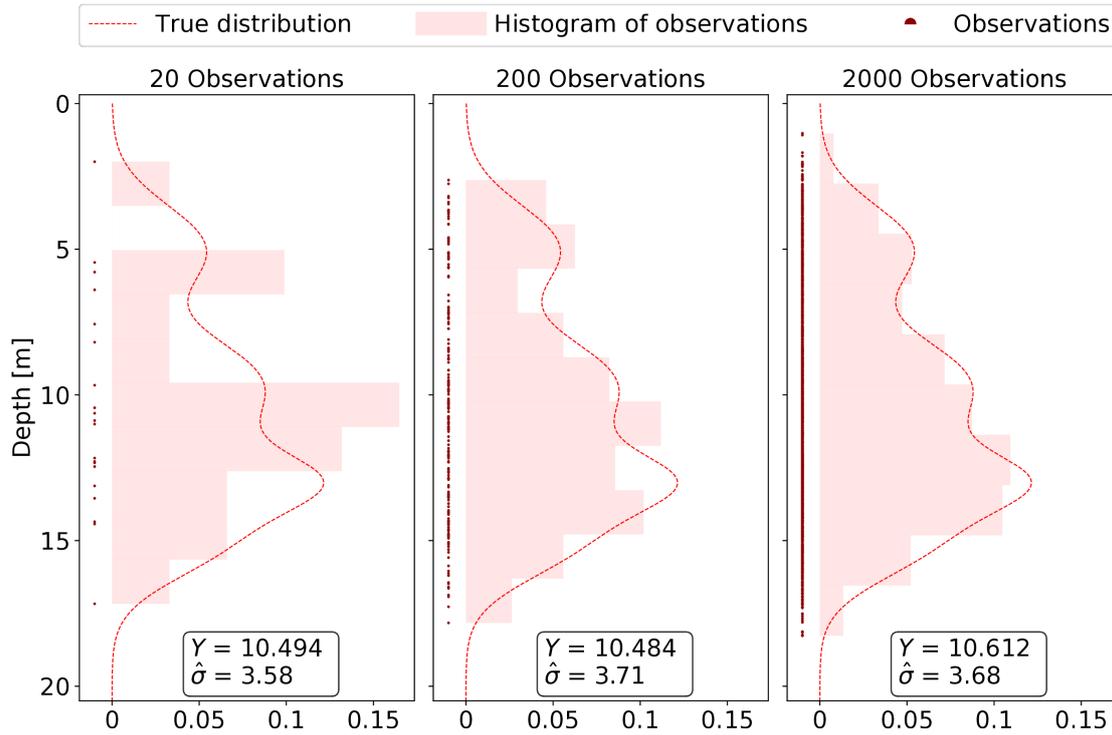
We will demonstrate the concept of Monte Carlo simulation with a relatable and intuitive example: Say that you have a fisherman sitting in a boat, and he does not know at what depth in the water he is more likely to catch a fish. He is interested in catching a lot of fish, and knows that if he registers every catch he might get a better idea of the probability as a function of depth, and hence find where it is expected that he will catch the most fish. First, he registers 20 fish, but is unable to say where the probability is higher. Even though there are depths where he has caught more fish than others - who is to say that that is not just a coincidence? See Figure 2.4.1 below. He wants more certainty, and he intuitively fishes some more, and stops after having caught 200 fish. The pattern that emerges is what he has been looking for! He sees that the probability of catching fish is larger for some areas, and he now thinks he knows where he will get most fish by estimating a mean of the observations. If he had continued the experiment, he could have obtained an even clearer idea, and with less uncertainty (see Figure 2.4.1).

In this example of the fisherman, the probability distribution was estimated by making a histogram out of the observations, where the peak described the area of highest probability for catching fish. This is in essence equivalent to estimating the mean ( $Y$  in Figure 2.4.1) of the independent and identically distributed (iid) random variables  $Z_i$ , that are the samples from the true distribution. The mean value of the observations works as an estimate of the parameter  $\mu$  of the true distribution, and as shown in Figure 2.4.1 the variance  $\sigma$  can also be estimated ( $\hat{\sigma}$ ). The technique of sampling repeatedly from an unknown distribution, for the purpose of estimating said distribution, is a form of Monte Carlo simulation. We will have a closer look at how well such an estimation works in the following, but as we have seen with this short example – the more samples we had, the better the estimation.

### 2.4.2 Convergence and accuracy

It is important to remember that Monte Carlo methods only provide an approximation of the actual solution, and hence the approximation error is essential to consider when analyzing the solutions. We will shortly claim that very large samples are necessary to acquire low variance estimates (i.e., low errors). There have been efforts toward reducing sample sizes, including the main objective behind variance-reducing sampling techniques (Wilson, 1984; L'Ecuyer, 1994), but that will not be highlighted in this thesis.

<sup>6</sup>Sawilowsky (2003) distinguishes between a simulation, an MC method, and an MC simulation. The simulation itself is described as an abstract model or representation of reality. He further distinguishes the MC method, and MC simulation, by stating that the method is "*use of stochastic techniques to solve... a deterministic problem*" (Moshman (1967, p. 250) via Sawilowsky (2003)), and the MC simulation is a combination of simulations with many repetitions (MC method) to obtain statistical properties of a system.



**Figure 2.4.1:** The distribution of fish as a function of depth estimated using Monte Carlo simulations (number of simulations described as observations). The true distribution (dashed line) has an expected value  $\mu = 10.60$  and a variance  $\sigma = 3.70$ . The sample mean  $Y$  is shown in each subplot, as an estimation of the mean of the true distribution, along with the variance of the samples, named  $\hat{\sigma}$ . The estimation improves with number of observations, a concept that is further discussed in subsection 2.4.2.

### Estimating a sample mean

First, we shall look at the error that comes with estimating a mean; the convergence rate of a Monte Carlo simulation. We will look at what rate the estimation error decreases as the number of points  $N$  used to estimate grows. Consider the independent, identically distributed (iid) random variables  $Z_i$  corresponding to random samplings from a distribution (as with the caught fish in the example above). The expectation value of the samples is  $E[Z_i] = \mu$ , the mean of the distribution. Then the mean value of the simulation is the expectation value of the random variable  $Y = (Z_1 + Z_2 + \dots + Z_N)/N$ ,

$$E[Y] = E\left[\frac{Z_1 + Z_2 + \dots + Z_N}{N}\right] = \frac{\sum_{i=1}^N \mu}{N} = \frac{N\mu}{N} = \mu. \quad (2.4.1)$$

The variable  $Y$  is called the *sample mean*, and will be different from the true mean of the distribution it intends to estimate; the distribution of  $Z_i$ .  $Y$  has a variance of

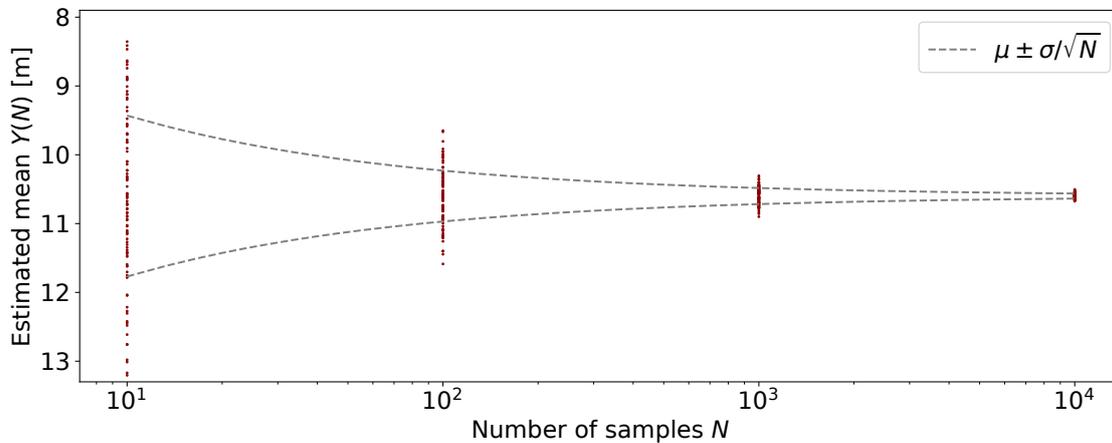
$$E\left[\left(\frac{Z_1 + \dots + Z_N}{N} - \mu\right)^2\right] = \frac{1}{N^2} \sum_{i=1}^N E[(Z_i - \mu)^2] = \frac{1}{N^2} N\sigma^2 = \frac{\sigma^2}{N}, \quad (2.4.2)$$

where  $\sigma^2$  is the variance of the iid random variables  $Z_i$ . Therefore, we see that the standard deviation of  $Y$  decreases as  $\sigma/\sqrt{N}$ . Thus, the error in estimating the mean will decrease with the number  $N$  of realizations as

$$\text{Error} \propto \frac{1}{\sqrt{N}}. \quad (2.4.3)$$

An improvement of the accuracy of the mean estimation is in other words expensive, as it demands a large number of samples for relatively small increases in accuracy. E.g., if you want to improve the accuracy by one digit, i.e., reduce  $\sigma$  by a factor of 10 we would have to increase the number of samples  $N$  by  $10^2$ .

We return to the example of the fisherman, to show how the estimate improves with the total number of simulations in a specific application. The distribution of fish as a function of depth, given by the dashed line in Fig. 2.4.1, has a known expectation value  $\mu$  and standard deviation  $\sigma$ . We try to estimate the expectation value for different numbers of samples from the true distribution. In Fig. 2.4.2 we have simulated for four different sample sizes  $N$ , and have run these simulations 100 times for each  $N$ . With trend lines plotted as described in the figure text, the figure clearly shows the convergence  $\sim 1/\sqrt{N}$  as expected in this Monte Carlo simulation.



**Figure 2.4.2:** Scatter plot of sample means of  $N$  samples, 100 simulations, and trend line showing the true mean of the distribution with one standard deviation. The expectation value of the true distribution (shown in Figure 2.4.1) from which the samples are collected is  $\mu = 10.6$  and the standard deviation is 3.7.

### Statistical and systematic error

Monte Carlo simulations use random variables to find the probability of specific outcomes, resulting in two intrinsic errors that we need to consider properly (Komori et al., 1994). The error in estimating the mean of a distribution, discussed above, is also called *statistical error*  $\epsilon_{stat}$ , and depends on the total number of simulations performed. The central limit theorem (e.g., Kadanoff (2000, Chapter 3.6)) concludes that, for a large number of independent simulations  $N$ , the statistical error  $\epsilon_{stat}$  becomes asymptotically Gaussian with mean zero and variance  $Var[\epsilon_{stat}] = \frac{1}{N} Var[Y(T)]$ , as we saw above (Eq. (2.4.2)).

The other error of interest is the *systematic error*  $\epsilon_{sys}$ , or *numerical error*, which is dependent on the accuracy of the numerical model and discretization used. In this case, the error comes from the fact that we are no longer sampling from the true distribution. In the fish distribution example (see Section 2.4.1) we have sampled directly from the true distribution, and thus the systematic error is zero.

In applications, a relatively large systematic error can be hard to discover when the true solution is unknown. However, the systematic error most often originates from too large time steps  $\Delta t$ . As discussed in Section 2.3.2 concerning time discrete approximations, we consider that if the

numerical method is correctly implemented and the correct conditions apply, then we assume that using a small enough time step will ensure a low systematic error.

### Tradeoff between step size and sample size

In simplified terms, we can express the total error of a Monte Carlo simulation as

$$\epsilon = \epsilon_{stat}(N) + \epsilon_{sys}(\Delta t), \quad (2.4.4)$$

where  $N$  is the sample size, and  $\Delta t$  is the time step size. As we generally need to consider both errors, we must balance the increase in number of Monte Carlo simulations and the reduction in step size, to ensure accuracy but also computational efficiency. The errors are accordingly also called *discretization error* ( $\epsilon_{sys}$ ) and *Monte Carlo averaging error* ( $\epsilon_{stat}$ ) (Detemple et al., 2006). A trade-off between these two errors is presented by Duffie & Glynn (1995), taking into account the weak convergence criteria. We have previously (Section 2.3) looked at both the Euler-Maruyama approximation and the Milstein approximation, which both have weak order of convergence 1. When using these numerical schemes to approximate a solution of a stochastic differential equation (SDE), it is done on a specific time interval  $[0, T]$ . This suggests that using Monte Carlo for a  $d$ -dimensional problem requires roughly  $d^2TN/\Delta t$  calculations, where  $N$  is the sample size and  $\Delta t$  is the time step (e.g., Platen and Bruti 2010, page 481). Of course, simply decreasing dimension  $d$  or the length of simulation interval  $T$ , will decrease the number of calculations needed. However, if we assume them given, we can purely concern ourselves with the relationship between step size and sample size. It is suggested by Duffie & Glynn (1995) that under suitable conditions, with a scheme of weak order  $\beta$ , the efficient choice is to increase  $N$  at the order  $(\Delta t)^{-2\beta}$ . In the case of both the Euler-Maruyama and the Milstein approximation, this implies that a halving of the time step size, should be accompanied by an increase of the number of simulations by a factor 4.

### Monte Carlo simulation using time discrete approximations

An example of a Monte Carlo simulation is sampling a solution from a time discrete approximation, e.g., the Euler-Maruyama approximation (see Section, 2.3.1) *repeatedly*. The numerical scheme approximates the Itô process  $X = \{X_t, t_0 \leq t \leq T\}$ , which is the solution of the stochastic differential equation (SDE)

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t.$$

The approximation is the continuous time stochastic process  $Y = \{Y(t), t_0 \leq t \leq T\}$ , satisfying the iterative scheme given by

$$Y_{n+1} = Y_n + a\Delta t + b\Delta W,$$

for equidistant time steps  $\Delta t$ , and increments of the Wiener process  $\Delta W$ . Thus, applying the theory behind Monte Carlo in this case involves running several realizations of the scheme up to the time  $\tau'$ , where the random Wiener process ensures a different result each time. Say we run  $N$  simulations, then we have  $N$  sample paths. Then, depending on our interest, we might want to estimate the mean value or perhaps the distribution from which the samples are drawn.

As an example, we can apply the Euler-Maruyama approximation to an SDE repeatedly. The SDE presented in subsection 2.2.4 is an excellent example due to its relevance in this thesis: the SDE whose Fokker-Planck equation is the Advection-Diffusion Equation (ADE). We have that  $a = u + \partial_x D$  and  $b = \sqrt{2D}$  as stated in Eq. (2.2.28) by comparing the ADE and the Fokker-Planck equation. We look at a system with constant diffusion and drift

$$D = D_0 \quad \text{and} \quad u = u_0. \quad (2.4.5)$$

for simplicity. Then, the approximating process that we can use to simulate the Itô process is

$$Y_{n+1} = Y_n + u_0 \Delta t + \sqrt{2D_0} \Delta W. \quad (2.4.6)$$

Since the numerical scheme uses random numbers in the modeling of Brownian motion, every outcome is different.

For completion in this section, we mention how the same system can also be solved with, e.g., the Milstein approximation as presented in subsection 2.3.1. The coefficients for the ADE are found in Eq. (2.2.28), and  $D$  and  $u$  are given above in Eq. (2.4.5). Thus, the approximation in Eq. (2.3.9) gives the following scheme we solve recursively

$$Y_{n+1} = Y_n + (u + \partial_x D) \Delta t + \sqrt{2D} \Delta W_n + \frac{1}{2} \sqrt{2D} \partial_x D b' \{(\Delta W_n)^2 - \Delta t\}. \quad (2.4.7)$$

However, due to the constant system (2.4.5), the derivative of the diffusion coefficient is zero,

$$\partial_x D = 0 \quad \text{and} \quad \frac{\partial}{\partial x} b = \frac{\partial}{\partial x} \sqrt{2D} = 0. \quad (2.4.8)$$

Hence, as Eq. (2.4.7) is reduced to Eq. (2.4.6); the two schemes are in essence identical.

### 2.4.3 Kernel Density Estimation

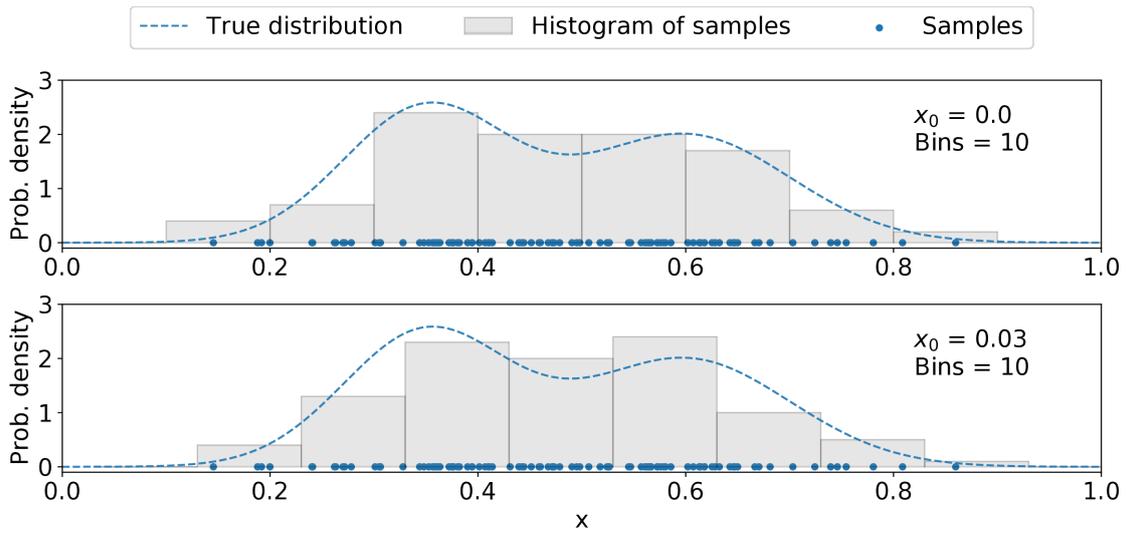
Kernel density estimation (KDE) is a technique to estimate a probability density function  $f(x)$  from its independent and identically distributed samples  $\{X_i\}$ . The contrasting, typically parametric technique, involves obtaining the best estimator  $\hat{\theta}$  of  $\theta$  for a given density family  $f(x|\theta)$ , e.g., such as the Normal family  $N(\mu, \sigma^2)$  where  $\theta = (\mu, \sigma^2)$ . Contrary to this type of parametric estimation, the kernel density estimation is what is called a nonparametric estimation method. It means that it estimates the distribution itself, i.e.  $\hat{f}(x)$  of  $f(x)$ . In this manner, no prior knowledge or any underlying distribution needs to be relied upon, and the estimation is driven by the structure in the data alone. In other words, the technique is highly applicable, as it can uncover important patterns in the data while filtering noise and ignoring irrelevant details (Gramacki, 2018, page vii). We shall look at a particular type of nonparametric density estimation that uses *kernels* – symmetric weighting functions of each sample – to estimate smooth probability density functions. The smoothness is largely controlled by a parameter called the *bandwidth*, i.e., the width of the kernel function. To introduce kernel density estimation properly, we take a detour by explaining the simplest nonparametric density estimator: the histogram, before discussing different kernels, error criteria, and finally bandwidth selection.

#### The histogram

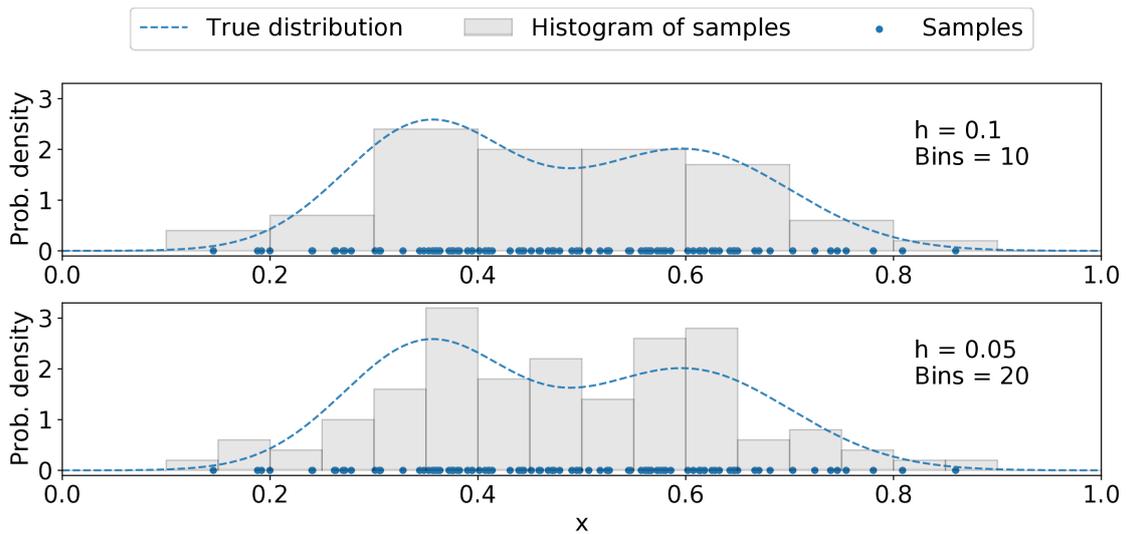
The simplest form of nonparametric density estimation is the widely known histogram. Typically, one divides the sample space into a number of bins of constant width, and the density is approximated by counting the number of samples within each bin. We saw an example of such approximation in Figure 2.4.1, where the probability density function was approximated, and from it, the distribution mean. To approximate a probability density function using a histogram, we start by sampling from the true (usually unknown) distribution  $f(x)$ . The number of data points  $X_0, X_1, \dots, X_n$  falling into the  $j$ -th bin  $B_j$ , are denoted  $\#\{X_i \in B_j\}$ . Thus, we estimate the probability density function by the estimator

$$\hat{f}(x) = \frac{\#\{X_i \in B_j\}}{nh} = \frac{k}{nh}, \quad (2.4.9)$$

for every  $x \in B_j$ . The approximation requires two parameters to be defined; the bin width  $h$  and the bin origin (lower boundary of bin  $B_0$ ). From this apparent simplicity of the method, also



**Figure 2.4.3:** Histograms with bin width  $h = 0.1$  for 100 samples (blue dots) from a bimodal distribution (dashed line). The difference in bin origin  $x_0$  gives drastic differences in estimations.

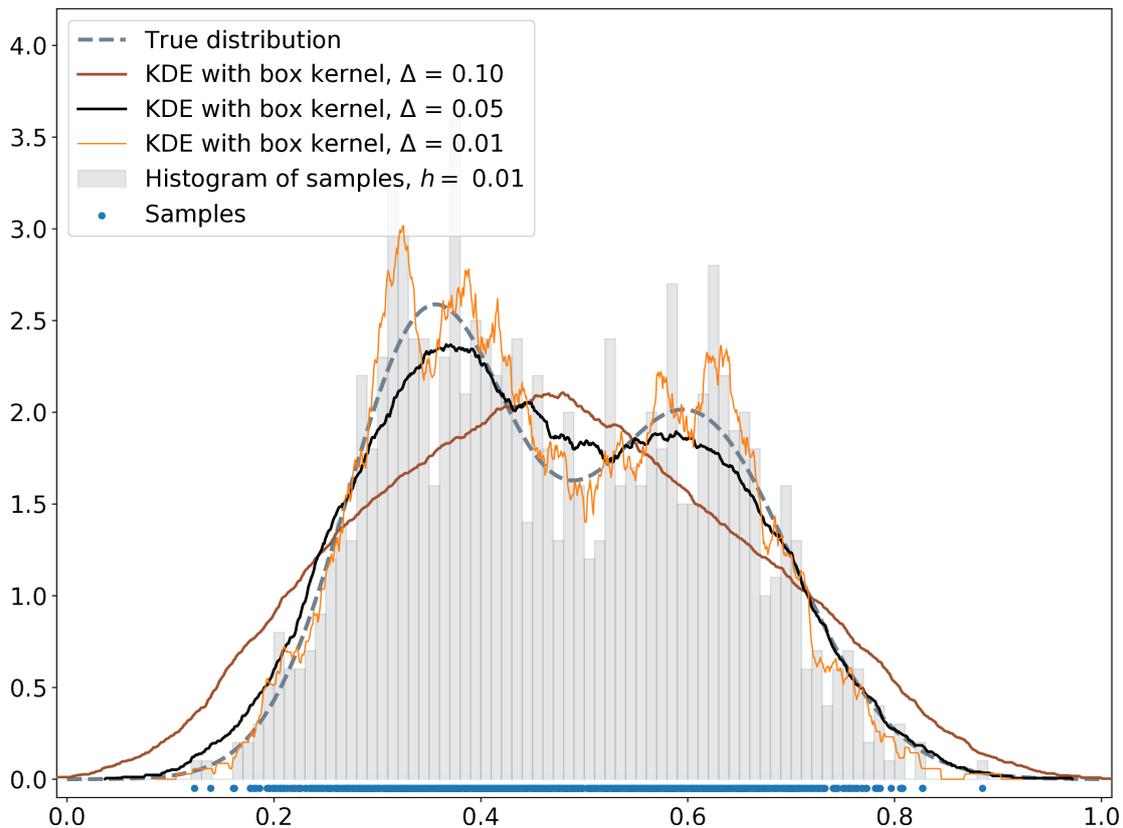


**Figure 2.4.4:** Histograms with bin origin  $x_0 = 0$  for 100 samples (blue dots) from bimodal distribution (dashed line). The difference in bin width  $h$  gives drastic differences in estimations.

originates some disadvantages. First, both the bin width and the choice of bin origin have a large influence on the estimation. Figure 2.4.3 shows how offsets of the bin origin change the shape of the final density estimate for constant bin width. Figure 2.4.4 shows on the other hand how varying bin widths affect the final density estimate. Of course, the discontinuity in the estimated density function is almost certainly not related to the underlying density distribution, and is (as shown in the figures presented) a result of the choices of bin origin and bin width, and a natural feature of the histogram.

### Kernel Density Estimation

There are ways to get around the disadvantages of the histogram presented above; both the loss of information due to "binning", and the discontinuity in the density function estimates. Kernel



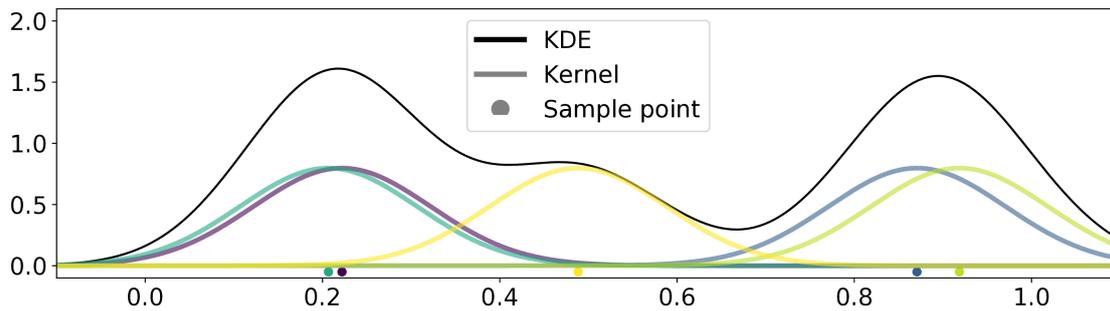
**Figure 2.4.5:** The distribution of 1000 random samples (blue dots) from a bimodal Gaussian distribution (dashed line) are estimated using both histogram and KDE. The histogram has bin origin  $x_0 = 0$ , and bin width  $h$ . Three kernel density estimations are calculated using three different bandwidths,  $\Delta$ .

Density Estimation (KDE) (Simonoff, 1996; Silverman, 1986; Scott, 1992; Gramacki, 2018) removes both the restriction of bins and can employ smooth functions to create continuous density estimates. It is a widely used tool for data smoothing of finite size samples.

If we picture the histogram in the previous subsection as a stack of blocks, one block for each data point, then the histogram is revealed by stacking these blocks in the appropriate grid (bins). However, if we do not restrict ourselves to the grid (bins), we can center the blocks on the data point it represents. By summing up all potential contributions, we find a new estimate of the distribution. The estimate is unaffected by the choice of grid (bin origin); it is driven by the underlying data. The estimator, analogous to the histogram estimator (2.4.9), is

$$\hat{f}(x) = \frac{\#\{X_i \in (x-h, x+h)\}}{2Nh} = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x-X_i}{h}\right), \quad (2.4.10)$$

where  $K$  is a squared density function often called *tophat* or *box-kernel*, and  $h$  is the width of the function. This is the idea behind KDE – it works by passing a *kernel function* over each sample point. A kernel function is restricted to being a non-negative, real-valued, integrable, and symmetric function, that must also satisfy the normalization requirement,  $\int_{-\infty}^{\infty} K(u)du = 1$ . The requirements ensure that the KDE results in an actual probability density function. In the case above, a box-kernel is placed over each point it represents, resulting in a discontinuous function estimate as seen in Figure 2.4.5 for a finite data sample. Some of the estimates are better than others, in this case they differ only due to the choice of *bandwidth*,  $\Delta$ , of the kernel. The parameter



**Figure 2.4.6:** Shows a Gaussian kernel density estimate, and how a Gaussian curve contributes in each point to the total estimation curve. The resulting estimate functions as a nonparametric model of the distribution of points (in this case uniform distribution). In above simulation a bandwidth of  $\Delta = 0.1$  has been used, and the kernels are resized to  $1/5$  of original value for a better visual presentation.

referred to as bandwidth, or also smoothing parameter, is the scaling parameter of the kernel function setting the standard deviation of the distribution.

However, KDE is a known tool for data smoothing, and multiple smoother kernels can be applied to construct a representation of the sample data points. The blocks considered need not be square boxes, but any kernel function  $K$  that satisfy the requirements. There are a number of different kernels, some of the more commonly used functions are presented in Figure 2.4.7. A typical smooth kernel is the widely used Gaussian kernel. Figure 2.4.6 shows a number of data points, and the overlying kernels representing each point. The kernels are then added up to estimate a density function as seen in the figure – a smooth estimate of the density. When estimating a distribution of sample points using KDE using a particular kernel and a set number of samples, only one parameter must be specified: the bandwidth. As with the bin width in the histogram, the kernels' width will drastically change the representation of data. We discuss the smoothing parameter further in subsection 2.4.3.

### Error criteria for density estimates

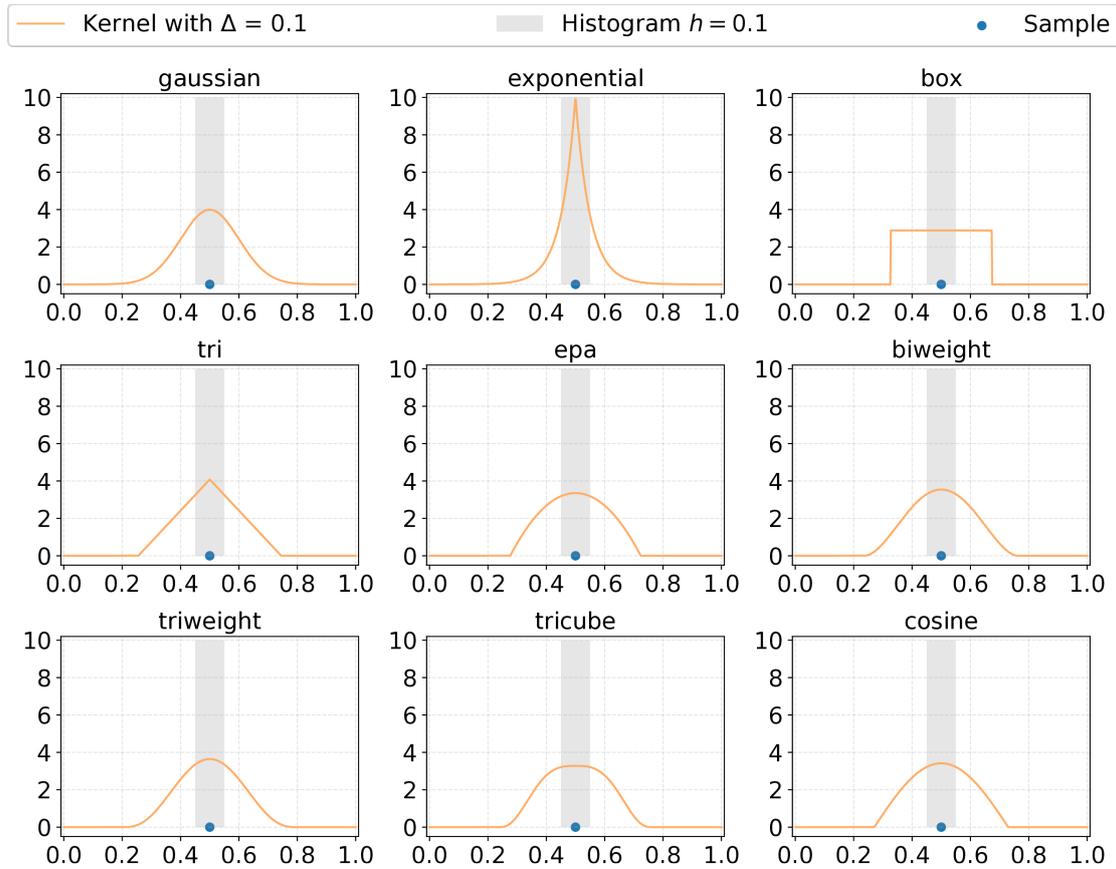
For optimizing the method of density estimation, we need some criteria to be able to measure an error. For a typical density estimation problem, we can define the mean-square error (MSE) criterion

$$\text{MSE}\{\hat{f}(x)\} = \text{E}[\hat{f}(x) - f(x)]^2 = \text{Var}\{\hat{f}(x)\} + \text{Bias}^2\{\hat{f}(x)\} \quad (2.4.11)$$

where  $\text{Bias}\{\hat{f}(x)\} = \text{E}[\hat{f}(x)] - f(x)$ . This criterion treats the density estimator as a point-wise estimator. It might be more applicable to define a criterion that considers the entire density surface. Examples of such global criteria are the  $L_\infty$  and  $L_1$  norms (see, e.g., Scott (1992, p. 38)), and the more applicable  $L_2$  norm (or Euclidean norm) defined as

$$\text{ISE} = \int [\hat{f}(x) - f(x)]^2 dx. \quad (2.4.12)$$

It is often referred to as the *integrated squared error (ISE)*. The criterion depends on both the true unknown distribution  $f(x)$ , the estimator used  $\hat{f}(x)$ , and the sample size. The ISE itself is a random variable, and thus it is more convenient to define a criterion that averages over realizations. For this reason, the mean of the ISE is another criterion; the *mean integrated squared error (MISE)*:



**Figure 2.4.7:** Typical kernels of bandwidth  $\Delta$  used in kernel density estimation, shown along with a sample point and the corresponding histogram of bin width  $h$ .

$$\begin{aligned}
 \text{MISE} &= E[\text{ISE}] = E \left[ \int [\hat{f}(x) - f(x)]^2 dx \right] \\
 &= \int E \left[ \hat{f}(x) - f(x) \right]^2 dx = \int \text{MSE}\{\hat{f}(x)\} dx \equiv \text{IMSE},
 \end{aligned} \tag{2.4.13}$$

or also named the *integrated mean squared error (IMSE)*. Scott (1992, p. 38) calls attention to the fact that it can be interpreted as both a measure of average global error and the accumulated pointwise error. The MISE criterion could be modified by including a weight function, which gives a criterion that is even more specified for a particular optimization of an estimator.

### Smoothing parameter

The level of smoothness of a KDE is typically controlled by the smoothing parameter known as the bandwidth or *window width* (often denoted  $bw$ ,  $t$  or  $h$  in literature). For a histogram with constant bin width, the primary and secondary smoothing parameters are the bin width and bin origin. For a kernel estimator, the bandwidth serves as the smoothing parameter (Scott, 2012). We use the notation  $\Delta$  for the kernel bandwidth in this thesis.

The performance of the estimator  $\hat{f}$  for  $f$  depends crucially on the value of  $\Delta$  (Fortmann-Roe et al., 2012). The *optimal bandwidth* is the bandwidth for which the estimator satisfies some criterion of error or objective-specific criterion. An example is optimizing the estimator with respect to

the ISE, and thus the bandwidth which gives the lowest ISE is the optimal bandwidth of the problem. In general, what is defined as an optimal bandwidth depends on our objective, as the optimal bandwidth is not an obvious and unique definition. There exist a number of bandwidth algorithms or "bandwidth estimators" applicable for different problems, e.g., *Silverman's rule of thumb*<sup>7</sup> (Silverman, 1986), *Scott's rule of thumb* (Scott, 1992), and *Sheather-Jones bandwidth algorithm* (Sheather and Jones, 1991). See, e.g., discussions in Silverman (1986) and Scott (2012) on methods for selecting the optimal smoothing parameter, and Gramacki (2018) for a relatively up to date introduction to bandwidth selectors.

In Figure 2.4.6 we saw how the kernels are added up to form a complete estimator of a distribution of points. To illustrate the effects of bandwidth choice in kernel density estimation, we have a closer look at the creation of a Gaussian KDE. We let  $\{X_i\}$  be  $N$  independent realizations from an unknown distribution  $f(x)$ , then the estimator is explicitly defined (e.g., Botev et al. (2010)) as the sum

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^N \phi(x, X_i), \quad x \in \mathfrak{R}, \quad (2.4.14)$$

where  $\phi$  is the Gaussian kernel with standard deviation set by the bandwidth  $\Delta$ ;

$$\phi(x, X_i) = \frac{1}{\sqrt{2\pi}\Delta} \exp\left(\frac{-(x - X_i)^2}{2\Delta^2}\right). \quad (2.4.15)$$

Figure 2.4.8 shows how different bandwidth choices heavily affect the resulting estimate smoothness for a given number of samples. Obviously, for an insufficient number of sample points, the global error tends to be large regardless of bandwidth choices. The details of the distribution will drown in a broad  $\Delta$ , while a narrow  $\Delta$  will mainly describe the distribution of the points of one specific realization, and not the distribution from which they are sampled. Generally speaking, the more samples available, the narrower the bandwidth can be allowed to be. For a further discussion about bandwidth selection in KDE, see, e.g., Kile (2010) and references therein.

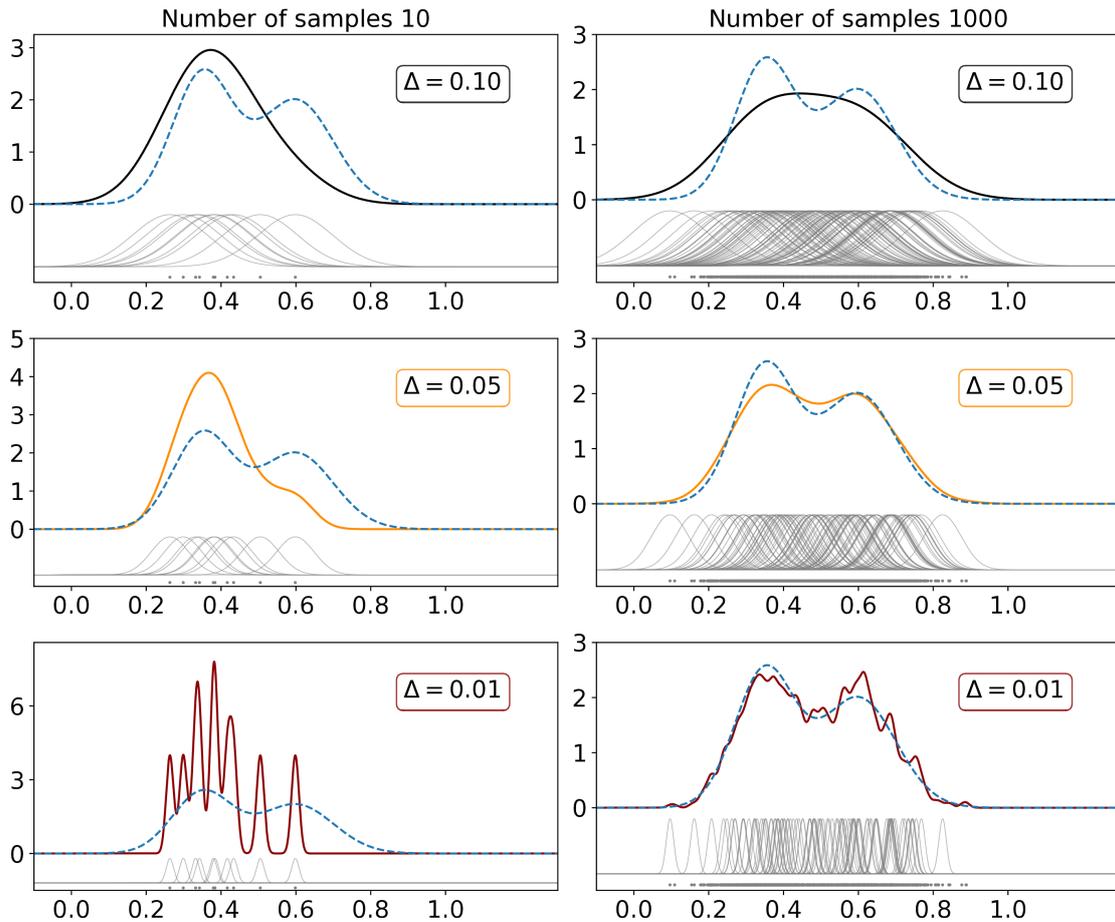
## 2.5 EQUIVALENCE OF EULERIAN AND LAGRANGIAN FORMULATIONS

We have been introduced to the advection-diffusion equation and seen how it can be solved using an Eulerian formulation in Section 2.1, or what we refer to as an *Eulerian fluid method*. In the rest of this chapter, we have looked at solving the advection-diffusion equation from a Lagrangian perspective, solving the problem using a *Lagrangian particle method*. We shall now look at the concept of equivalence between the two methods by comparing the two formulations of the same problem numerically, and investigating further how they differ. The advection-diffusion equation,

$$\frac{\partial}{\partial t} C + \frac{\partial}{\partial x} (uC) = \frac{\partial}{\partial x} \left( D \frac{\partial}{\partial x} C \right), \quad (2.5.1)$$

describes the physical phenomena where particles, energy, or other physical quantities are transferred inside a physical system due to both advection and diffusion. The Eulerian fluid method for solving the partial differential equation (PDE), Eq. (2.5.1), entails calculating the resulting probability distribution of the physical quantity in question as a function of time and space, by means of a finite difference scheme. In this subsection we shall use the Crank-Nicolson scheme as presented

<sup>7</sup>Numerical examples can be found at <https://kdepy.readthedocs.io/en/latest/bandwidth.html> (implementation of KDE in Python 3.5+ package).



**Figure 2.4.8:** Gaussian kernel density estimate (KDE) for samples (gray dots) from a bimodal distribution (dashed line) for different values of bandwidth  $\Delta$ . Sample points and scaled kernels are shown below the estimation (right column only displays every tenth kernel).

in Appendix A.3 for solving Eq. (2.5.1). Solving the same problem using the Lagrangian particle method essentially means solving the stochastic differential equation (SDE),

$$dX_t = (u + \partial_x D) dt + \sqrt{2D} dW_t, \quad (2.5.2)$$

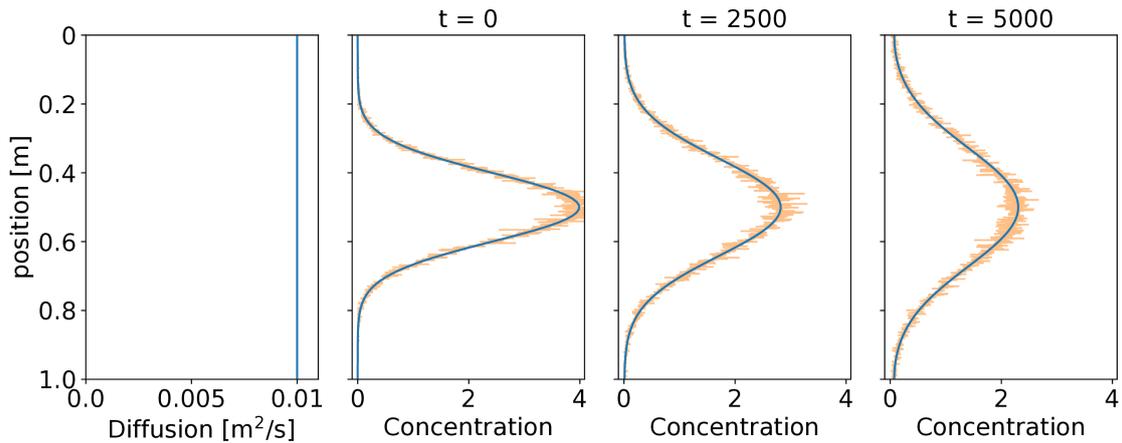
whose Fokker-Planck Equation is the advection-diffusion equation, as discussed in Section 2.2.4. The drift and diffusion coefficient were in this case found to be

$$a(x, t) = u(x, t) + \partial_x D(x, t) \quad \text{and} \quad b(x, t) = \sqrt{2D(x, t)}. \quad (2.5.3)$$

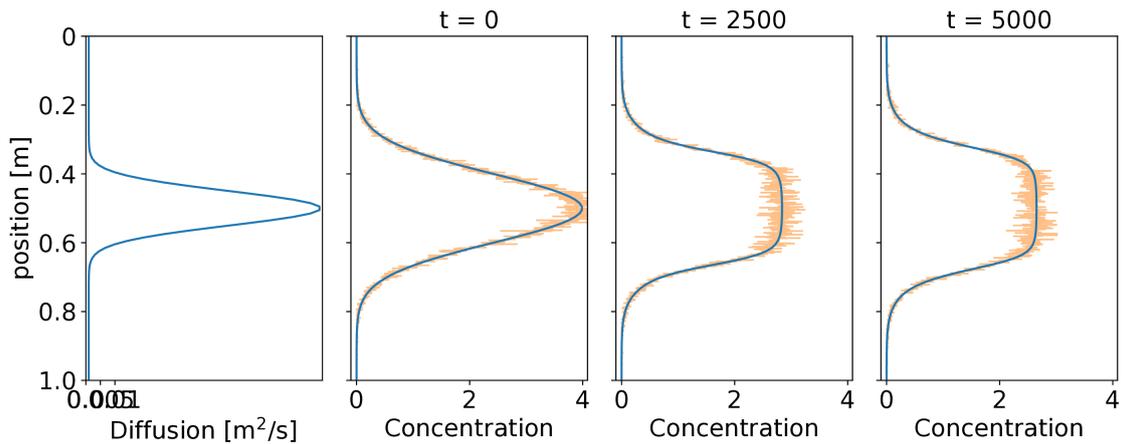
The solution of the SDE (2.5.2), is calculated by the means of Monte Carlo simulations by using the Euler-Maruyama scheme (2.3.5) for integration.

### Computational example: Constant diffusion in an initially Gaussian system

The comparison between solving the PDE and the SDE can be made for a purely diffusive system, and we will first show it for a constant diffusion  $\partial_x D = 0$ . For a system with initial distribution described by a Gaussian function, the two methods are applied, and the results are as expected.



**Figure 2.5.1:** From an initially Gaussian distributed concentration, the solution of the two separate formulations are presented at two instants  $t > 0$ . Blue is the *Eulerian* solution, yellow is the *Lagrangian* solution. The diffusion (scaled with  $D_0 = 0.01$ ) is a stationary and continuously differentiable function, a constant, shown in the left subplot. Periodic boundary conditions are applied. The Monte Carlo method obtains a solution using  $N = 100000$  simulations, and a time step of  $dt = 0.0002$ . The histogram estimates the solution with 1000 bins. The Eulerian method has a spatial discretization of  $dx = 0.01$ . Time  $t$  here is just time steps taken.



**Figure 2.5.2:** From an initially Gaussian distributed concentration, the solution of the two separate formulations are presented at two instants  $t > 0$ . Blue is the *Eulerian* solution, yellow is the *Lagrangian* solution. The diffusion (scaled with  $D_0 = 0.01$ ) is a stationary and continuously differentiable Gaussian function shown in the left subplot. Periodic boundary conditions are applied. The Monte Carlo method obtains a solution using  $N = 100000$  simulations, and a time step of  $dt = 0.0002$ . The histogram estimates the solution with 1000 bins. The Eulerian method has a spatial discretization of  $dx = 0.01$ . Time  $t$  here is just time steps taken.

The evolution of the probability distribution, shown in Figure 2.5.1 for both formulations, shows overlapping solutions indicating equivalence. The constant diffusion widens the probability density distribution, describing a spread of the tracer (or other physical quantity) as time passes. Would the system have evolved further, the concentration would eventually have ended up evenly distributed. The constant diffusion can be replaced by a variable diffusion: We can look at a system with diffusion described by a Gaussian function, as shown in the left-most subplot of Figure 2.5.2. The two methods produce overlapping results and appear to behave in the exact same manner yet again.

Both the examples presented have continuously differentiable diffusion functions  $D(x)$ , complying to the criterion (2.2.24) of smoothness on the coefficients  $a(\cdot)$  and  $b(\cdot)$ . However, in the next subsection, we will see that whenever the smoothness assumption fails, the two formulations do not produce the same result. The the SDE in Eq. (2.5.2) can no longer describe through the Fokker-Planck equation the same evolution of the probability distribution as the advection-diffusion equation (2.5.1).

### 2.5.1 "Moderately smooth functions"-criteria and when it fails

The importance of the assumption that the coefficients  $a(\cdot)$  and  $b(\cdot)$  are "moderately smooth functions" should not be taken lightly. It is a crucial assumption for the link to the advection-diffusion equation, but even so is often a forgotten or disregarded principle (as in, e.g., Chiri et al. (2020); De Dominicis et al. (2013)). The criteria (2.2.24) on the coefficients  $a(\cdot)$  and  $b(\cdot)$ , practically means that the diffusion,  $D$ , must be continuous and must have a continuous derivative, such that the second derivative exists. Furthermore, the velocity field must also be a continuous function, such that  $\partial_x a(x, t)$  exists.

#### Computational example: Comparing Eulerian and Lagrangian methods under the well-mixed condition

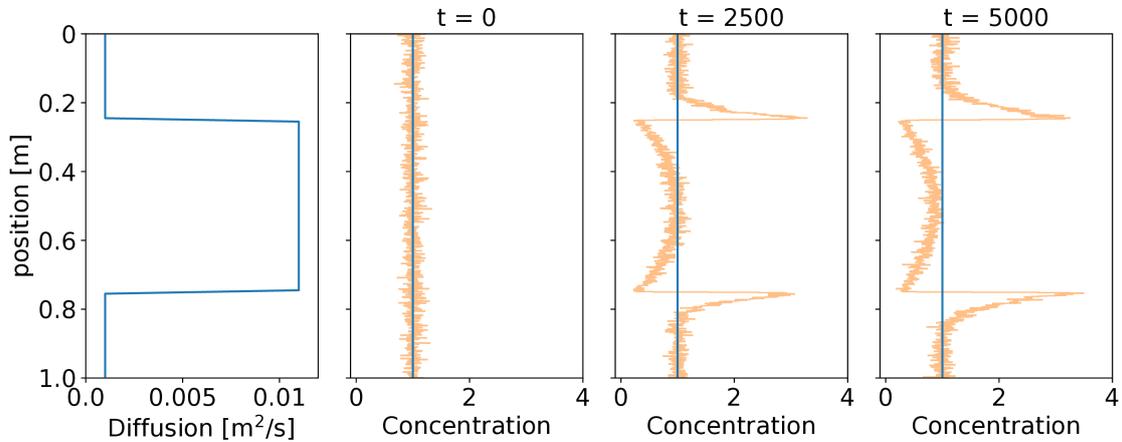
The importance of the fulfillment of the smoothness criterion can be shown by looking at a system for which the smoothness criterion is not fulfilled, and compare it to a slightly adjusted system that does fulfill the criterion. We shall look at a purely diffusive system, letting  $u = 0$ . As seen in, e.g., Chiri et al. (2020) and De Dominicis et al. (2013), discontinuous step functions are sometimes used to describe the diffusion in a system. The step function can easily be replaced by a smoother counterpart with a continuous derivative to show how this will change the computation results. As an example, we shall use a box function (see Figure 2.5.3), and a smoothed-out box function (see Figure 2.5.4), as the diffusion coefficient  $D(x)$ . The box function is chosen instead of the simple step function to keep the diffusion function continuous at the boundaries when periodic boundary conditions are applied.

We initiate the system in a well-mixed state (Thomson, 1987). This is a condition of the system that implies that for a diffusive system  $D(x) > 0 \forall x$ , an initially evenly distributed tracer will remain in the well-mixed state for all times. This can easily be seen from the diffusion equation

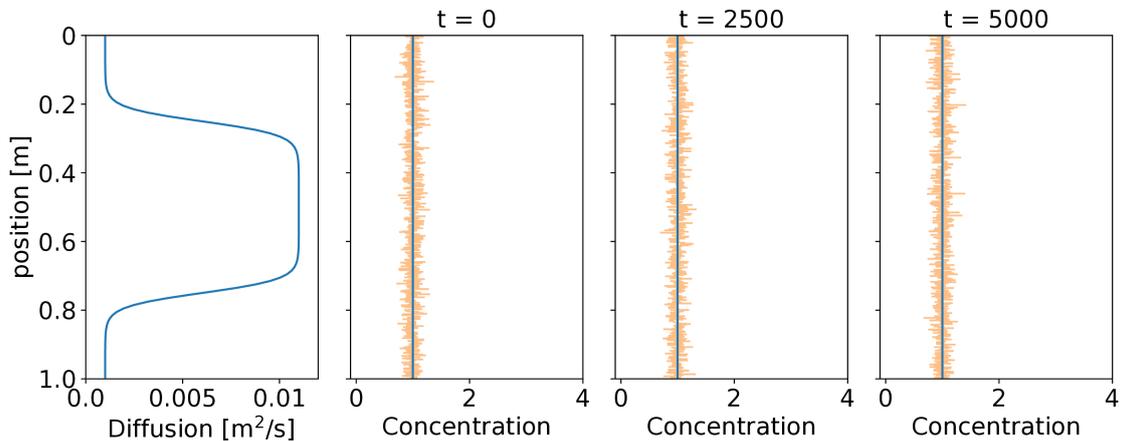
$$\frac{\partial C}{\partial t} = \frac{\partial D}{\partial x} \frac{\partial C}{\partial x} + D \frac{\partial^2 C}{\partial x^2}, \quad (2.5.4)$$

that by initially letting  $\partial_x C = 0$  everywhere, concentration does not change in time either,  $\partial_t C = 0$ . The condition is commonly used to test implementations of Lagrangian particle models, and applied in this case to see a clear difference when the formulations are not equivalent. Starting by looking at a square box diffusion function, i.e., a discontinuous function with no defined derivative. The coefficient  $a(x)$  is shown for zero velocity in Figure 2.5.3. The Eulerian and Lagrangian methods are applied to the problem, and solutions are shown after  $t$  time steps. As expected from the Eulerian solution, the system stays well-mixed throughout the integration time. However, the Lagrangian solution shows striking deviations at and around the points of discontinuity in  $a$ . In other words, the figure clearly describes the consequences of disregarding the smoothness criteria (2.2.24).

Slight changes in the diffusion function gives a smooth coefficient  $a(\cdot)$ , and we can demonstrate how the two formulations are equivalent under the same well-mixed condition as in Figure 2.5.3. The system in Figure 2.5.4 demonstrates how a smoothed out box function as diffusion coefficient, a continuously differentiable function, does not affect the well-mixed distribution, and by that demonstrates again the principle of equivalence between the two formulations. This concept is investigated further in two dimensions in Appendix B.1: Squared and smooth box diffusion in zero advection, where the exact same behavior is confirmed.



**Figure 2.5.3:** From an initially well-mixed concentration, the solution of the two separate formulations are presented at two instants  $t > 0$ . Blue is the *Eulerian* solution, yellow is the *Lagrangian* solution. The diffusion (scaled with  $D_0 = 0.01$ ) is a stationary and discontinuous function shown in the left subplot. Periodic boundary conditions are applied. The Monte Carlo method obtains a solution using  $N = 100000$  simulations, and a time step of  $dt = 0.0001$ . The histogram estimates the solution with 1000 bins. The fluid method has a spatial discretization of  $dx = 0.01$ . Time  $t$  here is just time steps taken.



**Figure 2.5.4:** From an initially well-mixed concentration, the solution of the two separate formulations are presented at two instants  $t > 0$ . Blue is the *Eulerian* solution, yellow is the *Lagrangian* solution. The diffusion (scaled with  $D_0 = 0.01$ ) is a stationary and continuously differentiable function ( $\partial_x D$  is continuous). Periodic boundary conditions are applied. The Monte Carlo method obtains a solution using  $N = 100000$  simulations, and a time step of  $dt = 0.0001$ . The histogram estimates the solution with 1000 bins. The fluid method has a spatial discretization of  $dx = 0.01$ . Time  $t$  here is just time steps taken.

## Chapter 3

# CONCENTRATION FIELD IN DIFFUSIVE DOUBLE GYRE FLOW

In the Eulerian formulation we solve the advection-diffusion equation which describes the time evolution of a probability density function (of some physical quantity). We refer to this method as the Eulerian fluid method. The solution can, e.g., describe the concentration field of some specimen or tracer present in a fluid. The same partial differential equation can also be solved using what we call a Lagrangian particle method. Instead of solving the equation directly, we solve a stochastic differential equation for a number of Lagrangian particles – i.e., simulate independent realizations of the stochastic process that is the solution of the SDE. The simulation results can be considered samples from the probability density distribution itself. Through density estimation we can produce matching results to the fluid method, as these methods are equivalent under certain criteria (see Section 2.2.4).

This chapter presents implementation and results using both formulations of the advection-diffusion problem. The Eulerian fluid method and the Lagrangian particle method are used to solve the advection-diffusion equation in the double gyre flow presented in Chapter 1, in a system with constant diffusion. All implementations are written in **Python**, and when appropriate, we list explicit implementations or imports of libraries.

### Double gyre system

The double gyre system presented in Chapter 1 is straightforwardly implemented as a velocity field in two dimensions. For the discretized domain in the fluid method, the implementation of numerically calculated derivatives by a central finite difference method is found in Section 3.1.2. The particle method uses a forward finite difference method (Section 3.3.1), and the method is not confined to the grid structure. The parameters of the double gyre flow field are

$$A = 0.1, \quad \epsilon = 0.1, \quad \omega = \frac{2\pi}{10}. \quad (3.0.1)$$

The field is defined on the domain  $x \in [0, 2]$  and  $y \in [0, 1]$ . For the sake of simplicity we use this domain in all numerical implementations (such as in Appendix B.1).

**Table 3.0.1:** Implemented time step sizes  $\Delta t$ , and the number of time steps  $Nt$ , for the maximum integration time  $T = 10$ .

$\Delta t$	0.1	0.05	0.025	0.0125	0.00625
$Nt$	100	200	400	800	1600
$\Delta t$	0.003125	0.001562	0.00078125	0.00039063	0.00019531
$Nt$	3200	6400	12800	25600	51200

### Setup for numerical simulations

The two methods for solving the advection-diffusion equation are run separately. For them to solve the same problem, the two system descriptions must be equivalent. In Section 3.1 and 3.3 we describe in detail the implementations of each method, but their common setup is introduced in the following.

*Discretization.* The fluid method will use a discretized spatial grid, while the particle method is continuous in space. Both methods are discrete in time, and we must set the time step  $\Delta t$  and also the length of the simulation as the number of time steps  $Nt$ . To compare simulations at the same point in time, we have chosen a set of available time steps through halving an initial large discretization. The largest time step is set to  $\Delta t = 0.1$ , and thus the range of available time steps is given as  $\{\Delta t\} = 0.1 \cdot \{(\frac{1}{2})^0, (\frac{1}{2})^1, (\frac{1}{2})^2, (\frac{1}{2})^3, \dots, (\frac{1}{2})^9\}$ . The maximum integration time is set to  $T = 10$ , and thus the number of time steps is given by  $Nt = T/\Delta t$ . The temporal discretizations available are summed up in Table 3.0.1.

*Initial conditions.* The initial condition for the computations in this chapter is set to be a two-dimensional Gaussian function, centered in the middle of the domain,

$$G(x, y) = \exp\left(-\frac{(x - \mu_x)^2}{2\sigma_x^2} - \frac{(y - \mu_y)^2}{2\sigma_y^2}\right), \quad (3.0.2)$$

for parameter values given in Table 3.0.2. In the fluid method, the initial probability distribution is normalized,

$$C(x, y, 0) = \frac{G(x, y)}{\iint G(x, y) dx dy}, \quad (3.0.3)$$

where the Gaussian function is numerically integrated over the grid. Matching the initial conditions of the fluid method, the initial position in each simulation ( $Np$  simulations at the time) in the particle method is drawn from the multivariate normal distribution<sup>1</sup>

---


$$\mathbf{x}_0, \mathbf{y}_0 = \text{np.random.multivariate\_normal}(\text{mean}, \text{cov}, \text{Np}).T$$


---

with mean and variance described by

$$\text{mean} = \begin{bmatrix} \mu_x & \mu_y \end{bmatrix}, \quad \text{and} \quad \text{cov} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \quad (3.0.4)$$

<sup>1</sup>The initial conditions can be replicated for  $Np$  particles by using the same seed for the **numpy** random number generator. This has been done when comparing the Euler-Maruyama scheme for different time step sizes (see Section 3.4.1), looking at strong convergence.

for values given in Table 3.0.2.

*Boundary conditions.* The spatial boundary condition is set to be periodic. For a Lagrangian particle, this means that any integration that returns a position outside the domain will find the periodic equivalence of the position at the opposite side. For the fluid method, the periodicity means that for the boundary grid points, their neighbours are found at the other side of the domain. The specific implementations of the boundary condition in the two methods are given in Section 3.1.2 for the fluid method and Section 3.3.1 for the particle method.

**Table 3.0.2:** Parameter values for initial probability density distribution in both formulations.

$\mu_x$	$\mu_y$	$\sigma_x$	$\sigma_y$
1.0	0.5	0.1	0.1

## 3.1 FLUID METHOD: NUMERICAL IMPLEMENTATION

To solve the partial differential equation called the advection-diffusion equation, we have discretized both the spatial and temporal domain. We have used the finite difference method Crank-Nicolson to approximate derivatives, and applied periodic boundary conditions. The system is solved using numerical linear algebra. This section shows all implementations in as much detail as needed to replicate, without necessarily providing the actual code. It follows a natural outline from discretization, implementing the chosen scheme and solving the resulting linear system. The section ends with a discussion about the constraints in the method. These assumptions, limits and concerns are taken into consideration when setting up of the numerical simulations. Results from computations will be presented in the next section, but first, a thorough presentation of the implementation.

### 3.1.1 Domain discretization

The spatial domain of interest is divided into grid cells with sides of lengths  $\Delta x$  and  $\Delta y$ . The non-dimensional domain is the rectangle  $x \in [0, 2]$  and  $y \in [0, 1]$ . The domain is discretized as

$$x = x_0 + i\Delta x, \quad \text{and} \quad y = y_0 + i\Delta y,$$

for  $x_0 = \frac{\Delta x}{2}$  and  $y_0 = \frac{\Delta y}{2}$ . The grid resulting from this discretization is presented in Figure 3.1.1 for a small example system. The numerical grid is implemented using `numpy.meshgrid`. Specifically for the one-dimensional arrays  $\mathbf{x}$  of length  $Nx$ , and  $\mathbf{y}$  of length  $Ny$ , the two-dimensional grid is produced as follows

---

```
xx, yy = np.meshgrid(x, y)
```

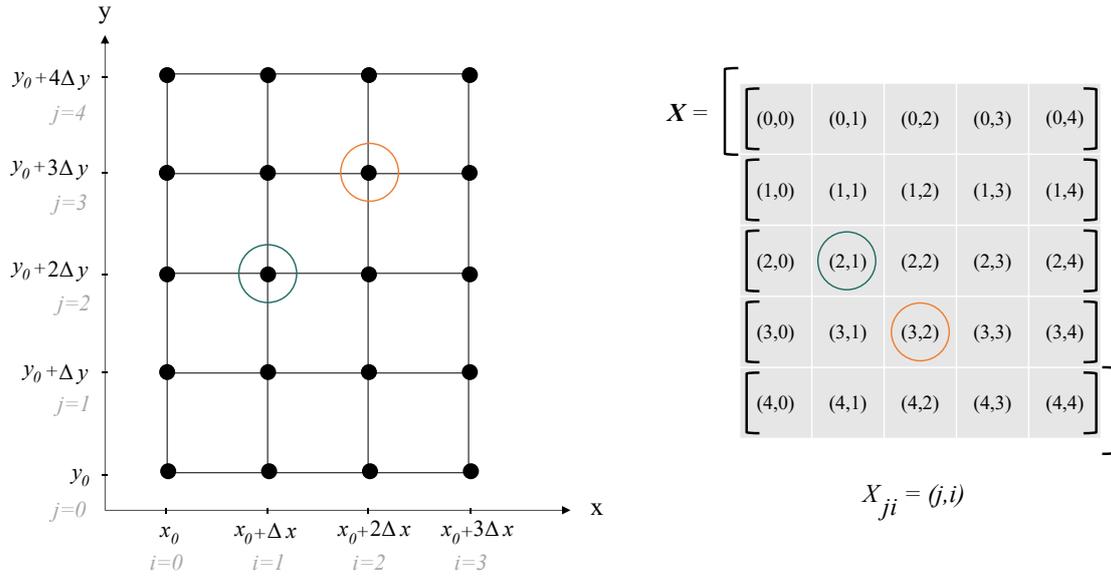
---

By this manner, the grid arrays  $\mathbf{xx}$  and  $\mathbf{yy}$  are of the shape  $(Ny, Nx)$ . The grid points are accessed row-wise as in the example matrix  $\mathbb{X}$ , presented as visual pseudocode to the right in Figure 3.1.1. All two-dimensional quantities are handled in a row-wise manner. All spatially dependent fields have the same structure as the matrix  $\mathbb{X}$ : The concentration  $\mathcal{C}_{j,i}$ , the velocities  $\mathcal{U}_{j,i}$  and  $\mathcal{V}_{j,i}$ , and the diffusion field  $\mathcal{D}_{j,i}$ .

The temporal discretization is given by the index  $n$ , and expressed

$$t = t_0 + n\Delta t,$$

where we only use a constant time steps  $\Delta t$ , and let  $t_0 = 0$ .



**Figure 3.1.1:** Two-dimensional grid shown for a small example domain. An example two-dimensional field matrix  $\mathbb{X}$  is shown to the right. Two points are explicitly marked in both the grid and the matrix. The matrix is presented with row zero at the top, and the last row at the bottom.

### Row-ordering

Even though the two-dimensional numerical fields - concentration  $\mathcal{C}$ , velocity components  $\mathcal{U}$  and  $\mathcal{V}$ , and diffusion  $\mathcal{D}$  - are defined as matrices, we shall more frequently use them in a column vector form. They are flattened out row-wise to adapt to the one-dimensional shape. E.g., the concentration matrix  $\mathcal{C}_{j,i}$  is flattened to become

$$\mathbf{C} = \begin{bmatrix} \mathcal{C}_{0,0} \\ \mathcal{C}_{0,1} \\ \mathcal{C}_{0,2} \\ \vdots \\ \mathcal{C}_{0,Nx-2} \\ \mathcal{C}_{0,Nx-1} \\ \mathcal{C}_{1,0} \\ \mathcal{C}_{1,1} \\ \vdots \\ \mathcal{C}_{j,i} \\ \mathcal{C}_{j,i+1} \\ \mathcal{C}_{j,i+2} \\ \vdots \\ \mathcal{C}_{Ny-1,Nx-2} \\ \mathcal{C}_{Ny-1,Nx-1} \end{bmatrix} \tag{3.1.1}$$

This is the format referred to as a *row-ordered column vector* in the following, or simply column vector.

### 3.1.2 Crank-Nicolson scheme in two dimensions

We used the Crank-Nicolson scheme to approximate the derivatives in the advection-diffusion equation, resulting in equations solved through numerical linear algebra. We will describe the derivation of the scheme for the specific equation and the specific discretization chosen. Periodic boundary conditions are applied and explained in detail.

#### Deriving the scheme explicitly for the two-dimensional advection-diffusion equation

The Crank-Nicolson scheme for the advection-diffusion equation in two-dimensions is found by first expressing the equation

$$\frac{\partial C}{\partial t} = \nabla \cdot (D\nabla C) - \nabla \cdot (\mathbf{u}C), \quad (3.1.2)$$

for two-dimensional operator  $\nabla = \partial_x \hat{\mathbf{x}} + \partial_y \hat{\mathbf{y}}$ . It is further assumed that

$$D = D(x, y) \quad C = C(x, y, t) \quad \mathbf{u} = \mathbf{u}(x, y, t) = u\hat{\mathbf{x}} + v\hat{\mathbf{y}}$$

By writing it out,

$$\frac{\partial C}{\partial t} = \frac{\partial D}{\partial x} \frac{\partial C}{\partial x} + D \frac{\partial^2 C}{\partial x^2} + \frac{\partial D}{\partial y} \frac{\partial C}{\partial y} + D \frac{\partial^2 C}{\partial y^2} - \frac{\partial u}{\partial x} C - u \frac{\partial C}{\partial x} - \frac{\partial v}{\partial y} C - v \frac{\partial C}{\partial y}, \quad (3.1.3)$$

the common order of derivatives can be collected, as

$$\frac{\partial C}{\partial t} = \left( \frac{\partial D}{\partial x} - u \right) \frac{\partial C}{\partial x} + \left( \frac{\partial D}{\partial y} - v \right) \frac{\partial C}{\partial y} + D \frac{\partial^2 C}{\partial x^2} + D \frac{\partial^2 C}{\partial y^2} + \left( -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) C. \quad (3.1.4)$$

The scalar field is discretized as

$$C(x, y, t) = C(x_0 + i\Delta x, y_0 + j\Delta y, t_0 + n\Delta t) = C(x_i, y_j, t_n).$$

To keep the notation from Section 3.1.1 defining the discretization, we shall describe the two dimensional matrix at a time  $t_n$  by

$$\mathcal{C}_{j,i}^n = C(x_i, y_j, t_n).$$

(It should be noted that in most implementations the first dimension in the matrix is also the first dimension in space ( $x$ ). Hence, the presentation above is used to not confuse the reader and stick with the structure of the grid implemented.) The approximations to the derivatives are expressed through the Crank-Nicolson scheme (Section A.3). They are listed below, expressing the derivatives evaluated at the point  $(x_i, y_j, t_n)$ .

$$\frac{\partial C}{\partial t} = \frac{\mathcal{C}_{j,i}^{n+1} - \mathcal{C}_{j,i}^n}{\Delta t} \quad (3.1.5)$$

$$\frac{\partial C}{\partial x} = \frac{1}{2} \left[ \frac{\mathcal{C}_{j,i+1}^{n+1} - \mathcal{C}_{j,i-1}^{n+1}}{2\Delta x} + \frac{\mathcal{C}_{j,i+1}^n - \mathcal{C}_{j,i-1}^n}{2\Delta x} \right] \quad (3.1.6)$$

$$\frac{\partial C}{\partial y} = \frac{1}{2} \left[ \frac{C_{j+1,i}^{n+1} - C_{j-1,i}^{n+1}}{2\Delta x} + \frac{C_{j+1,i}^n - C_{j-1,i}^n}{2\Delta x} \right] \quad (3.1.7)$$

$$\frac{\partial^2 C}{\partial x^2} = \frac{1}{2} \left[ \frac{C_{j,i+1}^{n+1} - 2C_{j,i}^{n+1} + C_{j,i-1}^{n+1}}{\Delta x^2} + \frac{C_{j,i+1}^n - 2C_{j,i}^n + C_{j,i-1}^n}{\Delta x^2} \right] \quad (3.1.8)$$

$$\frac{\partial^2 C}{\partial y^2} = \frac{1}{2} \left[ \frac{C_{j+1,i}^{n+1} - 2C_{j,i}^{n+1} + C_{j-1,i}^{n+1}}{\Delta y^2} + \frac{C_{j+1,i}^n - 2C_{j,i}^n + C_{j-1,i}^n}{\Delta y^2} \right] \quad (3.1.9)$$

The scheme is central in space, and hence result in a spatial five-point stencil, as seen in Figure A.6 of Appendix A.3, considering neighboring points in both directions. The derivation is shown in full in Appendix C.2, and here we present only the resulting scheme for convenience. The solution can be written, for the left hand side matrix  $\mathbb{L}$  and right hand side matrix  $\mathbb{R}$ , and the row-ordered column vector form of  $\mathbf{C}$  (see Section 3.1.1):

$$\mathbb{L} \cdot \mathbf{C}^{n+1} = \mathbb{R} \cdot \mathbf{C}^n. \quad (3.1.10)$$

The matrices of the scheme are square  $(Nx \cdot Ny) \times (Nx \cdot Ny)$ , and the concentration vector  $\mathbf{C}$  is of shape  $(Nx \cdot Ny)$ . Let the matrices  $\mathbb{L}$  and  $\mathbb{R}$  be indexed through their row number  $r$  and column number  $c$ , as  $[r, c]$ , then the matrices are expressed explicitly through

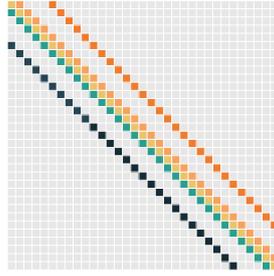
$$\mathbb{L} : \begin{array}{ll} [r, r] & \longrightarrow 1 + D\lambda^x + D\lambda^y - (-\partial_x u - \partial_y v) \Delta t / 2 \\ [r, r \pm 1] & \longrightarrow \mp (\partial_x D - u) \alpha^x - D\lambda^x / 2 \\ [r, r \pm Nx] & \longrightarrow \mp (\partial_y D - v) \alpha^y - D\lambda^y / 2 \end{array} \quad (3.1.11)$$

$$\mathbb{R} : \begin{array}{ll} [r, r] & \longrightarrow 1 - D\lambda^x - D\lambda^y + (-\partial_x u - \partial_y v) \Delta t / 2 \\ [r, r \pm 1] & \longrightarrow \pm (\partial_x D - u) \alpha^x + D\lambda^x / 2 \\ [r, r \pm Nx] & \longrightarrow \pm (\partial_y D - v) \alpha^y + D\lambda^y / 2 \end{array} \quad (3.1.12)$$

where we have simplified, using the following variables

$$\lambda^x = \frac{\Delta t}{(\Delta x)^2} \quad \lambda^y = \frac{\Delta t}{(\Delta y)^2} \quad \alpha^x = \frac{\Delta t}{4\Delta x} \quad \alpha^y = \frac{\Delta t}{4\Delta y}.$$

Before applying any boundary conditions, the matrices of the spatial five-stencil Crank-Nicolson scheme are five-diagonal sparse matrices, as illustrated in Figure 3.1.2. The offsets at  $\pm 1$  and  $\pm Nx$ , have opposing signs as found in Eq. (3.1.11) and (3.1.12), and the main diagonals are connected through  $\mathbb{L}_{r,r} = -\mathbb{R}_{r,r} + 2$ . An additional short note about the calculation of the matrices is found in Appendix C.2, explaining in detail the construction of the matrices as functions of the fields  $D(x, y)$  and  $\mathbf{u}(x, y, t)$ , and their derivatives.



**Figure 3.1.2:** The five-point stencil of the Crank-Nicolson scheme result in five-diagonal matrices  $\mathbb{R}$  and  $\mathbb{L}$ . Additionally to the main diagonal, the offsets are  $\pm 1$  and  $\pm Nx$ .

### Periodic boundary conditions

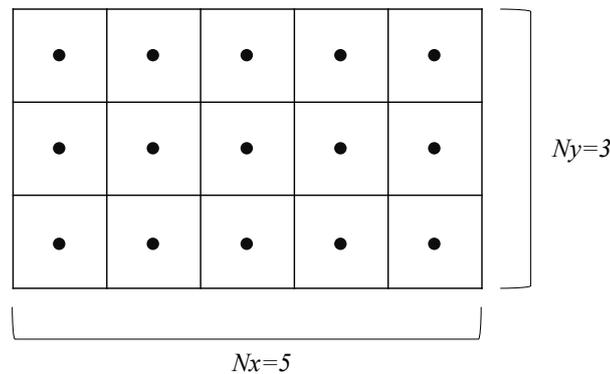
Following the spatial five-point stencil of the Crank-Nicolson scheme, periodic boundary conditions are applied in two dimensions. The implementation is described in detail by applying it to a smaller example system. The example system is presented as a  $(3 \times 5)$  grid ( $Ny = 3$ ,  $Nx = 5$ ), as pictured in Figure 3.1.3. The grid points are named by number as in Figure 3.1.4, in the order they appear in the concentration vector  $\mathbf{C}$ , in other words natural row order. Figure 3.1.4 shows in detail how the rows of the matrix are found, based on where we find the neighbouring points in the 5-point stencil. This approach results in the final matrix represented by Figure 3.1.5.

### Estimating derivatives

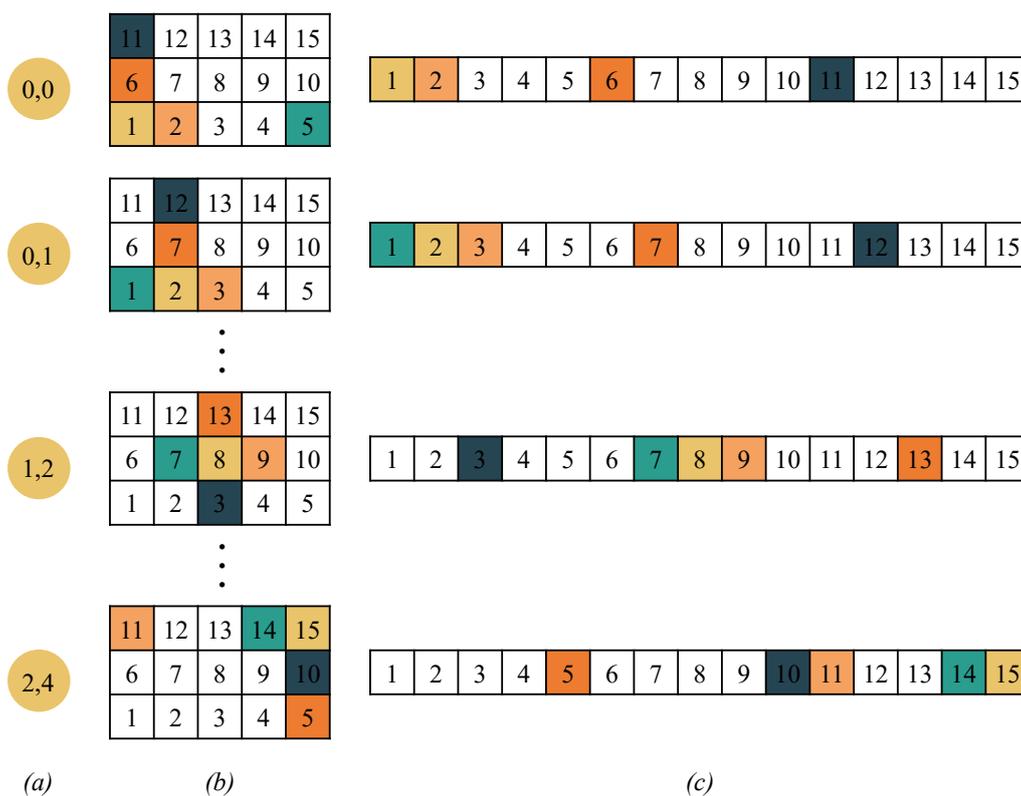
The Crank-Nicolson scheme matrices contain terms including the differentiated velocities in both dimensions,  $\partial_x u$  and  $\partial_y v$ . The derivatives are calculated using the central finite difference method (see Section A.2), solving

$$\partial_x \mathbf{U} = \mathbb{M}_{\hat{x}} \cdot \mathbf{U} \quad \text{and} \quad \partial_y \mathbf{V} = \mathbb{M}_{\hat{y}} \cdot \mathbf{V}, \quad (3.1.13)$$

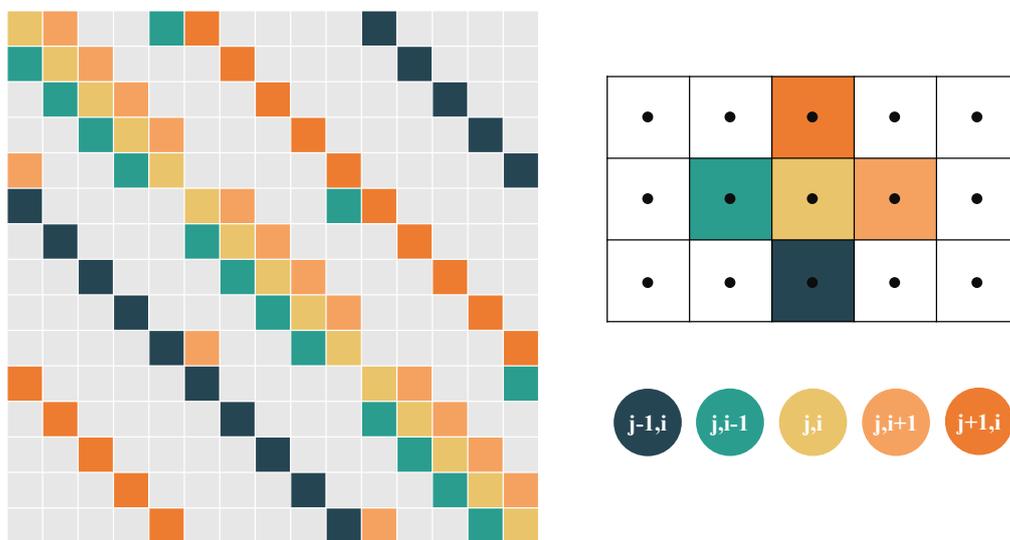
where  $\mathbf{U}$  and  $\mathbf{V}$  are row-ordered column vectors of the two dimensional velocity fields  $\mathcal{U}_{j,i}$  and  $\mathcal{V}_{j,i}$ .  $\mathbb{M}_{\hat{k}}$  is the central difference matrix in dimension  $\hat{k}$ . The central difference matrices are implemented with periodic boundaries.



**Figure 3.1.3:** Example system grid with centered grid points.



**Figure 3.1.4:** Visual description of boundary conditions, where the colors represent the different stencil points described in 3.1.5. The figure shows (a) the grid index  $(j, i)$ , (b) shows the stencil on the grid according to the periodic boundary condition, and (c) the location of the same points in the concentration vector  $\mathbf{C}$ .



**Figure 3.1.5:** Matrix structure used in Crank-Nicolson scheme, with periodic boundary conditions. Shown for an example grid. To the right, the grid with 5-point stencil shown, where colored cells correspond to grid points  $(i, j)$ .

### Iteratively solving linear system

The matrix equation (3.1.10) at a time  $t_n$ ,

$$\mathbb{L} \cdot \mathbf{C}^{n+1} = \mathbb{R} \cdot \mathbf{C}^n, \quad (3.1.14)$$

is solved by performing two consecutive operations. First an inner product to calculate the right hand side  $\mathbf{b} = \mathbb{R} \cdot \mathbf{C}^n$ . Followed by an iterative method for solving the linear equation  $\mathbb{L} \cdot \mathbf{C}^{n+1} = \mathbf{b}$ . The chosen iterative method, the biconjugate gradient stabilized method, is imported and implemented as follows:

---

```
from scipy.sparse.linalg import bicgstab

# Solve equation  $Lc = b$  at time  $n+1$ 
b      = R.dot(c)
c, conv = bicgstab(L, b, tol = 1e-12)
```

---

The program exits whenever the tolerance is not met. The method is selected because the matrix  $\mathbb{L}$  is not symmetric, and to avoid the computationally expensive operation of inverting the matrix.

### 3.1.3 Constraints of numerical model

As presented in Appendix A.3, the Crank-Nicolson scheme is unconditionally stable. The scheme has different behavior when solving different systems; thus the constraints presented in this subsection are constraints for solving an advection-diffusion problem specifically. Even though the scheme considered unconditionally stable, there are still some measures that should be taken to ensure a correct solution.

#### Advection- and diffusion-dominated systems

The choice of implementing a central finite difference scheme is well reasoned with the wish of not obtaining oscillating solutions. However, for non-smooth solutions, this is not guaranteed. For the advection-diffusion problem, the balance between length scale and the value of transport-terms is described through the Péclet number (for mass) as

$$Pe = \frac{\text{advective transport rate}}{\text{diffusive transport rate}} = \frac{Lu}{D}, \quad (3.1.15)$$

for a diffusion coefficient  $D$  and velocity  $u$ . A system is said to be advection-dominated if  $Pe \gg 1$ , and diffusion-dominated if reversely  $Pe \ll 1$ . To avoid completely the oscillating behavior, the grid size  $\Delta x$  and  $\Delta y$  must be small enough. For the central difference scheme, the restriction on the numerical approximations is set by the *cell Péclet number*

$$Pe_{cell,x} = \frac{\Delta x u}{D} \quad \text{and} \quad Pe_{cell,y} = \frac{\Delta y v}{D}. \quad (3.1.16)$$

The cell Péclet number is restricted to

$$Pe_{cell} \leq 2. \quad (3.1.17)$$

to avoid oscillations and negative values in the solution according to Hundsdorfer (2003, p. 67).

### Restrictions on diffusion relative grid size

The constraints presented above suggest that we can not simulate a system with a diffusion coefficient that is "too low," as this will result in the restriction of a relatively fine grid. To avoid unnecessarily long and computationally costly computations, we have not worked with systems larger than the grid size of  $(400 \times 800)$ , which is a grid cell size of  $\Delta x = \Delta y = 0.0025$ . Grids are from here on out referred to by their matrix shape  $(Ny \times Nx)$ .

Our efforts have been focused on a slightly coarser grid through the calculations. We have concentrated our efforts on the  $(200 \times 400)$  grid, which has worked well in most simulations (all of Chapter 4 is presented in this grid resolution). To stay within the restrictions the cell Péclet number sets on our calculations, we stay clear of the oscillating solution by calculating for appropriate diffusion coefficients. A list of parameters for four different constant diffusion coefficients  $D_0$  are presented in Table 3.1.1.

**Table 3.1.1:** Parameter dependency shown for four different values of  $D_0$ . Dark gray cells have cell Péclet numbers above the limit (3.1.17). Time steps  $\Delta t$  are calculated using Eq. (3.1.19).

$D_0$	$\Delta x = \Delta y$	$(Ny \times Nx)$	$Pe_{cell}$	$\Delta t$
0.01	0.02	$(50 \times 100)$	0.6277	0.04
	0.01	$(100 \times 200)$	0.3141	0.01
	0.005	$(200 \times 400)$	0.1571	0.0025
	0.0025	$(400 \times 800)$	0.0785	0.000625
0.005	0.02	$(50 \times 100)$	1.2554	0.08
	0.01	$(100 \times 200)$	0.6281	0.02
	0.005	$(200 \times 400)$	0.3142	0.005
	0.0025	$(400 \times 800)$	0.1571	0.00125
0.001	0.02	$(50 \times 100)$	6.2770	0.4
	0.01	$(100 \times 200)$	3.1408	0.1
	0.006667	$(150 \times 300)$	2.0942	0.044444
	0.005	$(200 \times 400)$	1.5707	0.025
	0.004	$(250 \times 500)$	1.2566	0.016
	0.003333	$(300 \times 600)$	1.0472	0.011111
	0.002857	$(350 \times 700)$	0.8976	0.008163
	0.0025	$(400 \times 800)$	0.7854	0.00625
0.0005	0.02	$(50 \times 100)$	12.5540	0.8
	0.01	$(100 \times 200)$	6.2816	0.2
	0.005	$(200 \times 400)$	3.1414	0.05
	0.0025	$(400 \times 800)$	1.5708	0.0125

### Relation between spatial and temporal resolution

As the cell Péclet number depends on both spatial resolution and the diffusivity, we must be careful when we change the resolution. Even though Crank-Nicolson is an unconditionally stable scheme, we must ensure that the solution to which it converges is the correct one. To ensure stability in *explicit* schemes when solving advection- and diffusion-dominated problems, the time step is often restricted to respectively  $\frac{|u|\Delta t}{\Delta x} \leq 1$  and  $\frac{D\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$ . This results in the limits

$$\Delta t \leq \frac{\Delta x}{|u|} \quad \text{and} \quad \Delta t \leq \frac{(\Delta x)^2}{2D}. \quad (3.1.18)$$

As we have chosen the Crank-Nicolson scheme for unconditional stability, we assume that these constraints are relatively harsh and have decided to opt for the less strict constraint

$$\Delta t \leq \frac{(\Delta x)^2}{D}. \quad (3.1.19)$$

Solutions that follow the above limit of time step size, along with a maximum cell Péclet number  $Pe_{cell} \leq 2$ , should give non-oscillating solutions.

## 3.2 FLUID METHOD: NUMERICAL RESULTS

In this section, we show the results from computing the probability density function as it evolves with time. Numerical implementations are fully described in the previous section, and only relevant parameters and constants are given here. The advection-diffusion equation is solved using the finite difference scheme Crank-Nicolson, and the iterative solver `bicgstab` (biconjugate gradient stabilized method) from the `scipy.sparse.linalg` library. The flow velocity field is the double gyre flow, and the diffusion is set to be constant over the domain,  $D(x, t) = D_0$ . The system is solved for a range of different discretizations and diffusion constants. From an initial Gaussian distribution, we first look at the convergence of the Crank-Nicolson scheme with respect to the spatial and temporal discretization. A large part of the results includes investigating the conservation of mass in the system, and seeing how poor choices of diffusion and discretization can result in oscillating and negative valued solutions.

### 3.2.1 Convergence of Crank-Nicolson numerical scheme

#### Convergence in time

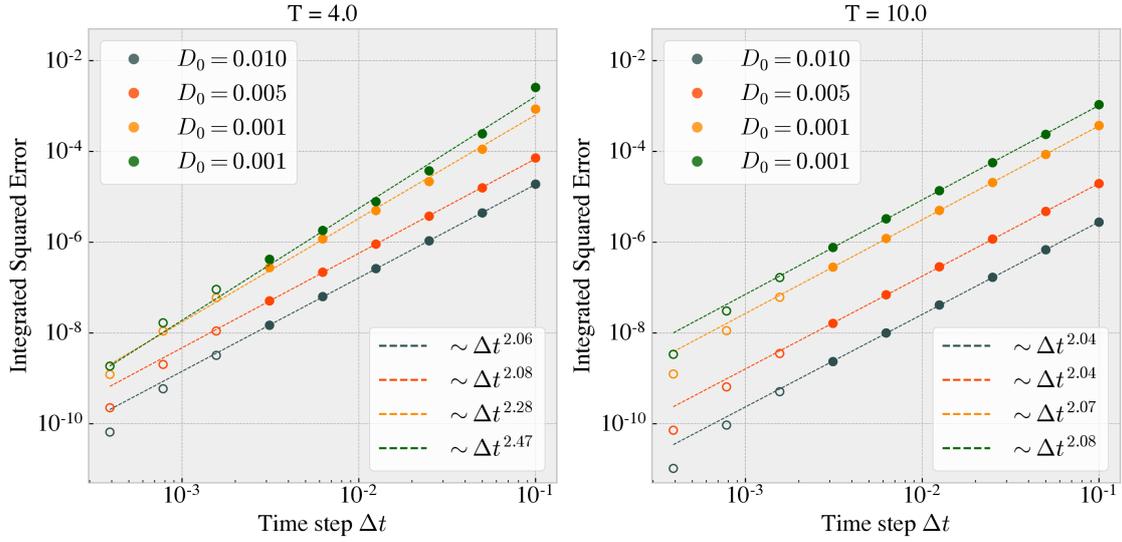
Figure 3.2.1 shows that the Crank-Nicolson scheme has an approximated second-order convergence in time, as expected. The integrated squared error is computed relative to a solution using the smallest time step available ( $\Delta t = 0.1 \cdot (1/2)^9$ ), and all solutions have been computed on a  $(200 \times 400)$ -grid. It shows that even for time steps much larger than the limit  $\Delta t \leq \frac{(\Delta x)^2}{D_0}$ , would allow, see Table 3.1.1, the solutions converge following a second order trend.

#### Convergence in space

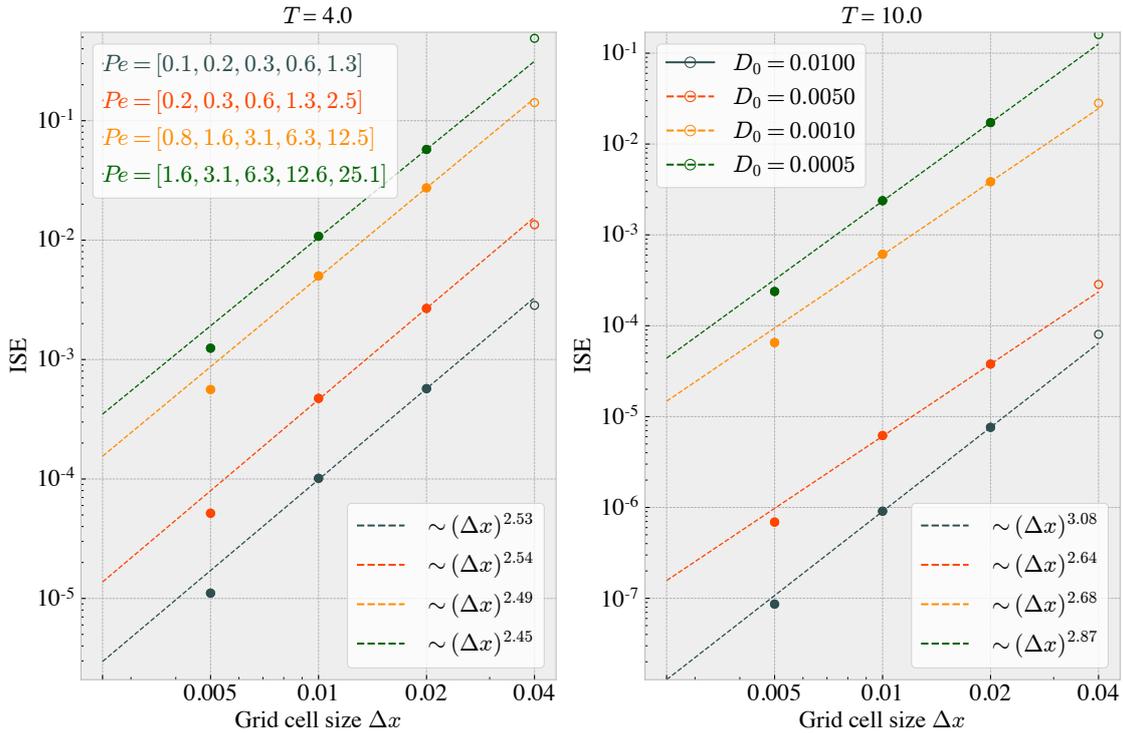
The convergence rate of the Crank-Nicolson scheme with regards to the grid discretization was investigated. We see that even for systems with  $Pe_{cell} = Pe_{cell}(D_0, \Delta x) > 2$ , the convergence trend follows a constant slope in the log-log plot in Figure 3.2.2. The figure shows the convergence of the scheme with grid discretization  $\Delta x$ . The time step is constant for all grids,  $\Delta t = 0.1 \cdot (1/2)^9$ , and satisfy  $\Delta t \leq (\Delta x)^2/D_0$  in all discretizations. The regression is computed using `curve_fit` from `scipy.optimize`.

### 3.2.2 Conservation of mass

Since diffusive and advective processes do not create or destroy mass, they only redistribute mass; the total mass should remain conserved in the system. As we have seen with Figures 3.2.1 and 3.2.2, the solution should converge to the true solution using a small enough time step  $\Delta t$  and grid spacing  $\Delta x$ . However, what also needs to be accounted for is the Péclet number, as mentioned in Section 3.1.3. Thus, we look at the conservation of mass as a function of time for different diffusion coefficients and different spatial discretizations in Figure 3.2.3. The time steps are regulated so



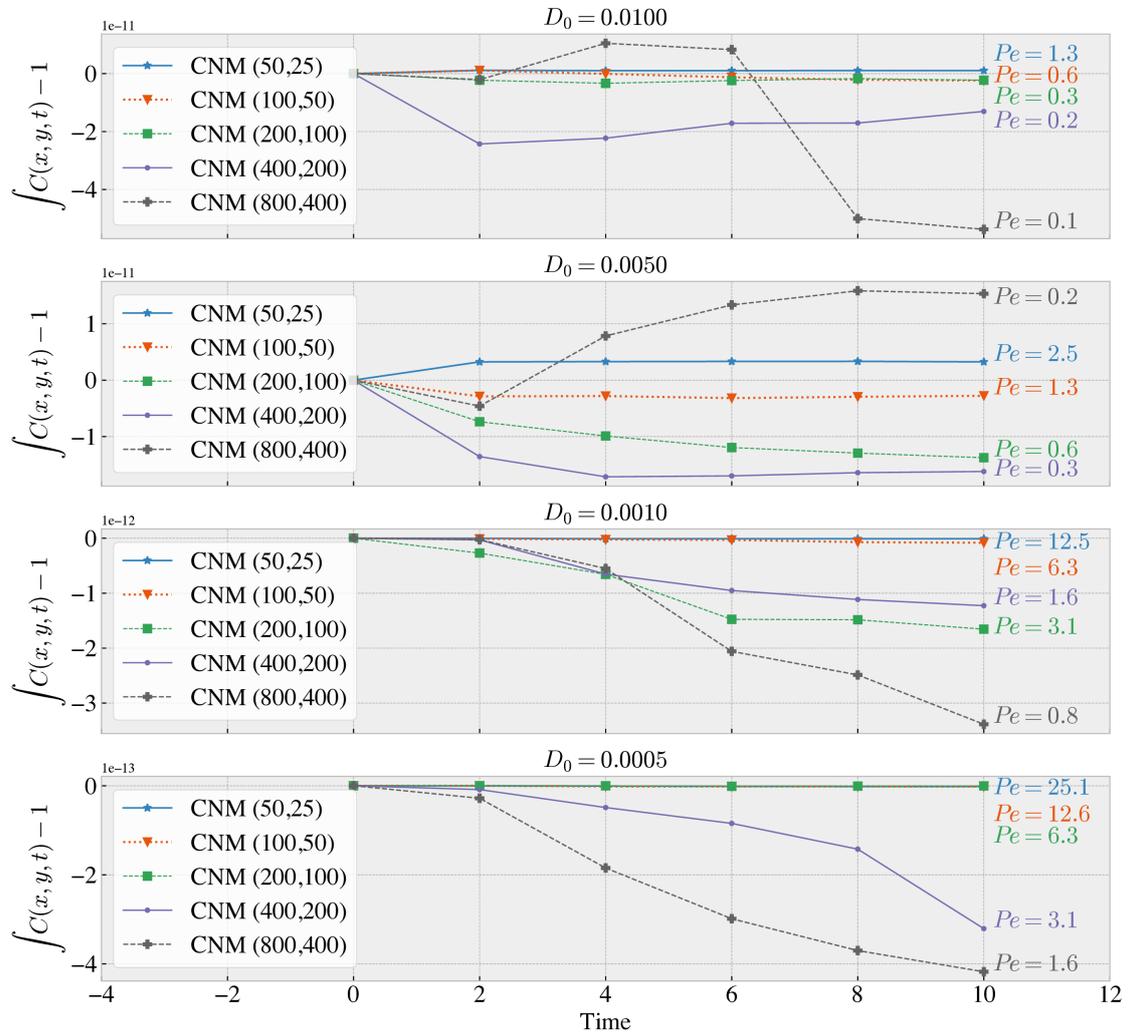
**Figure 3.2.1:** Log-log plot of the convergence in time of the Crank-Nicolson scheme, for four different diffusion constants  $D_0$ , on a  $(200 \times 400)$ -grid. The integrated squared error is relative solutions using a small time step,  $\Delta t = 0.1 \cdot (1/2)^9$ . The slopes are calculated by taking the logarithm of the dataset (for each  $D_0$ ), and computing a linear regression. Only filled data points are used in the regression analyses.



**Figure 3.2.2:** Log-log plot of the convergence of the Crank-Nicolson method in space, for four different diffusion constants  $D_0$  and four different discretizations. The integrated squared error (ISE) is calculated relative a finer discretization, a  $(400 \times 800)$ -grid. Upper legends apply in both subplots, and Péclet numbers are given in order from finest to coarsest discretization (including for the relative grid of discretization  $\Delta x = 0.0025$ ). Lower legends show the approximated slope – the convergence rate – calculated by direct curve fitting to the function  $f(x, a, b) = ax^b$ . Only filled in data points are used in the curve fitting. The time step is  $\Delta t = 0.1 \cdot (1/2)^9$  for all calculations.

that they are sufficiently small to satisfy  $\Delta t \leq (\Delta x)^2/D_0$ , and selected from the range of time steps  $\Delta t$  in Table 3.0.1. The deviations during an integration time of  $T = 10$ , are of the scale  $10^{-13}$  to  $10^{-11}$ . It shows how simulations with higher Péclet numbers deviate less, and how higher diffusion gives rise to higher deviations.

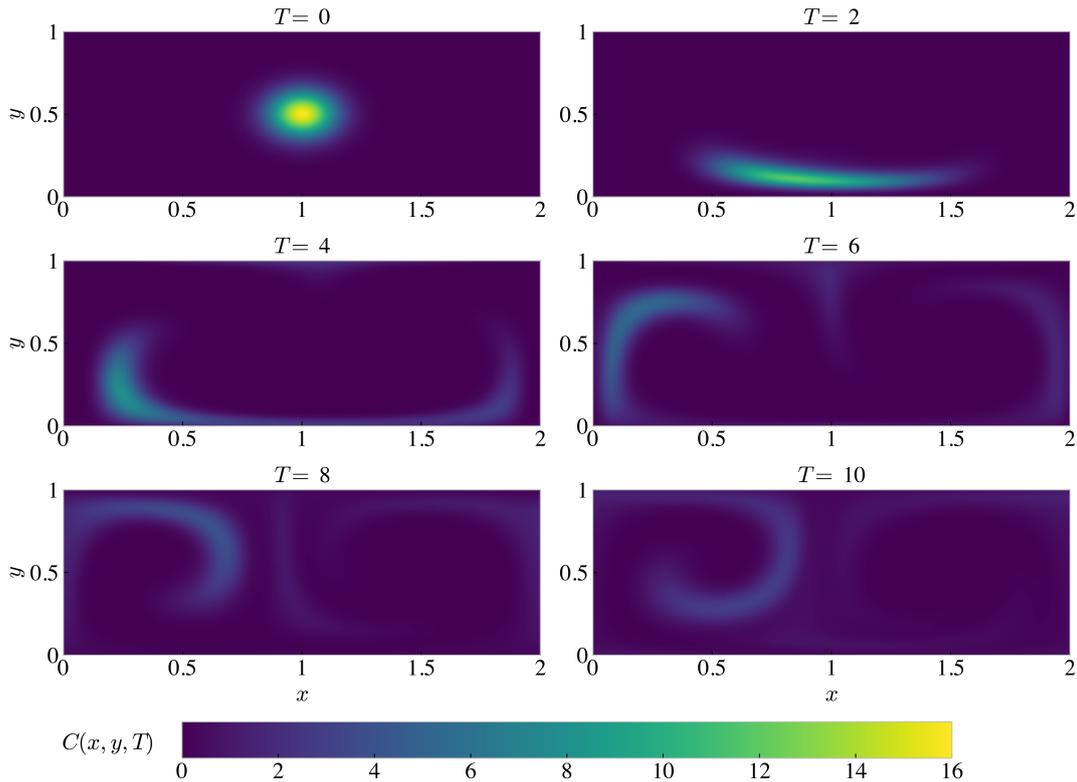
For more details, we also looked at extreme values of the solutions. We found the minima of the distribution in all four diffusive systems. Plots showing the minima are found in Appendix D, Figures D.4a and D.4b. They show that in systems with lower diffusion,  $D_0 = 0.001$  and  $D_0 = 0.0005$ , the concentration solution contains unphysical negative numbers. In all but one subplot, we find negative values as the minima of the simulations at some time in the integration. This oscillating behavior was expected due to higher cell Péclet numbers, but is also present in calculations with assumed sufficiently low cell Péclet numbers. The distributions have been plotted at different times  $T$  to observe these oscillations, and are found in Appendix D, Figures D.5-D.8. The striking feature is particularly visible in Figure D.8a, describing the same system as the top right subplot in Figure D.4b.



**Figure 3.2.3:** The deviation from original mass ( $\iint C(x, y, 0) dx dy = 1$  over the domain) as a function of time, for Crank-Nicolson method (CNM), with grid size  $(N_x, N_y)$ . To the right, in colors corresponding to the descriptions in the legend, the maximum cell Péclet number is shown.

### 3.2.3 Concentration solution

From an initially Gaussian distribution, the fluid method gives the solution as a distribution on a specific grid. As already mentioned, subplots of the results can be found in Appendix D (Figures D.5-D.8), in a small scale, for a variety of diffusion constants, grid sizes and times. For a grid of size  $(200 \times 400)$ , the results are plotted for all four diffusion constants. Figure 3.2.4 shows the system with the lowest diffusion, and thus most clear features. The thin filaments expected in the double gyre flow are present, but the features are rapidly diffusing even in this system with relatively low diffusion. For higher diffusion constants  $D_0 = \{0.001, 0.005, 0.01\}$ , the results are given in Figures D.2b-D.3b in Appendix D.



**Figure 3.2.4:** The probability density distribution at different integration times  $T$ , resulting from solving the advection-diffusion equation using Crank-Nicolson on a  $(200 \times 400)$ -grid for the smallest time step ( $\Delta t = 0.0001953125$ , see Table 3.0.1).

## 3.3 PARTICLE METHOD: NUMERICAL IMPLEMENTATION

To solve the advection-diffusion equation in the Lagrangian formulation, the stochastic differential equation (SDE), Eq. (2.2.29),

$$dX_t = a(X_t)dt + b(X_t)dW_t \quad \text{with} \quad a = u + \partial_x D, \quad \text{and} \quad b = \sqrt{2D},$$

has been solved for a large number of simulations. This SDE was found in Section 2.2.4 to have a Fokker-Planck equation that is equivalent to the advection-diffusion equation. To solve the SDE we have used the time-discrete approximation Euler-Maruyama, with periodic boundary conditions applied. The system is integrated with discrete time steps  $\Delta t$  and a Wiener process associated with the step size. The probability density function is found using kernel density estimation (KDE).

This section shows all implementations in as much detail as needed to replicate, without necessarily providing the actual code. The outline of this implementation section includes the derivation of the Euler-Maruyama scheme in two dimensions, boundary conditions, and approximation of derivatives using finite differences. Running the scheme sufficiently many times (enough particles), as well as wishing to save the state of the system at particular times, require some memory. The section ends with a discussion about the resolution of the Wiener process for different time discretizations, and how the problem has been partitioned to stay within reasonable computational time. Implementations relevant for KDE are not presented, and are put off to the next chapter where it plays a larger role (see Chapter 4.1). Results from computations will be presented in the next section, but first, a thorough presentation of the implementation.

### 3.3.1 Two-dimensional Euler-Maruyama scheme

We used Euler-Maruyama to integrate the two-dimensional stochastic process  $\mathbf{X}_t$  for a large number of Lagrangian particles, resulting in a set of solutions that together can be interpreted through KDE as a concentration. In the following, the implementation of the two-dimensional scheme is explained, along with boundary conditions. Theoretical concepts in this section will not be introduced further. For introductions see Chapter 2.2 for stochastic differential equations (SDEs); including the Itô process (Section 2.2.3), the Fokker-Planck equation (Section 2.2.4), and an introduction to the Wiener process (Section 2.2.2). The Euler-Maruyama scheme is presented in Chapter 2.3.1, and a general introduction to Monte-Carlo methods is given in Chapter 2.4.

For the two-dimensional Itô process (Eq. (2.2.19)),

$$d\mathbf{X}_t = \mathbf{a}(\mathbf{X}_t)dt + \mathbf{b}(\mathbf{X}_t)d\mathbf{W}_t,$$

the coefficients that still ensures that its Fokker-Planck equation describes the time evolution of a probability density function equivalent to the advection-diffusion equation, are

$$\mathbf{a} = \begin{bmatrix} u + \partial_x D \\ v + \partial_y D \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \sqrt{2D} \\ \sqrt{2D} \end{bmatrix}. \quad (3.3.1)$$

The Euler-Maruyama scheme is then, for the two-dimensional position  $\mathbf{X} = \begin{bmatrix} x \\ y \end{bmatrix}$  written

$$\mathbf{X}_n = \begin{bmatrix} x_{n-1} + (u + \partial_x D)\Delta t + \sqrt{2D}\Delta W_n^x \\ y_{n-1} + (v + \partial_y D)\Delta t + \sqrt{2D}\Delta W_n^y \end{bmatrix} = \mathbf{X}_{n-1} + \mathbf{a}\Delta t + b\Delta\mathbf{W} \quad (3.3.2)$$

for  $b = \sqrt{2D}$ . Each increment of the Wiener process,  $\Delta W^k$ , is  $\mathcal{N}(0, \Delta t)$ . The derivatives of the diffusion function,  $\partial_x D$  and  $\partial_y D$ , are calculated using forward difference with a difference  $\delta x$  and  $\delta y$  set to  $10^{-6}$ .

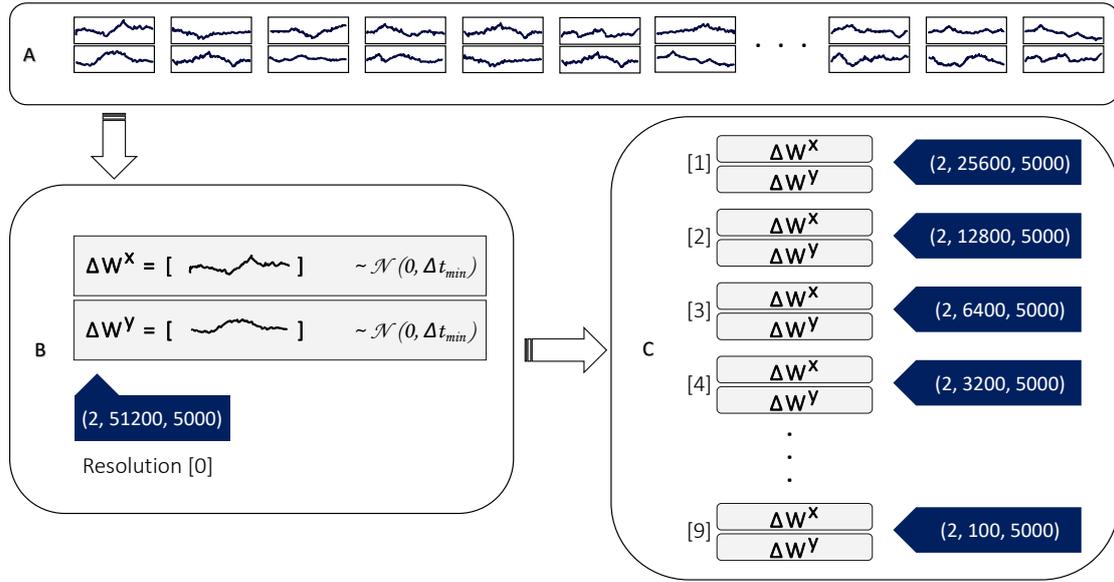
#### Boundary conditions

The iterative scheme (3.3.2) is implemented straightforwardly, using periodic boundary conditions in the spatial dimensions. This ensures that any particle leaving the domain on either side reappears on the opposite side,

---


$$\begin{aligned} \mathbf{X}[0] &= \mathbf{X}[0]\%Lx \\ \mathbf{X}[1] &= \mathbf{X}[1]\%Ly \end{aligned}$$


---



**Figure 3.3.1:** Illustration of adapting the Wiener process to the temporal resolution at hand. **A:** 20 files containing the cumulative sum of random normal increments of scale  $\sqrt{\Delta t_{min}}$ . **B:** From each file, the increments of the finest resolution Wiener process can be found for all  $Np = 5000$  number of particles. **C:** The same file is used to calculate the increments of coarser Wiener processes (larger time steps). In total, using all files in **A**, results in 100 000 Wiener processes, all expressed in 10 different resolutions (numbered in the figure from highest to lowest resolution).

where  $Lx$  and  $Ly$  are domain lengths, and  $\mathbf{X.shape} = (2, Np)$  for two dimensions and  $Np$  particles, where  $\mathbf{X}$  holds the particle positions.

### 3.3.2 The Wiener process for multiple resolutions

The two-dimensional Wiener process  $\mathbf{W}_t$  is used in the Euler-Maruyama scheme (3.3.2). For each iterative step in the scheme, two increments  $\Delta W$  of the Wiener process is required, one in each dimension. Because the process depends on the discretization,  $\Delta W \sim \mathcal{N}(0, \Delta t)$ , the implementation is flexible for different time steps  $\Delta t$ . It is based on creating and accessing files for the finest temporal resolution that is available (Table 3.0.1).

The idea is, illustrated through Figure 3.3.1, to first (**A**) create wiener processes in two dimensions

---

```

dW = np.random.normal(loc = 0, \
                      scale = np.sqrt(dt_min), \
                      size = (2, Nt_max, Np))

W = np.cumsum(np.insert(dW, 0, \
                       np.zeros(dW[0,0,:].shape), axis=1), axis = 1)

```

---

for the finest resolution that we wish to look at; for `Nt_max` number time steps, and `dt_min` step size. It is saved to file as the cumulative sum.

In total, we ran the procedure 20 times, creating  $20 \cdot Np$  two-dimensional Wiener processes over the time interval  $[0, Nt\_max \cdot dt\_max]$  (see Table 3.0.1). Number of particles per round is set to `Np = 5000` to keep computational time practical, resulting in 100 000 available particles in total.

Provided with the cumulative sum from file (Figure 3.3.1, **B**), the coarser resolution process increments can be found by combining increments:

---

```

Nskip = int(dt/self.dt_min)
# Get DW_n at all time steps n
DW     = W[:,::Nskip, :][:, 1:] - W[:,::Nskip, :][:, :-1]

```

---

Each of the resulting arrays (Figure 3.3.1, **C**) of shape  $(2, Nt, Np)$  describe the same process as in **B** (when they origin from the same process in **A**), only with a coarser resolution. E.g., the coarsest resolution gives the Wiener process increments only at 100 time-instants. A severe downside to the method is that the number of time steps must be provided beforehand, however, this is a prerequisite if we want to investigate the convergence of the Euler-Maruyama method in the strong sense (see Section 2.3.2).

### 3.3.3 Monte Carlo simulations

The handling of the Wiener process for a range of different time step sizes, sets the approach of the simulations. One *simulation* is iteratively computing, for  $Nt$  time steps,

$$\mathbf{X}_n = \mathbf{X}_{n-1} + \mathbf{a}\Delta t + b\Delta \mathbf{W},$$

for  $\mathbf{X}$ .shape =  $(2, 1)$  describing a single coordinate in two dimensions. The *Monte Carlo* simulation is repeatedly computing realizations of the above approximation. The total number of simulations is denoted  $N$ , and is also called the total number of *Lagrangian particles*, or just the number of particles. The simulations are split up into  $M$  runs, with  $Np$  particles per run, given in Table 3.3.1.

**Table 3.3.1:** The setup in the Lagrangian particle model includes  $Np$  number of particles per run, for  $M$  runs. This makes a total of  $N_{max}$  Lagrangian particles.

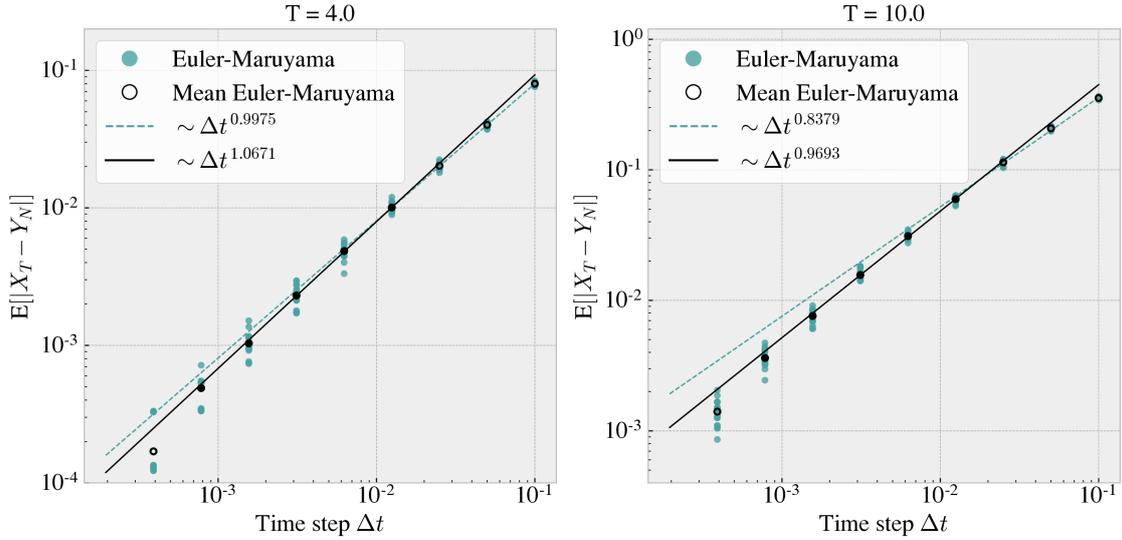
$Np$	$M$	$N_{max}$
5000	20	100000

## 3.4 PARTICLE METHOD: NUMERICAL RESULTS

In this section, we show the results from computing the probability density function as it evolves in time in the Lagrangian formulation. The implementation is fully described in the previous section, and only relevant parameters and constants are given here. The advection-diffusion equation is solved through Monte Carlo simulations of the appropriate stochastic differential equation, using the discrete-time approximation Euler-Maruyama for integration. The flow velocity field is the double gyre flow, and the diffusion is set to be constant over the domain,  $D(x, t) = D_0$ . From an initial Gaussian distribution, we consider the convergence of the Euler-Maruyama method, both in a weak and a strong sense. We also look at how kernel density estimation (KDE) is used to transform the simulations of separate Lagrangian particles into a probability density function.

### 3.4.1 Strong and Weak Convergence

The convergence of the Euler-Maruyama scheme is calculated in both the weak and strong sense (see Section 2.3.2). The Figures 3.4.1 and 3.4.2, confirm that the Euler-Maruyama scheme has a strong convergence



**Figure 3.4.1:** Strong convergence of the Euler-Maruyama scheme after a time  $T$ , at different time steps  $\Delta t$ , for a constant diffusion  $D_0 = 0.001$ . Showing results from 20 simulations, each of 5000 realizations, along with the mean of these simulations. The slopes of the lines are computed using curve fitting to the function  $f(\Delta t) = a(\Delta t)^b$ . The dashed line is a curve fit for all simulations, while the full line only uses the average value at each time step for the curve fit, and only calculated for filled data points.

$$\mathbb{E}[|X_T - Y_N|] \leq K(\Delta t)^\gamma, \quad \gamma \approx 1,$$

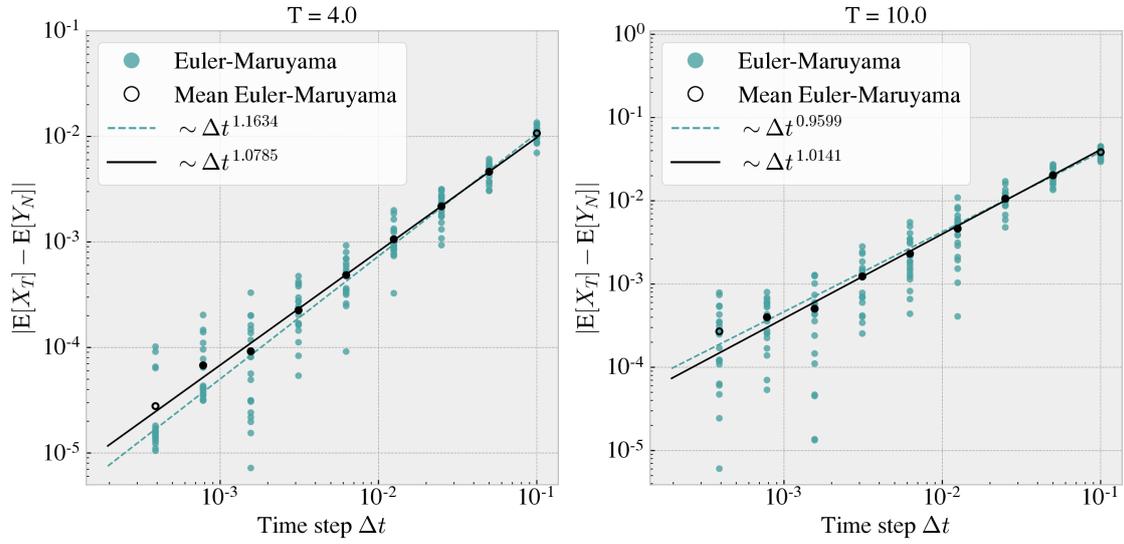
and a weak convergence

$$|\mathbb{E}[X_T] - \mathbb{E}[Y_N]| \leq K(\Delta t)^\beta, \quad \beta \approx 1,$$

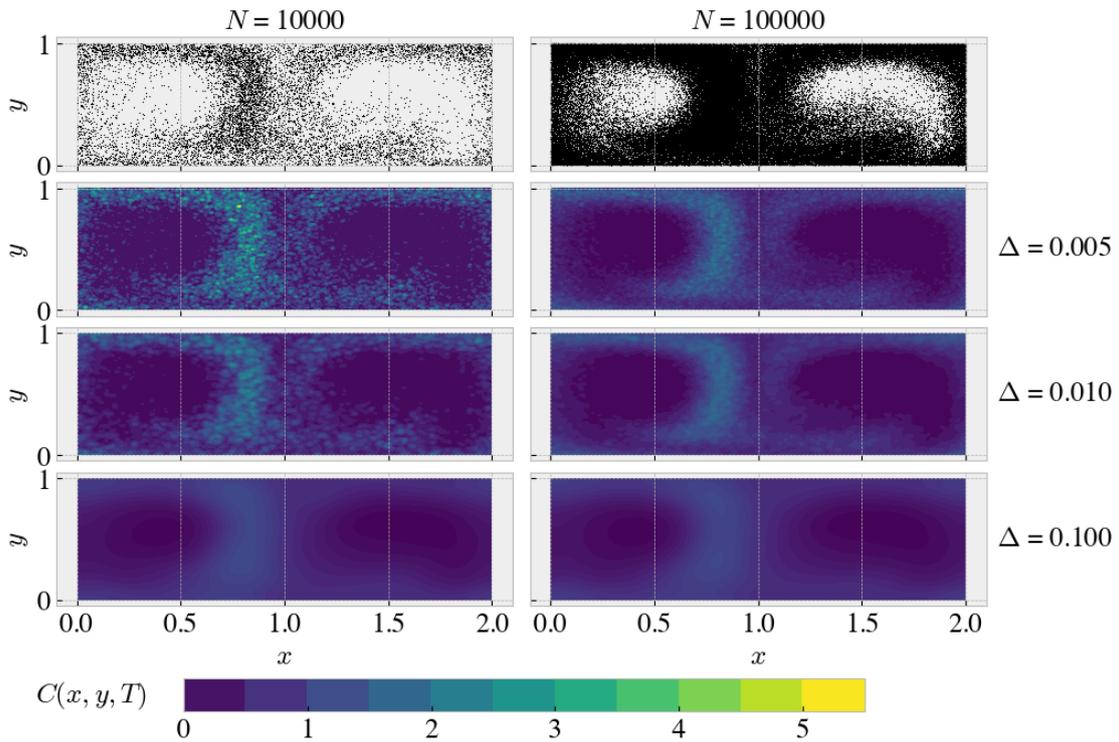
as anticipated for a constant diffusion coefficient. This result comes from simulations with 100 000 Lagrangian particles, at both  $T = 4$  and  $T = 10$ , and the rate of convergence in respectively the strong and weak sense,  $\gamma$  and  $\beta$ , were found through curve fitting.

### 3.4.2 Estimated concentration solution

After the integration time  $T = 10$ , the particles are distributed as seen in the first subplots of Figure 3.4.3. Attempting to use the Monte Carlo simulations of 10 000 and 100 000 Lagrangian particles to approximate a distribution through kernel density estimation (KDE) result in different distributions as seen in Figure 3.4.3. The distributions are highly sensitive to the relationship between the bandwidth selected, and the number of particles used in the estimation. The bandwidth is a crucial parameter of KDE, that we will look at more closely in the next chapter, Chapter 4.



**Figure 3.4.2:** Weak convergence of the Euler-Maruyama scheme after a time  $T$ , at different time steps  $\Delta t$ , for a constant diffusion  $D_0 = 0.001$ . See Figure 3.4.1 for further explanation of setup.



**Figure 3.4.3:** Contour plot of the Euler-Maruyama scheme results for 10 000 and 100 000 Lagrangian particles, after an integration time  $T = 10$ . The diffusion is constant,  $D_0 = 0.001$ , and the time step is  $\Delta t = 0.05$ . First row is a scatter plot of the particles. Using the results to estimate the probability density distribution result in the three lowest rows of the plot. Kernel density estimation (KDE) with bandwidth,  $\Delta$ , set to three different values.

## Chapter 4

# BANDWIDTH SELECTION IN KERNEL DENSITY ESTIMATION

This chapter compares solutions from the Eulerian fluid method and the Lagrangian particle method for solving the advection-diffusion equation, through kernel density estimation of the particle method results. The system is the same as implemented in Chapter 3; a double gyre flow with constant diffusion. The optimal width of the kernel function, the co-called bandwidth, is found by minimizing the integrated squared error relative to the Eulerian fluid solution.

### 4.1 IMPLEMENTATION

The fluid method gives the solution on a grid: the probability density function describing the concentration over the domain. This grid was thoroughly described in Section 3.1.1. The particle method returns  $N$  (number of simulations) two-dimensional coordinates inside the domain, that can be viewed as samples from the solution distribution. To compare the two solvers, we estimate a density function based on the samples by using a Gaussian kernel density estimator (see Section 2.4.3).

#### 4.1.1 Kernel Density Estimator

The density function based on the samples from the particle method was estimated using **FFTKDE** from the **KDEpy** package, which takes in a scalar bandwidth through the parameter **bw**, used for all samples. To account for periodic boundary conditions, nine duplicates of the particle method solution were used. This means that eight copies of the original particle solution are placed around the original solution, surrounding the original domain. Consequently, the correct boundary conditions are applied when computing the kernel density estimate for the entire new domain,  $x \in [-2, 4]$  and  $y \in [-1, 2]$ . See Appendix C.1 for a full description of the implementation of the estimator, including the boundary conditions.

#### 4.1.2 Implementation of double gyre system with constant diffusion

In the two following sections, Sections 4.2 and 4.2.3, we continue from last chapter by further investigating the solution of the advection-diffusion equation in a system with double gyre flow and constant diffusion. In Chapter 3, this system was described for a number of parameters. To keep the implementations simpler in this chapter, i.e., to focus on other parts of the simulations, we have set the values of both the grid size and the time step. The following parameters have been used.

• Discretization grid (fluid method), $(Ny \times Nx)$	$(200 \times 400)$
• Lagrangian particles per simulation (particle method), $Np$	5000
• Maximum number of runs (particle method), $M$	20
• Range of total number of particles (particle method), $N$	$\{100, \dots, 100000\}$
• Time step $\Delta t$	$0.00019531 (= 0.1 (\frac{1}{2})^9)$
• Set of diffusion constants, $\{D_0\}$	$\{0.01, 0.005, 0.001, 0.0005\}$
• Set of available times, $\{T\}$	$\{0, 2, 4, 6, 8, 10\}$

For a better understanding of the maximum number of runs  $M$ , Lagrangian particles per simulation  $Np$ , and total number of particles  $N$ , refer to description in Section 3.3.3, Table 3.3.1, where this is explained through the implementation of the particle method. The double gyre flow parameters are the same as in the previous chapter, given in (3.0.1).

### Curve-fitting problems

To investigate the dependency of the optimal bandwidth with respect to number of particles used by the particle method, the data can be fitted to a function of choice using curve-fitting. After some experimentation, we settled on the functional form

$$f(N) = \Delta_{optimal} = \frac{a}{N^b} + c, \quad (4.1.1)$$

for which the data points  $(N, \Delta_{optimal})$ , for specific systems given by the diffusion coefficient  $D = D_0$ , are attempted to fit. We have set the bounds to  $a \in [0, \infty)$ ,  $b \in [0, \infty)$ ,  $c \in [0, 10^{-10}]$ , using the following method:

---

```

from scipy.optimize import curve_fit

def function(x, a, b, c):
    return a*(x**(-b))+ c

opt_parms, parm_cov = curve_fit(function, Np_vec, optimal_bw, \
                                bounds = ([0,0,0 ], [np.inf, np.inf, 1e-10]))
a, b, c = opt_parms

```

---

## 4.2 OPTIMAL BANDWIDTH IN DOUBLE GYRE SYSTEM

This section presents the results from experiments described in Section 4.1.2: Comparing solutions of the advection-diffusion equation in the double gyre system with constant diffusion. We focus on the particle method, investigating the bandwidth of the kernel density estimator. The bandwidth is optimized with regard to the integrated squared error (ISE) relative to the fluid solution. In following, we present results from computations of the ISE for four different diffusion constants:  $D_0 = \{0.01, 0.005, 0.001, 0.0005\}$ . We look at the behavior of the error through the integration period by investigating the error as a function of bandwidth at different times  $T = \{2, 4, 6, 8, 10\}$ . We will also look at the optimal bandwidth as a function of both diffusion, time, and the number of simulations available.

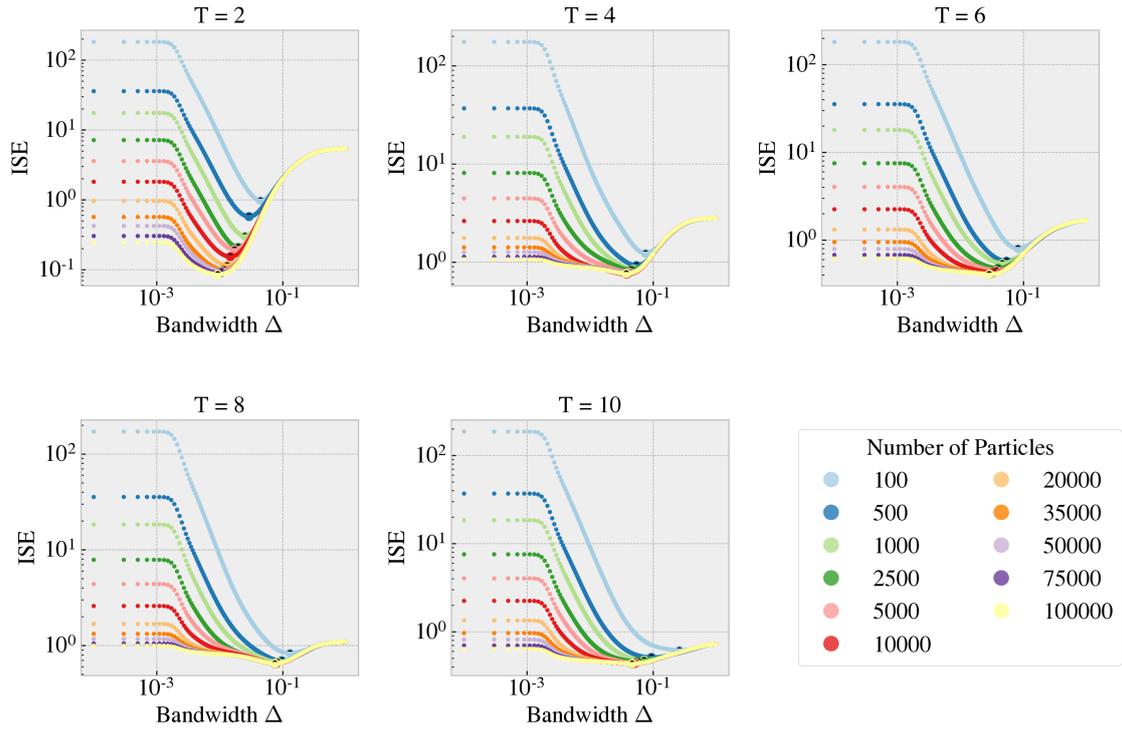
### 4.2.1 Optimal bandwidth with regards to ISE

Figures 4.2.1-4.2.4 give the integrated squared error (ISE) as a function of bandwidth in systems with different diffusion coefficients  $D(x, y) = D_0$ . The overall trend shows a constant ISE for the lowest bandwidths before the error reaches a minimum in most subplots, and then increases again with the bandwidth. The optimal bandwidth  $\Delta_{optimal}$  is considered to be the bandwidth for which the ISE has a minimum.

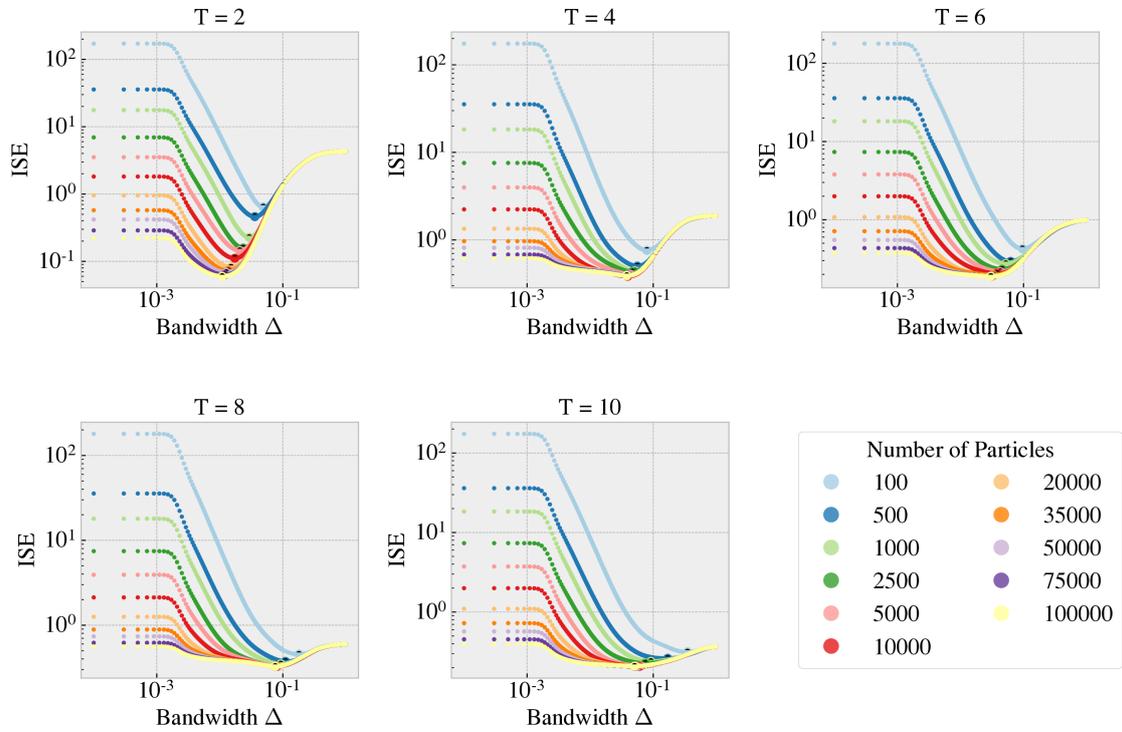
Bandwidths smaller than the size of the grid cells used to compute the kernel density estimation,  $\Delta < \Delta x$ , give a constant ISE. As we have used a square grid and symmetric kernels in our estimations, then any kernel with bandwidth set to be smaller than the cell length will be integrated as part of the density of that cell no matter the bandwidth, as long as the entire kernel falls within the cell. For Figures 4.2.1-4.2.4, the spatial grid has discretization  $\Delta x = \Delta y = 5 \cdot 10^{-3}$ , clearly concurring with the behaviour observed in the subplots.

The optimal bandwidth has a decreasing trend with number of particles; the optimal bandwidth is lower when the density estimation is based on more Lagrangian particles. As a function of time, the optimal bandwidth appears to increase from the first subplot at time  $T = 2$  to the maximum integration time  $T = 10$ . It is also noticeable from the figures that the ISE is lower for simulations with many particles versus fewer particles, particularly in proximity to the optimal bandwidth and for even lower bandwidths. For higher values of the bandwidth, the errors appear quite coinciding. Another noticeable feature is that the ISE appears to generally be higher in systems with lower diffusion relative systems with higher diffusion, as seen for  $N = 100000$  in Table D.1.

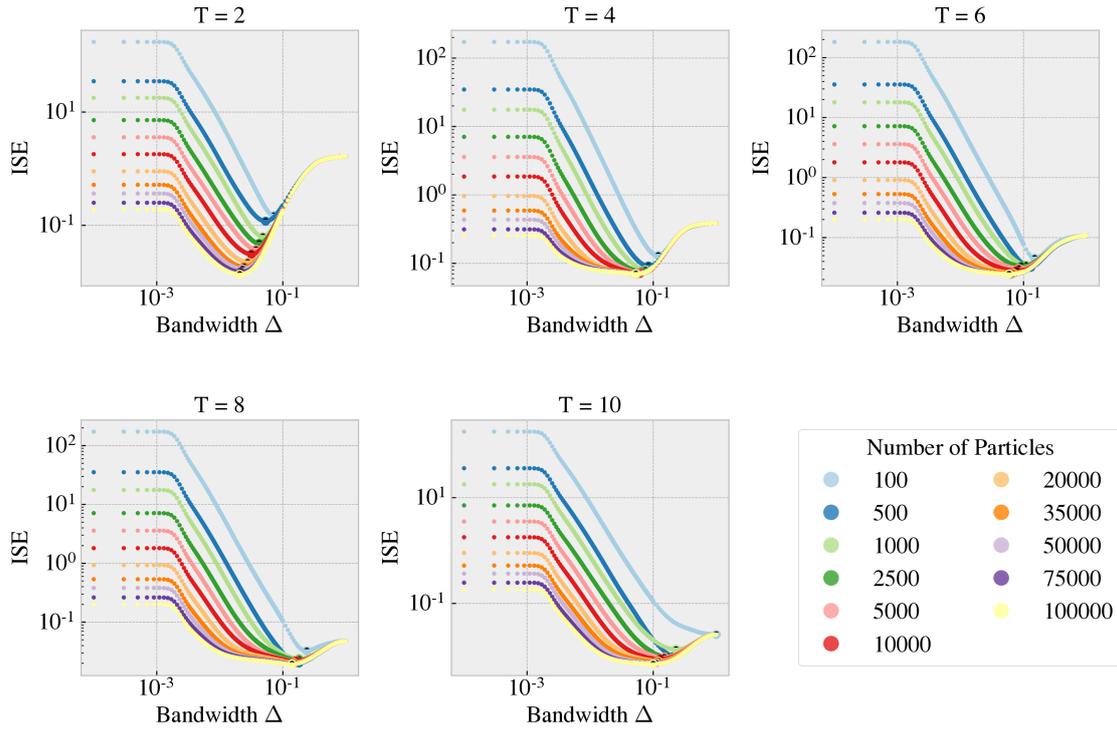
Comparing Figures 4.2.1-4.2.4 qualitatively, we find that the larger the diffusion is, the larger is also the optimal bandwidth. In Figure 4.2.3 ( $D_0 = 0.005$ ) and Figure 4.2.4 ( $D_0 = 0.01$ ), the solution presented for  $N = 100$  Lagrangian particles shows a concerningly large optimal bandwidth, approaching the size of the system, and potentially larger than the range of bandwidths tested. It is a clear sign that the system is not properly described by such few particles, at least for diffusion of this scale, and after an integration time sufficiently large to smear out the system. Too few particles will not be able to describe properly the remaining structures of the distribution, and relies on high bandwidths to represent the system as well-mixed instead.



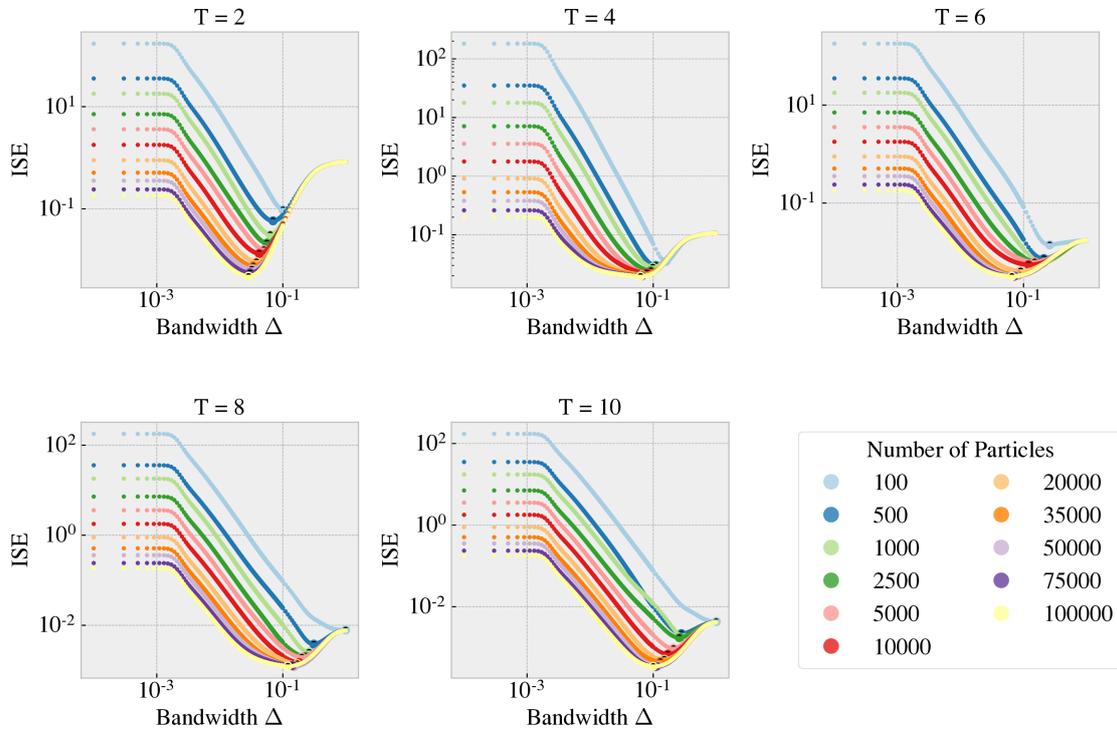
**Figure 4.2.1:** Plot of the integrated squared error (ISE) between the fluid method solution and the kernel density estimate of the particle method solution using  $N$  number of particles, in the double gyre system. The systems are simulated for  $D_0 = 0.0005$ ,  $\Delta t \approx 0.0002$ , on a  $(200 \times 400)$ -grid, and plotted for a time  $T$ .



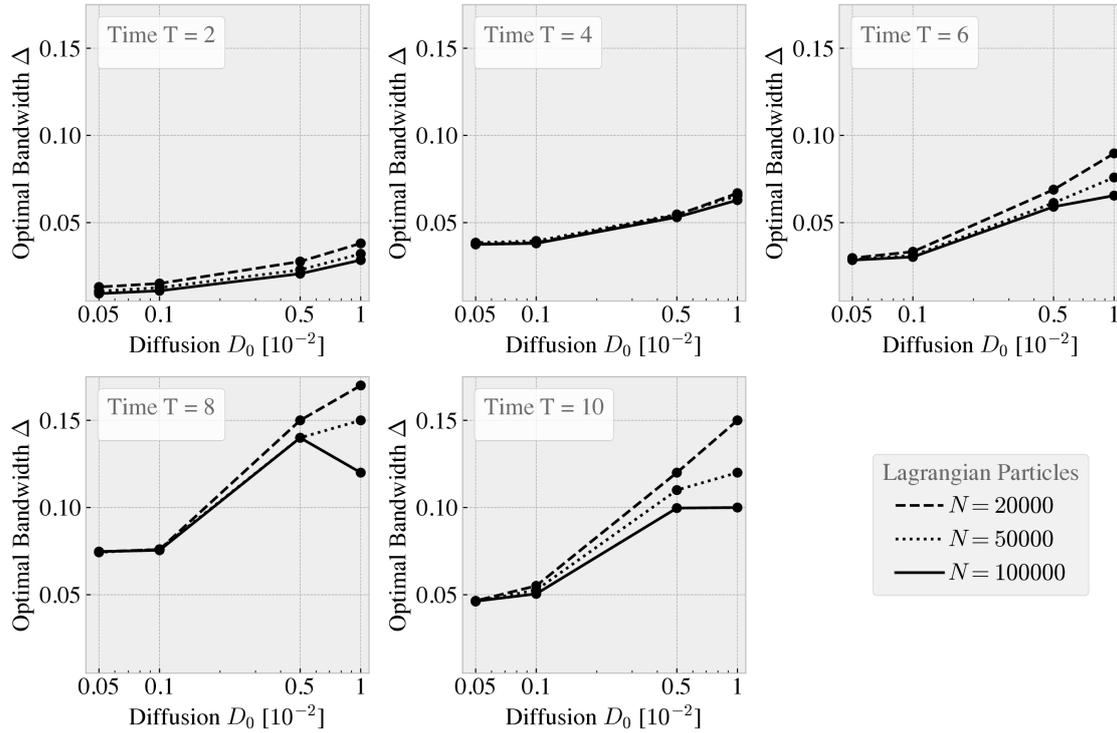
**Figure 4.2.2:** Plot of the integrated squared error (ISE) between the fluid method solution and the kernel density estimate of the particle method solution using  $N$  number of particles, in the double gyre system. The systems are simulated for  $D_0 = 0.001$ ,  $\Delta t \approx 0.0002$ , on a  $(200 \times 400)$ -grid, and plotted for a time  $T$ .



**Figure 4.2.3:** Plot of the integrated squared error (ISE) between the fluid method solution and the kernel density estimate of the particle method solution using  $N$  number of particles, in the double gyre system. The systems are simulated for  $D_0 = 0.005$ ,  $\Delta t \approx 0.0002$ , on a  $(200 \times 400)$ -grid, and plotted for a time  $T$ .



**Figure 4.2.4:** Plot of the integrated squared error (ISE) between the fluid method solution and the kernel density estimate of the particle method solution using  $N$  number of particles, in the double gyre system. The systems are simulated for  $D_0 = 0.01$ ,  $\Delta t \approx 0.0002$ , on a  $(200 \times 400)$ -grid, and plotted for a time  $T$ .

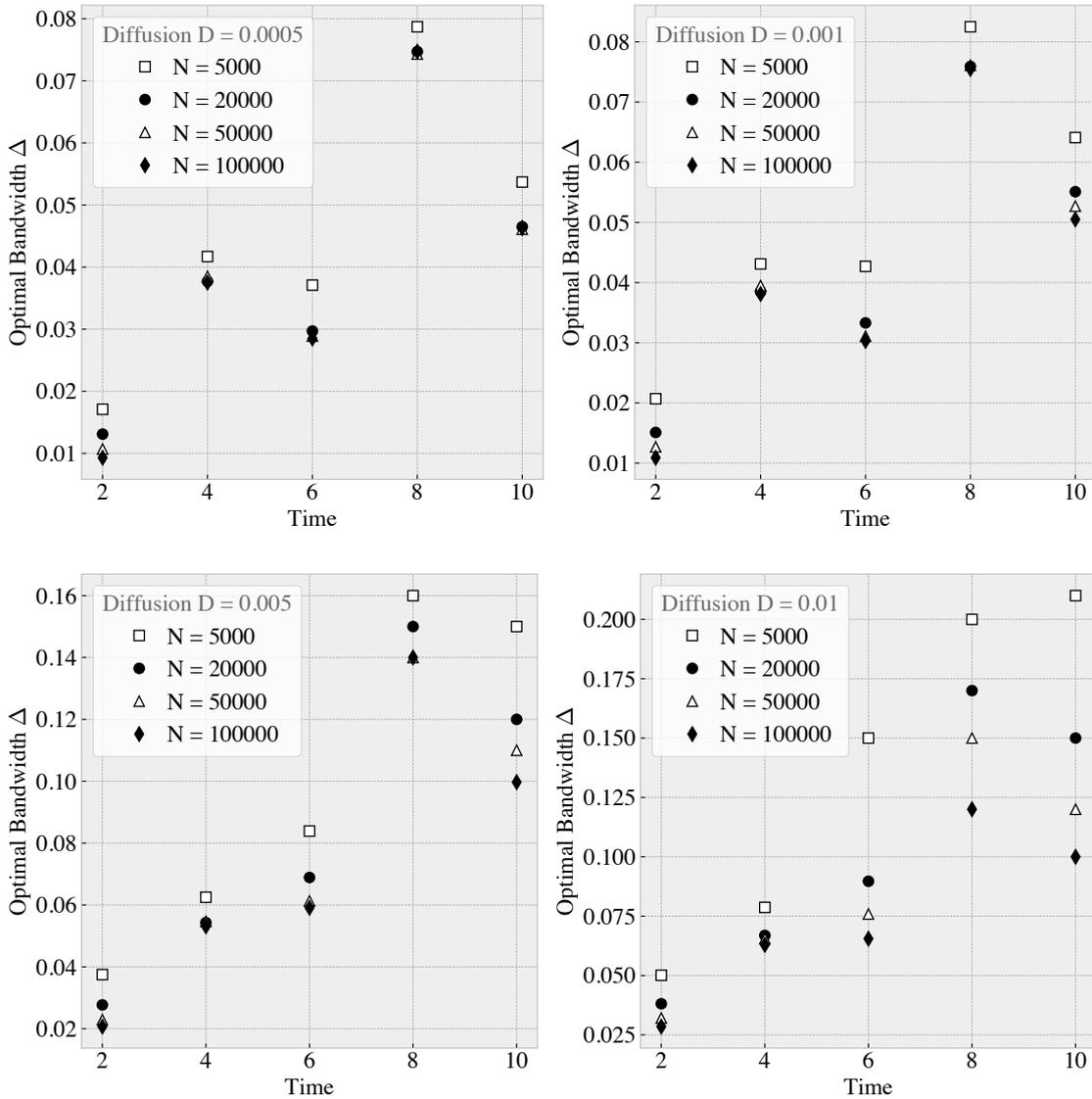


**Figure 4.2.5:** The optimal bandwidth  $\Delta_{\text{optimal}}$  as a function of diffusion constant  $D_0$ .  $N$  is the number of Lagrangian particles (number of simulations) in the particle method, and  $T$  is integration time. These calculations are done on a  $(200 \times 400)$ -grid, for a time step  $\Delta t \approx 0.0002$ . Note that the x-axis is plotted on a logarithmic scale.

#### 4.2.2 Optimal bandwidth as a function of diffusion and time

Figure 4.2.5 shows the optimal bandwidth as a function of diffusion for a selected few cases. The trend is increasing for higher diffusion. The figure shows a larger spread in optimal bandwidths for higher diffusion in systems with different number of particles  $N$ . The relationship between the optimal bandwidth and the number of Lagrangian particles is investigated further in the next section.

The optimal bandwidth as a function of time is presented in Figure 4.2.6, see also Table D.1 in Appendix D for  $N = 100000$  specifically. The optimal bandwidth shows an increasing trend, but the results are somewhat ambiguous as they show what we can consider oscillating behavior, with alternating low and high values. These might be effects due to the periodic velocity field, as this is particularly visible in the systems with lower diffusion and thus more advection-dominated. The flow field has a period of 10, and thus might be behind the oscillations in the subplots in Figure 4.2.6, where high values occur between the time when the field has a maximum perturbation ( $t = 2.5, 7.5$ ) and when it is back to a central position ( $t = 0, 5, 10$ ). However, the premise is questionable and would require many data points to base a conclusion on.



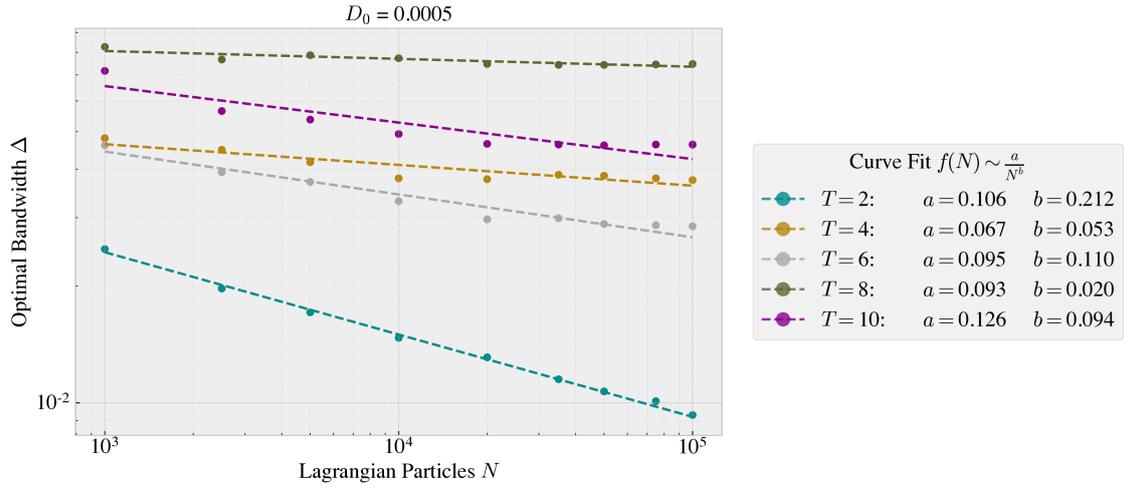
**Figure 4.2.6:** Optimal bandwidth as a function of time, for four cases with different number of Lagrangian particles used in simulations.

### 4.2.3 Optimal bandwidth as a function of Lagrangian particles

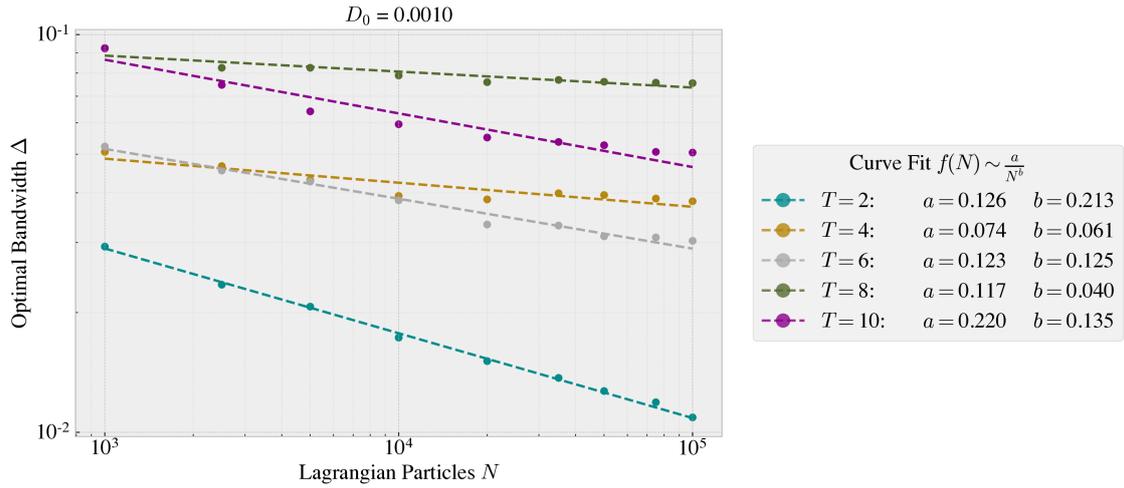
We have estimated the relationship between the optimal bandwidth and the number of Lagrangian particles, and assumed (see Eq. (4.1.1) in implementations) that the relationship can be described through a function inversely proportional to the number of Lagrangian particles

$$\Delta_{optimal} \approx \frac{a}{N^b}. \quad (4.2.1)$$

With results from the previous section fresh in mind, we choose to only use data from simulations with  $N \geq 1000$ , to keep clear of too low number of particles in our simulations. Figures 4.2.7-4.2.10 show the curve fit to the function (4.1.1), and the resulting parameters  $a$  and  $b$ , giving constant slopes in a log-log representation. The parameter  $b$  describes the rate at which the optimal bandwidth falls,  $(\frac{1}{N})^b$ , and is further presented in Table 4.2.1, as a function of both time and diffusion. The mean of all parameter estimations of  $b$  presented in Table 4.2.1 is



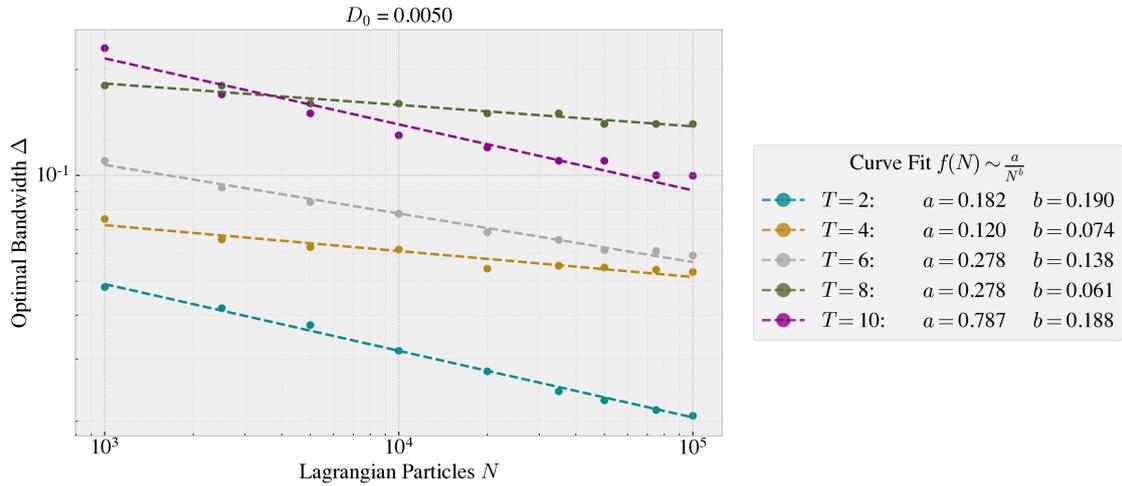
**Figure 4.2.7:** The optimal bandwidth  $\Delta_{optimal}$  versus total number of particles  $N$  used in particle method, with a curve fit to function  $f(N)$ . The curve fit parameters are given in the legend. The curve fit parameter estimation is here made for the double gyre system with constant diffusion  $D_0 = 0.0005$ , at times  $T$ .



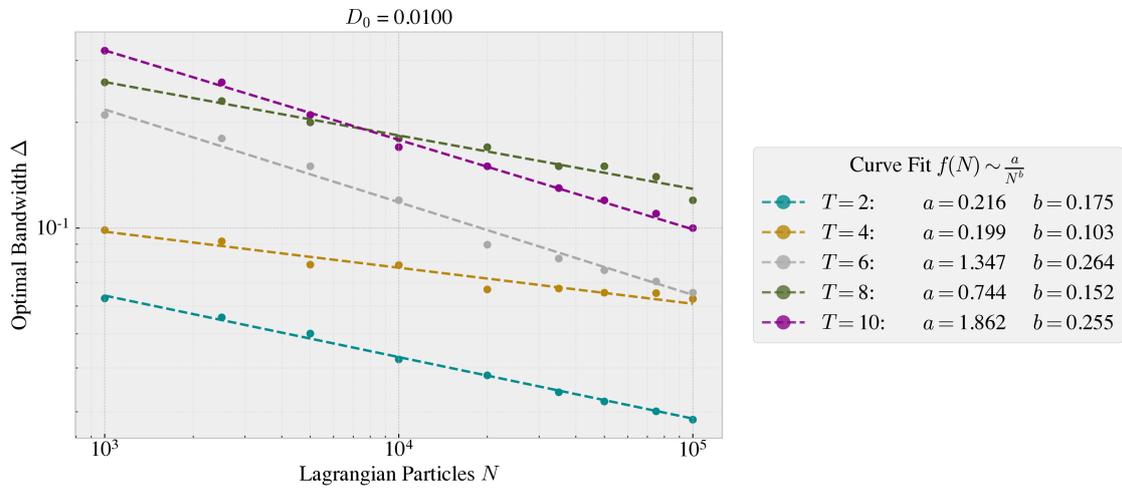
**Figure 4.2.8:** The optimal bandwidth  $\Delta_{optimal}$  versus total number of particles  $N$  used in particle method, with a curve fit to function  $f(N)$ . The curve fit parameters are given in the legend. The curve fit parameter estimation is here made for the double gyre system with constant diffusion  $D_0 = 0.001$ , at times  $T$ .

$$b = 0.133 \pm 0.070,$$

given with one standard deviation of the data used. Averaged values over all diffusion values for particular times, or all times for a particular value diffusion constant, give lower variances. At early integration times ( $T = 2, 4$ ) the variance of the parameter estimations is lower than for later times, as we see in Table 4.2.1.



**Figure 4.2.9:** The optimal bandwidth  $\Delta_{optimal}$  versus total number of particles  $N$  used in particle method, with a curve fit to function  $f(N)$ . The curve fit parameters are given in the legend. The curve fit parameter estimation is here made for the double gyre system with constant diffusion  $D_0 = 0.005$ , at times  $T$ .



**Figure 4.2.10:** The optimal bandwidth  $\Delta_{optimal}$  versus total number of particles  $N$  used in particle method, with a curve fit to function  $f(N)$ . The curve fit parameters are given in the legend. The curve fit parameter estimation is here made for the double gyre system with constant diffusion  $D_0 = 0.01$ , at times  $T$ .

**Table 4.2.1:** Optimization parameters from curve fit to function  $f(N) \sim 1/N^b$ , where  $f(N)$  is the optimal bandwidth  $\Delta_{optimal}$ , and  $N$  is number of Lagrangian particles used in total for simulation. See curve fits and data points in Figures 4.2.7-4.2.10. Average values are given as mean over respectively all columns at a specific times  $(\cdot)_D$ , and all rows at specific diffusion  $(\cdot)_t$ , with one standard deviation of the set of values. Lower right corner (green cell) shows the mean value and standard deviation of all data parameter estimations.

Curve Fit Parameter: $b$					
$T \setminus D$	0.01	0.005	0.001	0.0005	$(Average)_D$
2	0.175	0.190	0.213	0.212	$0.197 \pm 0.016$
4	0.103	0.074	0.061	0.053	$0.073 \pm 0.019$
6	0.264	0.138	0.125	0.110	$0.160 \pm 0.062$
8	0.152	0.061	0.040	0.020	$0.068 \pm 0.050$
10	0.255	0.188	0.135	0.094	$0.168 \pm 0.060$
$(Average)_t$	$0.190 \pm 0.062$	$0.130 \pm 0.055$	$0.115 \pm 0.061$	$0.098 \pm 0.065$	$0.133 \pm 0.070$

## Chapter 5

# DISCUSSION AND CONCLUSION

### 5.1 DISCUSSION

The Advection-Diffusion Equation is commonly applied to solve transport problems in industry as well as in research, and we have in this thesis applied it to an advection-diffusion problem in a periodic domain, implementing a double gyre flow as presented in Shadden et al. (2005), with a constant diffusion coefficient. In this thesis, we investigated the two formulations of fluid motion, Eulerian and Lagrangian; and saw how the advection-diffusion equation can be approached in both formulations. For the Eulerian fluid method and the Lagrangian particle method to give the same solution to advection-diffusion problems, the advection and diffusion coefficients must be "smooth functions." We saw through one- and two-dimensional numerical examples in Chapter 2.5 and Appendix B, how unfulfilled criteria in practice makes the particle method no longer equivalent with the Eulerian fluid method, as it no longer describes the same behaviour as the advection-diffusion equation. Due to the strong deviations from the Eulerian method we have seen in our examples, the consequences presented seem too grave for us not to comment on. In the main numerical work of this thesis we have not had to consider the criteria further, as we have kept the diffusion a constant throughout, and thus implemented a "smooth function" complying with the requirements.

We implemented the Eulerian fluid method in Chapter 3 by using the finite difference method Crank-Nicolson, and presented some unforeseen challenges in terms of restrictions to the model. Eulerian methods tend to be unstable for advection-dominated problems, and the requirement to keep the maximum cell Peclet number sufficiently low is easy to implement in theory to avoid oscillating solutions. However, with limited computing power the resolution required to solve for low diffusion was not achievable, and we opted for a solution with a grid size of  $\Delta x = \Delta y = 0.005$  for most computations. This way we avoided the strongest oscillations that we found for lower grid resolutions in Figures D.7 and D.8, giving unphysical negative values as in Figure D.4b. The chosen discretization resolution was sufficiently fine for the higher diffusion constants we used,  $D_0 = 0.01, 0.005, 0.001$ , but not for the lowest  $D_0 = 0.0005$ , which gave a maximum cell Peclet number larger than 2 ( $Pe_{cell} \approx 3.14$ ). Even though oscillating solutions and constrictions in diffusion scale were a challenge, the scheme presented itself as a robust method with excellent convergence of the Crank-Nicolson scheme in both time and space, and the iterative solver was consistently convergent. It was only for systems with large oscillations, due to coarse grids or low diffusion constants, that the iterative method would not converge.

For the Lagrangian particle method, the two-dimensional Euler-Maruyama scheme was easily implemented and with no particular constraints, and the challenges that presented themselves were related to computational efforts of creating and keeping in memory large arrays of random numbers for the Wiener process. Downsides to our chosen method of storing the Wiener process

as arrays in files were that the number of time steps, or in other words the integration time, had to be specified before computing the Wiener process. The partitions we opted for, running 5000 particles in 20 runs, gave affordable computations for sufficiently many particles for our application. The scheme gave an anticipated order of convergence of 1 in a weak sense, relative high-resolution solutions, and a strong order of convergence of 1 due to the fact that the method is equivalent to the higher-order Milstein scheme for the constant diffusion we implemented.

To compare the two methods, we applied a kernel density estimator from `KDEpy` as, e.g., Figure 3.4.3 shows, turning the sample-based particle method solution into a discrete density function. By implementing the KDE on the same grid points as the Eulerian grid, the comparison through the integrated squared error was made. Through minimizing the ISE, we found the optimal bandwidth for systems with diffusion  $D_0$  at different integration times, and for a range of Lagrangian particles, and the results were presented in Figures 4.2.1-4.2.4. We found an optimal bandwidth in all except three cases, where a relatively high diffusion and too few particles resulted in an optimal bandwidth set as high as 1 (highest tested bandwidth), which is half the length of the domain, deeming it inaccurate. This low number of particles was neglected in further plots. Generally, we found that the optimal bandwidth has an increasing trend with both diffusion and time, as shown in Figures 4.2.5 and 4.2.6. The latter was questioned as to whether it shows signs of the periodicity of the flow field affecting the bandwidth, but cannot be confirmed with the amount of data points presented. However, it does present an interesting feature in the calculation of the optimal bandwidth, where it potentially is linked with the period and maybe even the perturbations ( $\epsilon$ ) of the flow field. The optimal bandwidth decreases with number of particles, and we used curve fitting to estimate the rate  $\Delta_{optimal} \sim \frac{1}{N^b}$ , for which we found the factor  $b \approx 0.133 \pm 0.070$ . It was estimated that the optimal bandwidth falls quicker with the number of particles for higher diffusion than for lower diffusion, as presented in Table 4.2.1. In addition to emphasizing the need for larger bandwidths when handling fewer particles, the results confirm that when the average distance between the particles increases, the bandwidth must also increase to maintain a smooth distribution. To acquire more data points, the optimal bandwidth could have been calculated more frequently throughout the integration, and also for a longer integration time to achieve better estimations of  $b$ , and also further investigate the bandwidth as a function of time as discussed above.

A lot of the interesting advective behavior in the flow was smeared out due to the harsh constraints in the direct fluid method, which forced us to apply higher diffusion than what we had intentions of. The distribution from the Eulerian method was plotted for all diffusion constants, in Figures D.2a-D.3b, demonstrating how the diffusion is erasing any distinct features during the integration time, and only for the two systems with lowest diffusivity we can still make out advection patterns after a time  $T = 10$ . Another concern is the periodic boundary conditions, which unintentionally were causing the spread of the distribution to become wider than needed be. By not offering any boundaries that forced the mass to stay within the domain walls, the distribution was not reflected back nor pushed together to create a slimmer shape. The distribution was allowed to flow freely over to the other side of the periodic boundary. If the distribution could have been contained by the domain boundaries, we could potentially have had slightly more informative results even with as high diffusion as we have looked at. It would require reflections implemented in the kernel density estimator as well. The initial condition of the distribution affected the evolution in time, as the simulations were initiated with a symmetric Gaussian function centered in the domain, with a standard deviation of 0.1. Other less spread out distributions, such as, e.g., a single point, or a uniform distribution within a square or circle, would be less diffused from the start of the integration, and thus the simulations would for a longer time have stronger advective features.

### 5.1.1 Future Work

Since much of the interesting advection-patterns are hidden in the solutions presented due to relatively large diffusion coefficients, one of the main suggestions for future work would be to solve advection-dominated systems for longer times. In essence, this means turning the diffusion down drastically, an alteration that must be accompanied by an appropriately fine spatial grid. As

mentioned in the discussions above, the boundary conditions applied in this thesis might not be the most relevant conditions for a gyre system in a small domain. Alternatively to periodic spatial boundary conditions, no-flux boundary conditions can be applied, or alternatively look at larger systems where the boundaries have less impact on the solution. Additionally, further investigating how the initial distribution affects the solutions can be worth studying. Finally, the integration time can be increased to look at how the system evolves further, and get a better indication as to how the optimal bandwidth behaves as a function of time. Due to the already large scope of this thesis, we have chosen not to look further into the alterations and suggestions above at this moment. Also, we do not have the computational effort and time required given the requirements for the high resolution that follows an advection-dominated problem; thus we have chosen not to proceed with further work at this moment.

## 5.2 CONCLUSION

In this thesis, we have investigated the Lagrangian particle method and how it compares to the traditional Eulerian fluid method for solving advection-diffusion problems, particularly in a double gyre flow with constant diffusion. We conclude that for this system, the two methods give the same solution for the analytical velocity field in two dimensions on a periodic domain. The numerical schemes applied both have their advantages and disadvantages. The fluid method was confirmed unstable for advection-dominated systems and found to require great computational efforts through using fine enough spatial discretization needed to overcome oscillations. However, it does conveniently return the solution density distribution on an Eulerian grid, and requires few implementation steps outside of the matrix implementation of the Crank-Nicolson finite difference scheme, due to easily accessible python libraries.

The particle method does not require the same large computations, but did demand large sets of Wiener processes that can be computationally heavy to produce, and in our case, store. The method is still considered to be intuitive and easily implemented. The Lagrangian particle solutions are considered samples from a probability density distribution, which has been estimated in this thesis using kernel density estimation, which is one method for presenting the solution as a concentration field rather than samples from one.

We have optimized the estimations by investigating an optimal kernel bandwidth with regards to the integrated squared error relative to a high-resolution Eulerian solution. The optimal kernel bandwidth was found to decrease with the number of Lagrangian particles, and the relationship was estimated using the function form  $\Delta_{optimal} \sim 1/N^b$ . The parameter  $b$  is estimated to be  $0.133 \pm 0.070$  throughout the integration time of  $T = 10$ , which also is the period of the time-varying velocity field. The optimal bandwidth was found to have a clear increasing trend with increasing diffusivity in the simulation system. Along with the need for larger bandwidths for fewer particles, this confirms that when the average distance between particles increases in general, the bandwidth must increase to maintain the construction of a smooth distribution. The bandwidth was finally found to have an ambiguous relation relative to the time, and we have suggested that it might be related to the periodicity of the velocity field. Further work will include lowering the diffusion to investigate the optimal bandwidth in a strongly advective system for longer integration times, and looking at potentially larger domains, and applying different boundary conditions, to not influence the solution in the degree presented in this thesis work.

# Bibliography

- R. A. Adams. *Calculus 1 : Selected chapters from : Calculus: A complete course*, eight edition, 2013.
- A. Atangana. *Groundwater Pollution*, chapter 3, pages 49–72. Academic Press, London, England, 2018. ISBN 0-12-809796-5.
- R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. v. d. Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*. 1994.
- R. B. Bird. *Transport phenomena*. Wiley, New York, 2nd ed. edition, 2002. ISBN 0471410772.
- C. Blomberg. Part v: Stochastic dynamics. In *Physics of Life: The Physicist's Road to Biology*. Elsevier Science & Technology, Oxford, 2007. ISBN 9780444527981.
- Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel density estimation via diffusion. *The Annals of statistics*, 38(5):2916–2957, 2010. ISSN 0090-5364.
- P. Brémaud. *Probability theory and stochastic processes*. Universitext. Springer, Cham, Switzerland, 1st ed. 2020. edition, 2020. ISBN 3-030-40183-9.
- H. Chiri, A. J. Abascal, and S. Castanedo. Deep oil spill hazard assessment based on spatio-temporal met-ocean patterns. *Marine Pollution Bulletin*, 154:111123, 2020. ISSN 0025-326X. doi: <https://doi.org/10.1016/j.marpolbul.2020.111123>. URL <http://www.sciencedirect.com/science/article/pii/S0025326X20302411>.
- J. H. Christensen. The danish eulerian hemispheric model — a three-dimensional air pollution model used for the arctic. *Atmospheric Environment*, 31(24):4169–4191, 1997. ISSN 1352-2310. doi: [https://doi.org/10.1016/S1352-2310\(97\)00264-1](https://doi.org/10.1016/S1352-2310(97)00264-1). URL <https://www.sciencedirect.com/science/article/pii/S1352231097002641>.
- G. Csanady. *Turbulent diffusion in the environment*. D. Reidel Publishing Company, Dordrecht, Holland, 1973.
- M. De Dominicis, N. Pinardi, G. Zodiatis, and R. Lardner. Medslik-ii, a lagrangian marine surface oil spill model for short-term forecasting – part 1: Theory. *Geoscientific Model Development*, 6, 11 2013. doi: 10.5194/gmd-6-1851-2013.
- J. Detemple, R. Garcia, and M. Rindisbacher. Asymptotic properties of monte carlo estimators of diffusion processes. *Journal of Econometrics*, 124:1–68, 2006.
- S. Fortmann-Roe, R. Starfield, and W. M. Getz. Contingent kernel density estimation. *PloS one*, 7(2):e30549, 2012. ISSN 1932-6203.
- T. D. Frank. *Linear and Nonlinear Fokker-Planck Equations*, pages 149–182. Springer US, New York, NY, 2020. ISBN 978-1-0716-0421-2. doi: 10.1007/978-1-0716-0421-2\_311. URL [https://doi.org/10.1007/978-1-0716-0421-2\\_311](https://doi.org/10.1007/978-1-0716-0421-2_311).

- D. P. French-McCay. Oil spill impact modeling: Development and validation. *Environmental Toxicology and Chemistry*, 23(10):2441–2456, 2004. doi: <https://doi.org/10.1897/03-382>. URL <https://setac.onlinelibrary.wiley.com/doi/abs/10.1897/03-382>.
- P. Frey. Chapter 3: The finite difference method, 2017. URL [https://www.ljll.math.upmc.fr/frey/cours/UdC/ma691/ma691\\_ch6.pdf](https://www.ljll.math.upmc.fr/frey/cours/UdC/ma691/ma691_ch6.pdf). Course Note for MA691.
- I. I. Gichman. *Stochastic differential equations*, volume 72 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, Berlin, 1972.
- A. Gramacki. *Nonparametric kernel density estimation and its computational aspects*, volume Volume 37 of *Studies in big data* ;. Springer, Cham, Switzerland, 1st ed. 2018. edition, 2018. ISBN 3-319-71688-3.
- U. Gräwe, E. Deleersnijder, S. H. A. M. Shah, and A. Heemink. Why the euler scheme in particle tracking is not enough: the shallow-sea pycnocline test case. *Ocean Dynamics*, 62:501–514, 2012.
- U. Gräwe. Implementation of high-order particle-tracking schemes in a water column model. *Ocean Modelling*, 36(1):80 – 89, 2011. ISSN 1463-5003. doi: <https://doi.org/10.1016/j.ocemod.2010.10.002>. URL <http://www.sciencedirect.com/science/article/pii/S1463500310001484>.
- C. Gualtieri, B. Cushman-Roisin, and D. Mihailovic. *Environmental Fluid Mechanics: Current issues and future outlook*, pages 3–17. CRC Press / Balkema, 11 2012. ISBN 978-0-415-62156-4.
- H. Haken. *Synergetics: Introduction and Advanced Topics*. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2004. ISBN 9783540408246.
- G. Haller. Lagrangian coherent structures. *Annual Review of Fluid Mechanics*, 47(1): 137–162, 2015. doi: 10.1146/annurev-fluid-010313-141322. URL <https://doi.org/10.1146/annurev-fluid-010313-141322>.
- G. Haller and G.-C. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D: Nonlinear Phenomena*, 147:352–370, 12 2000. doi: 10.1016/S0167-2789(00)00142-1.
- W. Hundsdorfer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, volume 33 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg : Imprint: Springer, Berlin, Heidelberg, 1st ed. 2003. edition, 2003. ISBN 3-662-09017-1.
- K. Ivanova, M. Ausloos, and H. Takayasu. Deterministic and stochastic influences on japan and us stock and foreign exchange markets. a fokker-planck approach. *arXiv.org, Quantitative Finance Papers*, 01 2003. doi: 10.1007/978-4-431-53947-6\_22.
- L. P. Kadanoff. *Statistical physics : statics, dynamics and renormalization*. World Scientific, Singapore, 2000. ISBN 9810237588.
- H. Kile. Bandwidth selection in kernel density estimation. Master’s thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2010.
- P. E. Kloeden. *Numerical solution of stochastic differential equations*, volume 23 of *Applications of Mathematics, Stochastic Modelling and Applied Probability*. Springer-Verlag, Berlin, Germany, 1st ed. 1992. edition, 1992. ISBN 3-662-12616-8.
- Y. Komori, Y. Saito, and T. Mitsui. Some issues in discrete approximate solution for stochastic differential equations. *Computers & Mathematics with Applications*, 28(10):269 – 278, 1994. ISSN 0898-1221. doi: [https://doi.org/10.1016/0898-1221\(94\)00197-9](https://doi.org/10.1016/0898-1221(94)00197-9). URL <http://www.sciencedirect.com/science/article/pii/0898122194001979>.
- D. P. Kroese, T. Brereton, T. Taimre, and Z. I. Botev. Why the monte carlo method is so important today: Why the mcm is so important today. *Wiley interdisciplinary reviews. Computational statistics*, 6(6):386–392, 2014. ISSN 1939-5108.

- D. P. Landau. *A guide to Monte Carlo simulations in statistical physics*. Cambridge University Press, Cambridge, 3rd ed. edition, 2009. ISBN 9780521768481.
- L. Lebreton, S. Greer, and J. Borrero. Numerical modelling of floating debris in the world's oceans. *Marine pollution bulletin*, 64:653–61, 03 2012. doi: 10.1016/j.marpolbul.2011.10.027.
- P. L'Ecuyer. Efficiency improvement and variance reduction. In *Proceedings of the 26th Conference on Winter Simulation, WSC '94*, page 122–132, San Diego, CA, USA, 1994. Society for Computer Simulation International. ISBN 078032109X.
- B. C. Levy. *Random processes with applications to circuits and communications*. Springer, Cham, Switzerland, 1st ed. 2020. edition, 2020. ISBN 3-030-22297-7.
- D. Lynch, D. Greenberg, A. Bilgili, D. McGillicuddy, J. Manning, and A. Aretxabaleta. *Particles in the Coastal Ocean: Theory and Applications*. 11 2014. ISBN 9781107061750. doi: 10.1017/CBO9781107449336.
- N. Maximenko, J. Hafner, and P. Niiler. Pathways of marine debris derived from trajectories of lagrangian drifters. *Marine Pollution Bulletin*, 65(1):51 – 62, 2012. ISSN 0025-326X. doi: <https://doi.org/10.1016/j.marpolbul.2011.04.016>. URL <http://www.sciencedirect.com/science/article/pii/S0025326X11002189>. At-sea Detection of Derelict Fishing Gear.
- H. E. M. Meier and A. Höglund. *Studying the Baltic Sea Circulation with Eulerian Tracers*, pages 101–129. Springer International Publishing, Heidelberg, 2013. ISBN 978-3-319-00440-2. doi: 10.1007/978-3-319-00440-2\_4. URL [https://doi.org/10.1007/978-3-319-00440-2\\_4](https://doi.org/10.1007/978-3-319-00440-2_4).
- A. S. Mountford and M. A. Morales Maqueda. Eulerian modeling of the three-dimensional distribution of seven popular microplastic types in the global ocean. *Journal of Geophysical Research: Oceans*, 124(12):8558–8573, 2019. doi: <https://doi.org/10.1029/2019JC015050>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019JC015050>.
- V. Onink, D. Wichmann, P. Delandmeter, and E. van Sebille. The role of ekman currents, geostrophy, and stokes drift in the accumulation of floating microplastic. *Journal of Geophysical Research: Oceans*, 124(3):1474–1490, 2019. doi: <https://doi.org/10.1029/2018JC014547>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JC014547>.
- T. Peacock and G. Haller. Lagrangian coherent structures: The hidden skeleton of fluid flows. *Physics Today*, 66:41–47, 2013.
- N. C. Petroni. Dynamical theory of brownian motion. In *Probability and Stochastic Processes for Physicists*, UNITEXT for Physics, pages 259–275. Springer International Publishing, 2020. ISBN 9783030484071.
- M. A. Pinsky. *An introduction to stochastic modeling*. Academic Press, Amsterdam, 4th ed. edition, 2011. ISBN 978-0-12-381416-6.
- E. Platen. *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*, volume 64 of *Stochastic Modelling and Applied Probability*. Springer Berlin Heidelberg : Imprint: Springer, Berlin, Heidelberg, 1st ed. edition, 2010. ISBN 3-642-13694-X.
- S. V. Prants, M. Y. Uleysky, and M. V. Budyansky. *Lagrangian Tools to Study Transport and Mixing in the Ocean*, pages 95–115. Springer International Publishing, Cham, 2017. ISBN 978-3-319-53022-2. doi: 10.1007/978-3-319-53022-2\_4. URL [https://doi.org/10.1007/978-3-319-53022-2\\_4](https://doi.org/10.1007/978-3-319-53022-2_4).
- K. Pratt, J. Meiss, and J. Crimaldi. Reaction enhancement of initially distant scalars by lagrangian coherent structures. *Physics of Fluids*, 27:035106, 03 2015. doi: 10.1063/1.4914467.
- H. Risken. *The Fokker-Planck Equation: Methods of Solution and Applications*, volume 18 of

- Springer Series in Synergetics*. Springer Berlin Heidelberg, Berlin, Heidelberg, second edition. edition, 1996. ISBN 9783540615309.
- H. Risken, C. Schmid, and W. Weidlich. Fokker-planck equation for atoms and light mode in a laser model with quantum mechanically determined dissipation and fluctuation coefficients. *Physics Letters*, 20(5):489 – 491, 1966. ISSN 0031-9163. doi: [https://doi.org/10.1016/0031-9163\(66\)90967-X](https://doi.org/10.1016/0031-9163(66)90967-X). URL <http://www.sciencedirect.com/science/article/pii/003191636690967X>.
- T. Sauer. *Numerical Analysis*. Pearson Education Limited, Harlow, Essex, England, second edition edition, 2014. ISBN 9781292023588.
- S. S. Sawilowsky. You think you’ve got trivials? *Journal of Modern Applied Statistical Methods*, 2(1):218–225, 2003. Article 21.
- D. W. Scott. Multivariate density estimation and visualization. In J. E. Gentle, W. K. Härdle, and Y. Mori, editors, *Handbook of Computational Statistics: Concepts and Methods*, Springer Handbooks of Computational Statistics, chapter 19, pages 549–569. Springer-Verlag, Berlin, Heidelberg, 2 edition, 2012. ISBN 9783642215506.
- W. D. Scott. *Multivariate Density Estimation. Theory, Practice, and Visualization*. John Wiley & Sons, Inc., 1992. ISBN 9780470316849. doi: 10.1002/9780470316849. Wiley Series in Probability and Statistics.
- E. Seville, P. Scussolini, J. Durgadoo, F. Peeters, A. Biastoch, W. Weijer, C. Turney, and R. Zahn. Ocean currents generate large footprints in marine palaeoclimate proxies. *Nature Communications*, 6, 03 2015. doi: 10.1038/ncomms7521.
- S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica. D*, 212(3):271–304, 2005. ISSN 0167-2789.
- S. Sheather and M. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B. Methodological*, 53:683–690, 01 1991. doi: 10.2307/2345597.
- B. Silverman. *Density estimation for statistics and data analysis*. Monographs on statistics and applied probability. Chapman and Hall, London, 1986. ISBN 0412246201.
- J. S. Simonoff. *Smoothing Methods in Statistics*. Springer Series in Statistics. Springer New York, New York, 1996. ISBN 1461284724.
- S. A. Socolofsky and G. H. Jirka. Environmental fluid mechanics part i: Mass transfer and diffusion, 2002. Engineering – Lectures.
- M. Spivak. *Calculus*. Publish or Perish, Houston, Tex, 3rd ed. edition, 1994. ISBN 0914098896.
- R. H. Stewart. *Introduction to Physical Oceanography*. Texas A&M University, 2008.
- A. Stohl, S. Eckhardt, C. Forster, P. James, and N. Spichtinger. On the pathways and timescales of intercontinental air pollution transport. *Journal of Geophysical Research*, 107, 12 2002. doi: 10.1029/2001JD001396.
- M. Tatari, M. Dehghan, and M. Razzaghi. Application of the adomian decomposition method for the fokker–planck equation. *Mathematical and Computer Modelling*, 45(5):639 – 650, 2007. ISSN 0895-7177. doi: <https://doi.org/10.1016/j.mcm.2006.07.010>. URL <http://www.sciencedirect.com/science/article/pii/S0895717706002883>.
- D. J. Thomson. Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *Journal of Fluid Mechanics*, 180:529–556, 1987. doi: 10.1017/S0022112087001940.

- N. G. van Kampen. *Stochastic processes in physics and chemistry*. North-Holland personal library. Elsevier, Amsterdam ;, 3rd ed. edition, 2007. ISBN 1-281-00388-3.
- E. van Sebille, S. M. Griffies, R. Abernathey, T. P. Adams, P. Berloff, A. Biastoch, B. Blanke, E. P. Chassignet, Y. Cheng, C. J. Cotter, E. Deleersnijder, K. Döös, H. F. Drake, S. Drijfhout, S. F. Gary, A. W. Heemink, J. Kjellsson, I. M. Koszalka, M. Lange, C. Lique, G. A. MacGilchrist, R. Marsh, C. G. M. Adame, R. McAdam, F. Nencioli, C. B. Paris, M. D. Piggott, J. A. Polton, S. Rühls, S. H. Shah, M. D. Thomas, J. Wang, P. J. Wolfram, L. Zanna, and J. D. Zika. Lagrangian ocean analysis: Fundamentals and practices. *Ocean Modelling*, 121: 49 – 75, 2018. ISSN 1463-5003. doi: <https://doi.org/10.1016/j.ocemod.2017.11.008>. URL <http://www.sciencedirect.com/science/article/pii/S1463500317301853>.
- E. van Sebille, S. M. Griffies, R. Abernathey, T. P. Adams, P. Berloff, A. Biastoch, B. Blanke, E. P. Chassignet, Y. Cheng, C. J. Cotter, E. Deleersnijder, K. Döös, H. F. Drake, S. Drijfhout, S. F. Gary, A. W. Heemink, J. Kjellsson, I. M. Koszalka, M. Lange, C. Lique, G. A. MacGilchrist, R. Marsh, C. G. Mayorga Adame, R. McAdam, F. Nencioli, C. B. Paris, M. D. Piggott, J. A. Polton, S. Rühls, S. H. Shah, M. D. Thomas, J. Wang, P. J. Wolfram, L. Zanna, and J. D. Zika. Lagrangian ocean analysis: Fundamentals and practices. *Ocean Modelling*, 121:49–75, 2018. ISSN 1463-5003. doi: <https://doi.org/10.1016/j.ocemod.2017.11.008>. URL <https://www.sciencedirect.com/science/article/pii/S1463500317301853>.
- E. Vanden-Eijnden. Lecture 6: Wiener process, 2006. URL <https://www.whoj.edu/fileserver.do?id=21268&pt=10&p=17232>. Accessed: 2021-02-22.
- D. Wichmann, P. Delandmeter, and E. van Sebille. Influence of near-surface currents on the global dispersal of marine microplastic. *Journal of Geophysical Research: Oceans*, 124(8):6086–6096, 2019. doi: <https://doi.org/10.1029/2019JC015328>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019JC015328>.
- J. Wilson. Variance reduction techniques for digital simulation. *American Journal of Mathematical and Management Sciences*, 4:277–312, 02 1984. doi: 10.1080/01966324.1984.10737147.
- R. Zwanzig. *Nonequilibrium statistical mechanics*. Oxford Unisersity Press, 2001.
- B. Øksendal. *Stochastic Differential Equations : An Introduction with Applications*. Universitext. Springer Berlin Heidelberg : Imprint: Springer, Berlin, Heidelberg, sixth edition. edition, 2003. ISBN 3-642-14394-6.

# Appendix

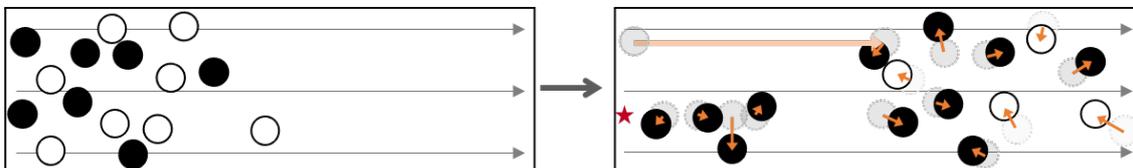
## APPENDIX

### A COMPLIMENTARY THEORY AND DERIVATIONS

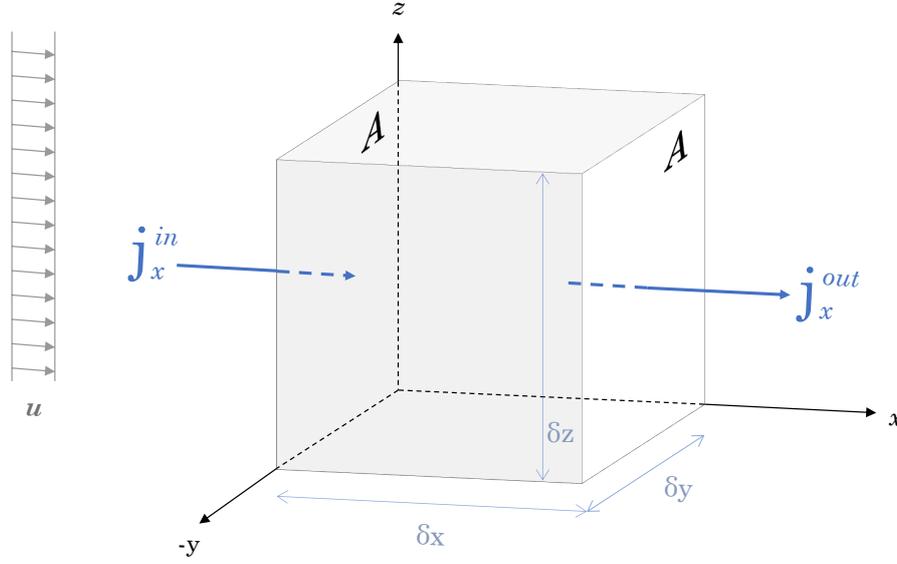
#### A.1 Derivation of the Advection-Diffusion Equation

The fluid mechanical processes investigated in this thesis can be classified into one of two main categories: transport and transformation. Transport processes in nature are when quantities are physically moved, changing of the spatial location, in the system. Transformations on the other hand, changes the composition of the system. In this section we derive the advection and diffusion equations separately in one dimension, before returning to the unified advection-diffusion equation, and finally, discuss how transformation can be incorporated into the transport equation.

Say you have a closed region with two different types of molecules present in a fluid. We can imagine the system as a box with black and white spheres representing the two different molecules, as in Figure A.1. A mean fluid flow is present (gray arrows), transporting the molecules to the right as time passes. They are also transported by random motion moving the spheres (orange arrows). At the same time transformation processes might occur; The molecules can react with other substances in the fluid, or be added to (or leaked from) the system through some source (or sink). Figure A.1 shows how the fluid mechanical processes of transport and transformation changes the composition of the mixture as well as the distribution of each species. Sources (or sinks) within the region could add (or subtract) from the concentrations. This is however out of the scope of this thesis.



**Figure A.1:** After a time  $\Delta t$  the two types of molecules have undergone the processes of transport and transformation. White molecules have reacted with substances in the fluid, and has a lower concentration after  $\Delta t$ , while there is a source increasing the concentration of black molecules, represented by a red star. A bold orange arrow shows the transport for one of the molecules produced by of the mean fluid flow, that is common for all the particles.



**Figure A.2:** Schematic of a control volume with a crossflow  $\mathbf{u}$ , inspired by Figure 2.1 in Socolofsky and Jirka (2002).

### The Advection Equation

We begin by considering the infinitesimal control volume presented in Figure A.2, containing some chemical or biological species. Let  $C(x, t)$  describe the concentration of the species, that only changes in the  $x$ -direction. The species is carried along with some flow  $\mathbf{u} = u\hat{\mathbf{x}}$ . Consider the average concentration  $\bar{C}(x, t)$  inside the control volume which in the  $x$ -direction is limited by  $[x - \frac{1}{2}\delta x, x + \frac{1}{2}\delta x]$  for a small number  $\delta x$ . Considering the conservation of mass of the species in this conserved system, the change of  $\bar{C}(x, t)$  per unit of time must be the net balance of inflow and outflow over the boundaries,

$$\frac{\partial}{\partial t}\bar{C}(x, t) = \frac{1}{\delta x} \left[ u(x - \frac{1}{2}\delta x, t)C(x - \frac{1}{2}\delta x) - u(x + \frac{1}{2}\delta x, t)C(x + \frac{1}{2}\delta x) \right]. \quad (\text{A.1})$$

Now, if we let  $\delta x \rightarrow 0$ , it follows that the concentration satisfies

$$\frac{\partial}{\partial t}C(x, t) + \frac{\partial}{\partial x}(u(x, t)C(x, t)) = 0, \quad (\text{A.2})$$

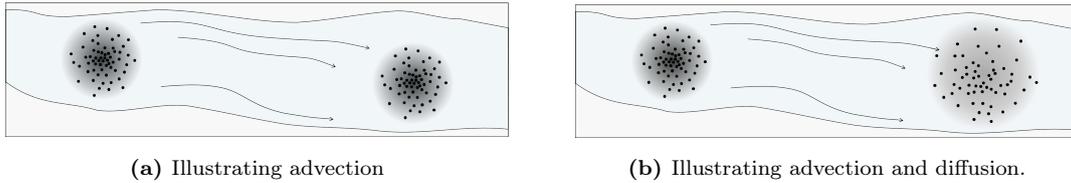
which is the advection (or convection) equation.

### The Diffusion Equation

In a similar way to deriving the advective flux, we can consider the transport via diffusion. We still consider the control volume shown in Figure A.2. The change in  $\bar{C}(x, t)$  is caused by gradients in the concentration field. Again considering the net balance of inflow and outflow at the two boundaries  $(x - \frac{1}{2}\delta x)$  and  $(x + \frac{1}{2}\delta x)$ , and following Fick's first law (2.1.3) of diffusive flux, we find

$$\frac{\partial}{\partial t}\bar{C}(x, t) = \frac{1}{\delta x} \left[ D(x - \frac{1}{2}\delta x, t) \frac{\partial}{\partial x}C(x - \frac{1}{2}\delta x, t) - D(x + \frac{1}{2}\delta x, t) \frac{\partial}{\partial x}C(x + \frac{1}{2}\delta x, t) \right], \quad (\text{A.3})$$

where  $D(x, t)$  is the diffusion coefficient. Again we let  $\delta x \rightarrow 0$ , ending up with the diffusion equation,



(a) Illustrating advection

(b) Illustrating advection and diffusion.

**Figure A.3:** The center of mass is transported downstreams for the purely advective system, while the diffusive system additionally shows a spread in particles.

$$\frac{\partial}{\partial t}C(x, t) = \frac{\partial}{\partial x}\left(D(x, t)\frac{\partial}{\partial x}C(x, t)\right), \quad (\text{A.4})$$

describing the change in concentration due to diffusive transport.

### The Advection-Diffusion Equation

The derivation of the Advection-Diffusion Equation (ADE) relies on the principle of superposition: advection and diffusion can be added together if they are linearly independent. We assume that they are; the presence of a mean flow does not bias the parallel diffusion process (Socolofsky and Jirka, 2002). Thus, we get the one-dimensional form of the advection-diffusion equation

$$\frac{\partial}{\partial t}C(x, t) + \frac{\partial}{\partial x}(u(x, t)C(x, t)) = \frac{\partial}{\partial x}\left(D(x, t)\frac{\partial}{\partial x}C(x, t)\right), \quad (\text{A.5})$$

that describes the physical phenomena where particles, energy, or other physical quantities are transferred inside a physical system due to both advection and diffusion. Depending on the context, the equation is referred to as the convection–diffusion equation, drift-diffusion equation, or (generic) scalar transport equation (Atangana, 2018).

In a similar way, the 3-D advection-diffusion can be developed as:

$$\frac{\partial}{\partial t}C(\mathbf{x}, t) + \nabla \cdot (\mathbf{u}(\mathbf{x}, t)C(\mathbf{x}, t)) = \nabla \cdot (D(\mathbf{x}, t)\nabla C(\mathbf{x}, t)) \quad (\text{A.6})$$

where we do not assume that the diffusion is constant in time or space, and  $\mathbf{u} = [u, v, w]$  is the three-dimensional velocity vector field.

### Incorporating transformation with the transport equation

Transformation is the "production (or loss) of a given species of interest through physical, chemical, or biological processes" (Socolofsky and Jirka, 2002). This means that the mass of the species is no longer conserved in a system with transformation processes. Examples of physical, chemical, and biological process are radioactive decay, dissolution of carbon dioxide in water, and photosynthesis. The processes transform the composition, while keeping the total mass of the system conserved.

To incorporate transformation with the transport equation (A.5), we must find an expression that describes the change in concentration of a given species under a transformation process. A system with transformation processes, can also be called a *reactive system*, and can hence be described by the characteristic reaction rate  $k$  of the transformation reaction. In a general  $n$ -th order reaction, the change in a species  $q$  is given by  $\partial_t q = \pm kq^n$ . A reaction term can be added directly to the advection-diffusion equation as

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}C) = \nabla \cdot (D\nabla C) + R, \quad (\text{A.7})$$

where  $R$  can describe multiple reactions of different orders. The above equation is referred to as the advection-diffusion-reaction equation, but we will in the following not concern ourselves with reactive systems and stick to only the transportation aspects of fluid mechanical processes.

## A.2 Finite Difference Methods

Partial differential equations (PDEs), such as the advection-diffusion equation (A.5), are differential equations with more than one independent variable, where the solution is a function or a vector of functions. Since PDEs typically have infinitely many solutions, it is necessary to specify both the domain and boundary conditions to achieve a particular solution. Even so, most PDEs do not have analytical solutions, and the solution can only be found through approximations. A numerical technique for solving differential equations is the *Finite Difference Method*. The derivatives are approximated through *finite differences*, which essentially are the differences obtained by incrementing successively the dependent variable of a function by a fixed amount, e.g.  $f(x + \Delta x) - f(x)$ . There are different types of finite differences, and thus also finite difference schemes applicable for solving PDEs. In this thesis we are particularly interested in the Crank-Nicolson implicit scheme, which is used to numerically solve the advection-diffusion equation. In the following we will classify finite differences using Taylor's theorem, before we in the next section will demonstrate how the Crank-Nicolson scheme can be applied to the advection-diffusion equation in one dimension.

### Taylor's Formula

The finite difference approximation of derivatives can be derived from the Taylor's formula by Taylor's Theorem (see Theorem 4 in Spivak (1994, p. 417), given in Appendix A.4). Taylor's formula with Lagrange remainder (e.g., Adams (2013, p. 275)) is

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}(\Delta x)^2 + \dots + \frac{f^{(n)}(x)}{n!}(\Delta x)^n + R_n(x + \Delta x), \quad (\text{A.8})$$

where  $f$  is some function which has derivatives up to order  $(n + 1)$  in an interval containing  $x$  and  $x + \Delta x$ , and a remainder term that satisfy  $\lim_{n \rightarrow \infty} R_n = 0$ . Truncating the series after  $k$  terms, results in an approximating polynomial  $P_k(x + \Delta x) \approx f(x + \Delta x)$  with a remainder term  $R_k$ . The error denotes the difference from the approximating polynomial of degree  $k$ ,

$$P_k(x + \Delta x) = f(x) + \frac{f'(x)}{1!}(\Delta x) + \dots + \frac{f^{(k)}(x)}{k!}(\Delta x)^k + R_k(x_1) \quad (\text{A.9})$$

and the original function  $f$ . The remainder is given by

$$R_n(x + \Delta x) = \frac{f^{(n+1)}(x_1)}{(n + 1)!}(\Delta x)^{n+1}, \quad (\text{A.10})$$

for  $x_1 \in [x, x + \Delta x]$ . The Taylor polynomials can be used to approximate derivatives, and we will show this for the lowest derivative first, by truncating the series after the second term:

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{\Delta}x + R_1(x + \Delta x), \quad (\text{A.11})$$

where the remainder term is  $R_1(x_1) \sim (\Delta x)^2$ . The expression can be solved for the derivative  $f'$ ,

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} - \frac{R_1(x_1)}{\Delta x}, \quad (\text{A.12})$$

and given in terms of the big-O-notation,

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + \mathcal{O}(\Delta x), \quad (\text{A.13})$$

where  $\mathcal{O}(\Delta x)$  indicates that the truncation error is proportional to  $\Delta x$ . The approximation is hence of order 1, according to the following definition (Definition 6.1 from Frey (2017, p. 80)):

**Definition 1** *The approximation of the derivative  $f'$  as a point  $x$  is of order  $p$  ( $p > 0$ ) if there exists a constant  $C > 0$ , independent of  $h$ , such that the error between the derivative and its approximation is bounded by  $C(\Delta x)^p$  (i.e. is exactly  $\mathcal{O}((\Delta x)^p)$ ).*

### Types of finite differences

Consider the function  $g = g(x, t)$ , discretized as  $g_i^n = g(i\Delta x, n\Delta t)$ , over a uniform grid

$$t_n = n\Delta t, \quad \text{and} \quad x_i = i\Delta x.$$

The truncations to the Taylor polynomial that we looked at above, involves approximating the derivative of a function by using the function itself evaluated at neighbouring points on the grid. Depending on at which neighbouring points we evaluate the function, the order of the approximations differ.

Using the approximation (A.13) of the first spatial derivative gives

$$\frac{\partial g}{\partial x}(x_i, t_n) \cong \frac{g_{i+1}^n - g_i^n}{\Delta x}, \quad (\text{A.14})$$

which is called the *forward difference* approximation. Another order 1 approximation is the *backward difference*, given as

$$\frac{\partial g}{\partial x}(x_i, t_n) \cong \frac{g_i^n - g_{i-1}^n}{\Delta x}. \quad (\text{A.15})$$

The first derivative  $g'$  can also be approximated through a *central difference* by combining the two expressions given above. Adding the expressions for the derivative,

$$2\frac{\partial g}{\partial x}(x_i, t_n) \cong \frac{g_{i+1}^n - g_i^n}{\Delta x} + \frac{g_i^n - g_{i-1}^n}{\Delta x}, \quad (\text{A.16})$$

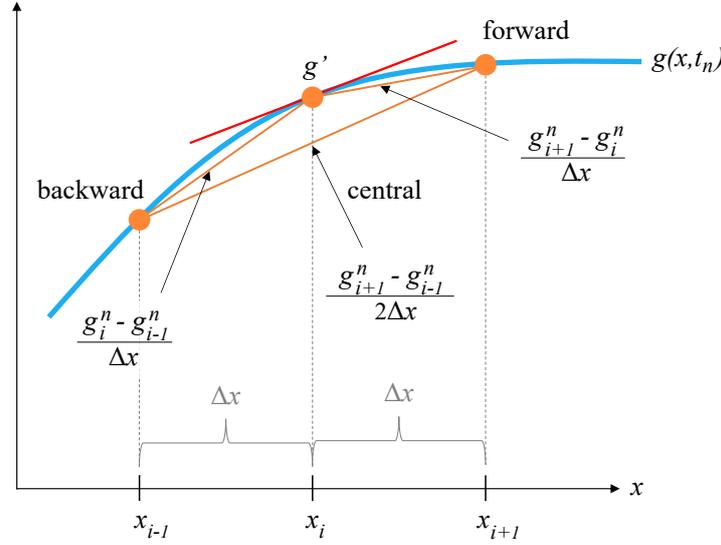
yields the central difference

$$\frac{\partial g}{\partial x}(x_i, t_n) \cong \frac{g_{i+1}^n - g_{i-1}^n}{2\Delta x}, \quad (\text{A.17})$$

which is an order 2 approximation. We find the order by writing the approximations as the truncated polynomials of the Taylor series

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}(\Delta x)^2 + \frac{f^{(3)}(x)}{3!}(\Delta x)^3 \quad (\text{A.18})$$

and



**Figure A.4:** Illustration of the backward, forward and central differences to approximate the first spatial derivative of a function  $g(x, t)$  at the point  $(x_i, t_n)$ . The domain is discretized with  $x_i = i\Delta x$  and  $t_n = n\Delta t$ , yielding the discretized function  $g_i^n = g(i\Delta x, n\Delta t)$ .

$$f(x - \Delta x) = f(x) - \frac{f'(x)}{1!} \Delta x + \frac{f^{(2)}(x)}{2!} (\Delta x)^2 - \frac{f^{(3)}(x)}{3!} (\Delta x)^3 \quad (\text{A.19})$$

for  $x_{1+} \in [x, x + \Delta x]$  and  $x_{1-} \in [x - \Delta x, x]$ . The central difference is

$$\frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} = f'(x) + \frac{f^{(3)}(x)}{3!} (\Delta x)^2 \quad (\text{A.20})$$

for  $x \in [x - \Delta x, x + \Delta x]$  (using the Intermediate Value Theorem, see, e.g., Spivak (1994, p. 120)). Expressing the derivative as in our notation, we get

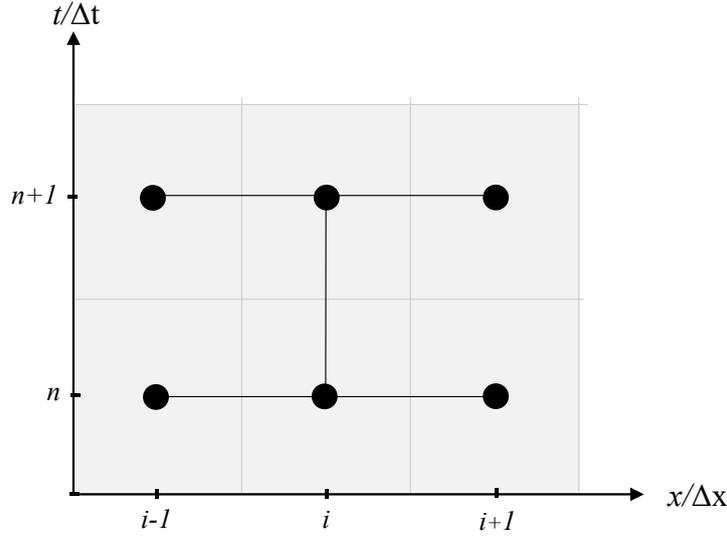
$$f'(x) \cong \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + \mathcal{O}((\Delta x)^2), \quad (\text{A.21})$$

where  $\mathcal{O}((\Delta x)^2)$  is the big-O-notation indicating an error proportional to  $(\Delta x)^2$ .

The three types of finite differences presented are shown in Figure A.4, describing how they differ in approximation of the slope. The methods can be written as linear systems, for each of the three types of finite differences as

$$\mathbf{D} \cdot \begin{bmatrix} g(x_{i-1}, t) \\ g(x_i, t) \\ g(x_{i+1}, t) \end{bmatrix} = g'(x_i, t) \quad \begin{cases} \mathbf{D}_{\text{central}} &= \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \cdot \frac{1}{2\Delta x} \\ \mathbf{D}_{\text{backward}} &= \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} \cdot \frac{1}{\Delta x} \\ \mathbf{D}_{\text{forward}} &= \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \cdot \frac{1}{\Delta x} \end{cases} \quad (\text{A.22})$$

When using a finite difference approach to compute the derivative on an entire grid at once, e.g., when solving partial or ordinary differential equations on a defined domain, the equations can



**Figure A.5:** The Crank-Nicolson stencil for a one-dimensional problem.

be written as a matrix system. As an example, the backward difference method can be used to approximate the first derivative of a one-dimensional problem  $g(x_i)$  on a grid of  $N$  points using the following set up

$$\frac{1}{\Delta x} \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & \dots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots & \\ 0 & 0 & \dots & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} g(x_0) \\ g(x_1) \\ g(x_2) \\ g(x_3) \\ \vdots \\ g(x_{N-3}) \\ g(x_{N-2}) \\ g(x_{N-1}) \end{bmatrix} = \begin{bmatrix} g'(x_0) \\ g'(x_1) \\ g'(x_2) \\ g'(x_3) \\ \vdots \\ g'(x_{N-3}) \\ g'(x_{N-2}) \\ g'(x_{N-1}) \end{bmatrix} \quad (\text{A.23})$$

where a forward difference is used at the boundary. This linear representation is very convenient when solving large systems numerically. We will shortly look at a specific numerical scheme used to solve differential equations using finite differences, and how the problem can be written in a matrix form that is solvable with established methods of linear algebra.

The example function was originally a function of both space and time  $g = g(x, t)$ , and the temporal derivative could have just as simply been used as the example. When using finite differences to solve time-dependent problems however, one is introduced to the concept of *explicit* and *implicit methods*. To illustrate, we calculate the temporal derivative of the function  $g(x, t)$  using the backward and the forward difference as

$$\frac{\partial g}{\partial t}(x_i, t_n) \cong \frac{g_i^n - g_i^{n-1}}{\Delta t}, \quad \text{and} \quad \frac{\partial g}{\partial t}(x_i, t_n) \cong \frac{g_i^{n+1} - g_i^n}{\Delta t}. \quad (\text{A.24})$$

The explicit method, the backward difference, uses the state of the system at a previous time  $(n - 1)$ , to calculate the state of the system at a current time  $(n)$ . The implicit method however, requires that one solves an equation for both the current  $(n)$  and the later time  $(n+1)$ , as we see in the forward difference above, requiring extra computations.

### A.3 Crank-Nicolson Scheme

The numerical scheme called *Crank-Nicolson* is an implicit second order method in time, known for its unconditional numerical stability. It is the numerical scheme used for solving the advection-diffusion equation in the Eulerian formulation in this thesis, and for that reason we have given it extra attention in this section, where the scheme is presented and applied to the advection-diffusion equation in one-dimension.

Consider a typical one-dimensional second order partial differential equation

$$\frac{\partial f}{\partial t} = F \left( x, t, f, \frac{\partial f}{\partial x}, \frac{\partial^2 f}{\partial x^2} \right), \quad (\text{A.25})$$

with discretization  $f(i\Delta x, n\Delta t) = f_i^n$ , for which the function  $F = F_i^n$  is evaluated for the values  $i$ ,  $n$ , and  $f_i^n$ . The Crank-Nicolson scheme uses the *forward Euler method* at the time  $n$

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = F_i^n \left( x, t, f, \frac{\partial f}{\partial x}, \frac{\partial^2 f}{\partial x^2} \right) \quad (\text{A.26})$$

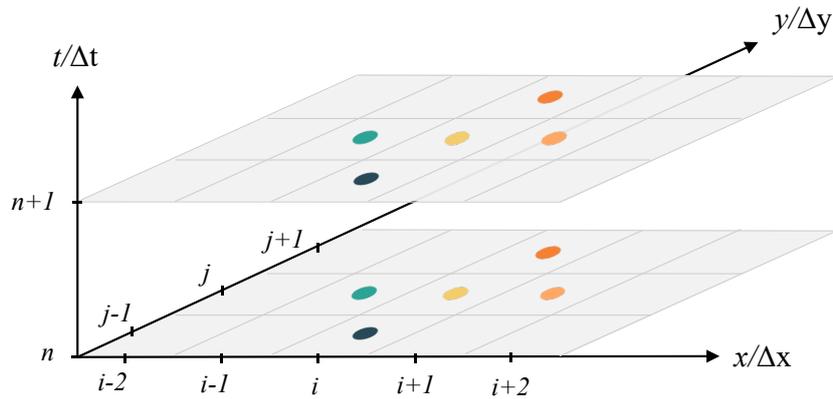
and the *backward Euler method* at  $n + 1$ ,

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = F_i^{n+1} \left( x, t, f, \frac{\partial f}{\partial x}, \frac{\partial^2 f}{\partial x^2} \right). \quad (\text{A.27})$$

Thus, for the second order partial differential equation (A.25), the Crank-Nicolson scheme is

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = \frac{1}{2} \left[ F_i^{n+1} \left( x, t, f, \frac{\partial f}{\partial x}, \frac{\partial^2 f}{\partial x^2} \right) + F_i^n \left( x, t, f, \frac{\partial f}{\partial x}, \frac{\partial^2 f}{\partial x^2} \right) \right]. \quad (\text{A.28})$$

Because the Crank-Nicolson scheme is a combination of an explicit and an implicit method, the scheme results in an implicit method. In one and two dimensions, the stencil expressed by the scheme is seen in Figure A.5 and Figure A.6 respectively.



**Figure A.6:** Illustration of the Crank-Nicolson stencil for a two-dimensional problem.

### Crank-Nicolson scheme for solving the Advection-Diffusion Equation

The first step in applying a finite difference method such as the Crank-Nicolson scheme to the advection-diffusion equation is discretizing the domain, e.g.,  $t \in [0, T]$  and  $x \in [0, x_f]$ , such that

$$\begin{aligned} 0 &= t_0 < t_1 < t_2 < \dots < t_{Nt-2} < t_{Nt-1} = T, \\ 0 &= x_0 < x_1 < x_2 < \dots < x_{Nx-2} < x_{Nx-1} = x_f. \end{aligned}$$

which we express  $x_i = i\Delta x$ , and  $t_n = n\Delta t$ , for equidistant spatial and temporal steps. Next, the PDE itself is discretized. This indicates that the requirement that the equation is fulfilled in the entire domain, is relaxed to apply only at the discrete points of the grid,

$$(2.1.5) \rightarrow \frac{\partial C(i\Delta x, n\Delta t)}{\partial t} + u \frac{\partial C(i\Delta x, n\Delta t)}{\partial x} = D \frac{\partial^2 C(i\Delta x, n\Delta t)}{\partial x^2}.$$

Then the derivatives can be replaced by the finite differences, which for the implicit Crank-Nicolson scheme, means transforming all parts of the PDE using the following

$$\begin{aligned} \frac{\partial C}{\partial t} &\rightarrow \frac{C_i^{n+1} - C_i^n}{\Delta t} \\ \frac{\partial C}{\partial x} &\rightarrow \frac{1}{2} \left( \frac{C_{i+1}^{n+1} - C_{i-1}^{n+1}}{2(\Delta x)} + \frac{C_{i+1}^n - C_{i-1}^n}{2(\Delta x)} \right) \\ \frac{\partial^2 C}{\partial x^2} &\rightarrow \frac{1}{2(\Delta x)^2} ((C_{i+1}^{n+1} - 2C_i^{n+1} + C_{i-1}^{n+1}) + (C_{i+1}^n - 2C_i^n + C_{i-1}^n)). \end{aligned} \tag{A.29}$$

Implementing the above scheme results in the linear system

$$\mathbb{L}\mathbf{C}^{n+1} = \mathbb{R}\mathbf{C}^n, \tag{A.30}$$

where  $\mathbf{C}$  has the form of a column vector, containing the state of the system at all points  $x_i$ , at respective times  $n$  (current) and  $n + 1$  (future). The system (A.30) can be solved through direct matrix inversion

$$\mathbf{C}^{n+1} = \mathbb{L}^{-1}\mathbb{R}\mathbf{C}^n, \tag{A.31}$$

and works well for obtaining a solution for a relatively small domain. However, matrix inversion is computationally expensive, and it must be noted that the size of the matrices  $\mathbb{L}$  and  $\mathbb{R}$  grow with the square of the size of the column vector  $\mathbf{C}$ . Some computational effort can be avoided by defining the matrices  $\mathbb{L}$  and  $\mathbb{R}$  as sparse matrices (i.e., containing very few non-zero elements).

Iterative methods (see, e.g., Barrett et al. (1994)) is another approach for solving linear systems such as Eq. (A.30). Consider the linear system  $\mathbb{L}\mathbf{C} = \mathbf{b}$  at a time  $n + 1$ , for  $\mathbf{b} = \mathbb{R}\mathbf{C}^n$ . The general objective of an iterative method is to construct a sequence of improving approximate solutions  $\{\mathbf{C}^{(k)}\}_{k=1}^{\infty}$ , that converges to the fixed vector  $\mathbf{C}^*$  which is the solution of the linear system. One such method specifically developed for solving non-symmetric linear systems (such as the Crank-Nicolson scheme applied to the advection-diffusion equation) is the Biconjugate Gradient Stabilized Method (often abbreviated BiCGSTAB), (see, e.g., Barrett et al. (1994, page 27)).

## A.4 Taylor's Theorem

Taylor's Theorem is here presented as in Spivak (1994, p.417).

**Theorem A.1 (Taylor's Theorem)** *Suppose that  $f', \dots, f^{(n+1)}$  are defined on  $[a, x]$ , and that  $R_{n,a}(x)$  is defined by*

$$f(x) = f(a) + f'(a)(x-a) + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n + R_{n,a}(x).$$

Then

$$(1) \quad R_{n,a}(x) = \frac{f^{(n+1)}(t)}{n!}(x-t)^n(x-a) \text{ for some } t \text{ in } (a, x).$$

$$(2) \quad R_{n,a}(x) = \frac{f^{(n+1)}(t)}{(n+1)!}(x-a)^{n+1} \text{ for some } t \text{ in } (a, b).$$

Moreover, if  $f^{(n+1)}$  is integrable on  $[a, x]$ , then

$$(3) \quad R_{n,a}(x) = \int_a^x \frac{f^{(n+1)}(t)}{n!}(x-t)^n dt$$

(If  $x < a$ , then the hypothesis should state that  $f$  is  $(n+1)$ -times differentiable on  $[x, a]$ ; the number  $t$  in (1) and (2) will then be in  $(x, a)$ , while (3) will remain true as stated, provided that  $f^{(n+1)}$  is integrable on  $[x, a]$ .)

## A.5 The Fokker-Planck equation

The following theorem is **Theorem 1** from Gichman (1972), page 102, given here for a clarification of the smoothness criteria on the coefficients  $a(\cdot)$  and  $b(\cdot)$ :

**Theorem A.2 (Kolmogorov's first equation)** *Assume the partial derivatives*

$$\begin{aligned} \frac{\partial}{\partial x} a(x, t), \quad \frac{\partial}{\partial x} b(x, t), \quad \frac{\partial^2 b(x, t)}{\partial x^2}, \quad \frac{\partial p(x, t, y, s)}{\partial s}, \\ \frac{\partial p(x, t, y, s)}{\partial y}, \quad \text{and} \quad \frac{\partial^2 p(x, t, y, s)}{\partial y^2} \end{aligned}$$

exists. Then for  $s > t$ ,  $p(x, t, y, s)$  satisfies

$$\frac{\partial p(x, t, y, s)}{\partial s} = \frac{1}{2} \frac{\partial^2}{\partial y^2} (b(y, s)p(x, t, y, s)) - \frac{\partial}{\partial y} (a(y, s)p(x, t, y, s))$$

In the above theorem, we have kept the original notation. We have used the notation  $p(x, t|x_0, t_0)$  in Chapter 2 for the transition probability density from an initial position  $x_0$  and initial time  $t_0$ . It should also be noted that the coefficient  $b$  is not defined as in subsection 2.2.4 in this thesis, but defined in the following way as  $b = \sigma^2$  for the stochastic differential equation

$$d\eta(t) = a(t, \eta(t))dt + \sigma(t, \eta(t))dw(t), \tag{A.32}$$

in Gichman (1972), e.g., on page 67. Hopefully without creating confusion, our notation is  $b = \sigma$ . To be specific, in the notation we have used in Chapter 2, the correct partial derivatives including the drift coefficient presented in the theorem above, is written in our notation as

$$\frac{\partial(b^2)}{\partial x} \quad \text{and} \quad \frac{\partial^2(b^2)}{\partial x^2}. \tag{A.33}$$

We have however kept the expressions simple,  $\partial_x b$  and  $\partial_x^2 b$  in Eq. (2.2.24), as they i.e. both assume that both the first and second derivatives of the coefficient  $b$  exist,

$$\frac{\partial(b^2)}{\partial x} = 2b \frac{\partial b}{\partial x} \quad \text{and} \quad \frac{\partial^2(b^2)}{\partial x^2} = 2 \left( \frac{\partial b}{\partial x} \right)^2 + 2b \frac{\partial^2 b}{\partial x^2}. \quad (\text{A.34})$$

## B COMPLIMENTARY NUMERICAL WORK

### B.1 Equivalence of Eulerian and Lagrangian formulations in 2D

Through the numerical examples in Section 2.5 we looked at a purely diffusive system with a box-shaped diffusion function in the middle of the domain. This is the two-dimensional problem equivalent to the one presented in Section 2.5.1: we look at a discontinuous diffusion coefficient with a square box shape, as well as a smooth diffusion coefficient describing the smoothed out counterpart. We solve the advection-diffusion equation using the fluid method and the particle method described in Chapter 3, and use kernel density estimation to compare the solutions.

#### Implementation of diffusive system in two variants

The square box function  $D_b$  is implemented in two dimensions as

$$D_b(\mathbf{x}) = D_b(x, y) = \begin{cases} D_0 & \text{if } |x - a| \leq R, \text{ and } |y - b| \leq R, \\ 0.1D_0 & \text{else, for } x \in [0, 2], y \in [0, 1], \end{cases} \quad (\text{B.1})$$

for parameters given in Table B.1.

The smooth box function  $D_s$  is implemented using a hyperbolic tangent function  $\theta$ , with respect to a distance function,  $d(\mathbf{x})$ ,

$$\theta(\mathbf{x}) = \frac{1 + \tanh\left(\frac{d(\mathbf{x})}{w}\right)}{2}, \quad (\text{B.2})$$

where  $w$  is the transition width (approximately controls the thickness of the smooth transition zone). The distance function is given by

$$d(\mathbf{x}) = d(x, y) = R - [(x - a)^\gamma + (y - b)^\gamma]^{\frac{1}{\gamma}}. \quad (\text{B.3})$$

This result in the following diffusion coefficient

$$D_s(\mathbf{x}) = \theta(\mathbf{x}) \cdot D_0 + 0.1D_0, \quad (\text{B.4})$$

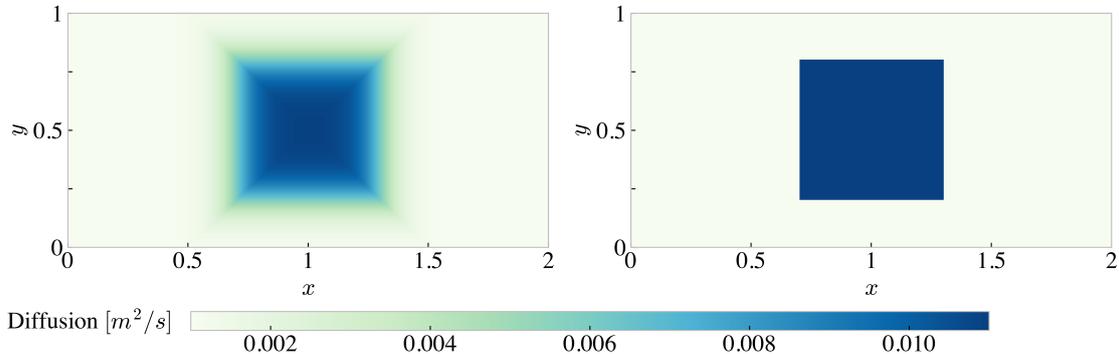
implemented for the parameters in Table B.1.

**Table B.1:** Parameters for implementation of the box function  $D_b(\mathbf{x})$  using Eq. (B.1), and the smooth box function  $D_s(\mathbf{x})$  using Eqns. (B.2-B.4).

$a$	$b$	$R$	$w$	$\gamma$
1	0.5	0.3	0.1	100

### Well-mixed initial condition

The well-mixed condition, discussed in Chapter 2.5, is implemented as a uniform field over the entire domain for the fluid method, ensuring that it integrates to one. For the particle method, the  $N$  particles' initial positions are drawn from the uniform distribution `np.random.uniform` in each dimension.



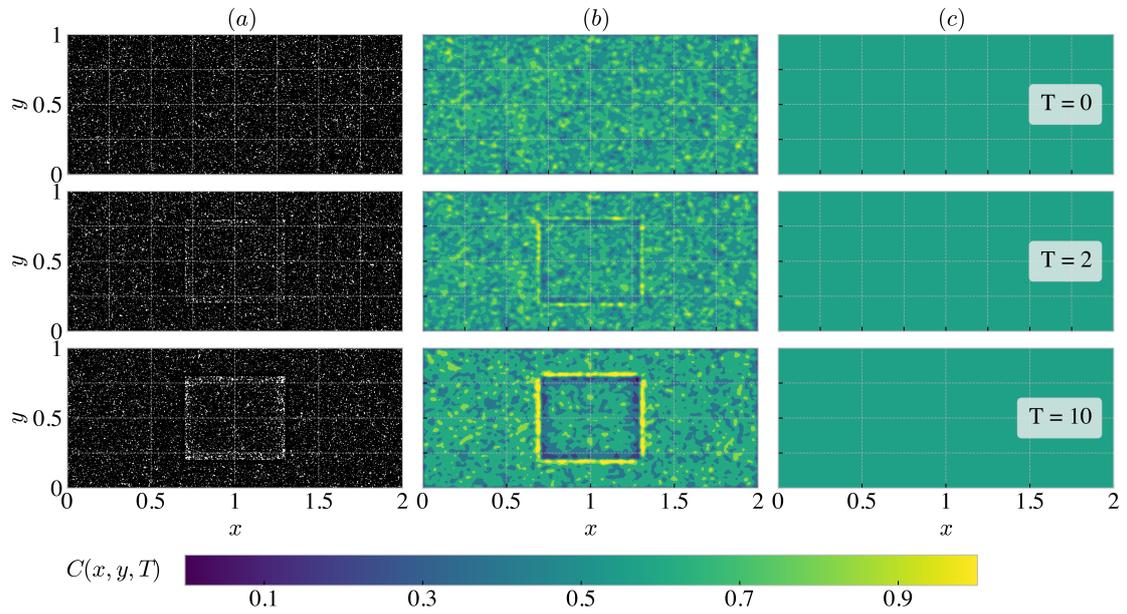
**Figure B.1:** The two-dimensional diffusion coefficients: left subplot shows the smooth function  $D_s(x, y)$ , and right subplot shows the discontinuous function  $D_b(x, y)$ . Both diffusion coefficients have a minimum value of 0.001 and a maximum value of 0.011, and a box shape of length 0.6. The diffusion constant is  $D_0 = 0.01$ , and they are displayed on a  $(200 \times 400)$ -grid.

### Squared and smooth box diffusion in zero advection

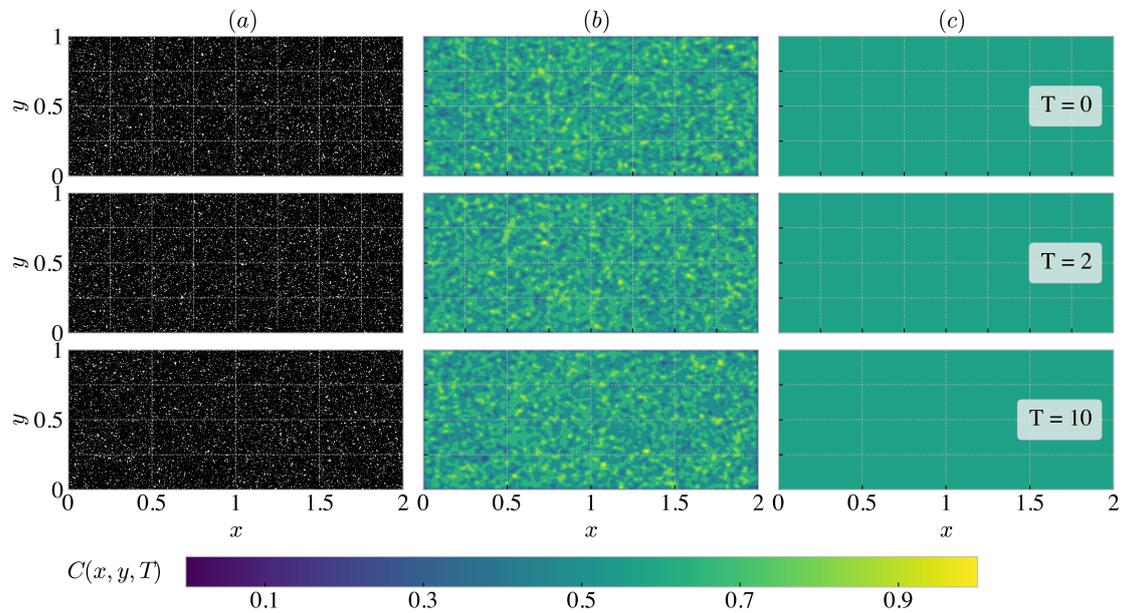
The criterion for there to be an equivalence between solving the advection-diffusion equation using the fluid method and the particle method, evolves around the smoothness of the drift and diffusion coefficients  $\mathbf{u}(x, y, t)$  and  $D(x, y, t)$ . For a purely diffusive system, the criterion (2.2.24) explicitly demands that the diffusion coefficient is twice continuously differentiable.

In Section 2.5.1, the diffusion coefficients investigated were the one-dimensional box function and a smoothed out counterpart, which respectively are discontinuous and smooth functions. The numerical example showed how the non-smooth box-shaped diffusion gave deviating results comparing the two formulations. However, the smoothed out counterpart of the non-differentiable box-diffusion turned the problem back to a satisfying description in both formulations. It fulfilled the requirements, and hence the two methods were found to give the same results. For the two-dimensional numerical example, we solve the advection-diffusion equation in a system with the two-dimensional versions of the box and the smoothed box given as given in Section B.1. Figure B.1 shows the two diffusion fields that we have investigated,  $D_s(\mathbf{x})$  and  $D_b(\mathbf{x})$ . With the initial condition being the well-mixed condition, the distribution is expected to remain a uniform distribution for however long the simulation lasts.

The advection-diffusion equation is solved for a non-smooth diffusion coefficient  $D_b$ , using Crank-Nicolson scheme in two dimensions, and is shown at different times to the right (*c*) in Figure B.2. The fluid method gives a uniform distribution at all integration times. Solving the same system using the particle method solution, result in a different result, as expected. Figure B.2 gives the Lagrangian particles to the left (*a*), and the estimated distribution in the center plot (*b*). The results resemble the one dimensional case, with growing concentration just outside the box boundaries, where the discontinuities are found. The two formulations are not equivalent for diffusion coefficient  $D_b$ . Attempting to solve the system for the differentiable counterpart of the box-shaped diffusion, gives the desired results: The two formulations appear to be equivalent. Figure B.3 shows both the Lagrangian particles (*a*), the estimated probability density (*b*), and the fluid method solution (*c*) describing the exact same well-mixed result.



**Figure B.2:** Concentration  $C(x, y, t = T)$  on a  $(200 \times 400)$ -grid. The system is nonadvective, with a discontinuous diffusion coefficient  $D_b$  given by Eqn. (B.1). The diffusion constant is  $D_0 = 0.01$ , and the bandwidth of the kernel density estimator is  $\Delta = 0.01$ .

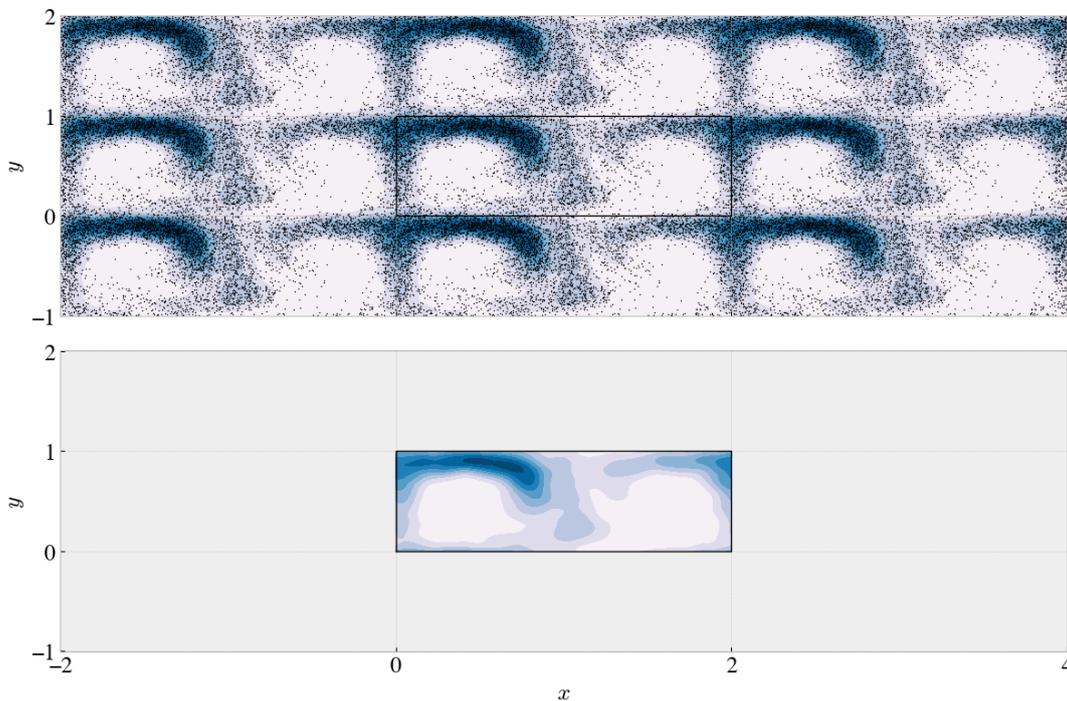


**Figure B.3:** Concentration  $C(x, y, t = T)$  on a  $(200 \times 400)$ -grid. The system is non-advective, with a smooth diffusion coefficient  $D_s$  given by Eqns. (B.2-B.4). The diffusion constant is  $D_0 = 0.01$ , and the bandwidth of the kernel density estimator is  $\Delta = 0.01$ .

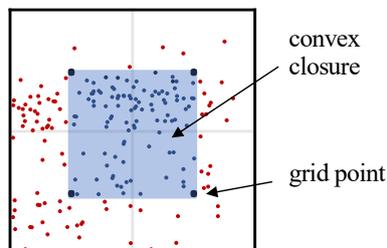
## C NOTES ON NUMERICAL IMPLEMENTATION

### C.1 KDE with periodic boundary conditions

Because we want to compare the results of the particle method directly with the fluid method results, we need to know the estimating distribution at the exact same grid points. We apply periodic boundary conditions by duplicating the solution, such that the original domain has eight duplicates surrounding it, as in Figure C.1. The kernel density estimator is applied to the entire area,  $x \in [-2, 4]$  and  $y \in [-1, 2]$ , such that the boundaries are treated periodically. The solution obtained in the original domain,  $x \in [0, 2]$  and  $y \in [0, 1]$ , is normalized to properly describe a density function. Note that the kernel density estimator expects no samples to lie outside of the convex closure (see Figure C.2) of the grid, and for that reason the grid is made slightly larger at all boundaries.



**Figure C.1:** Plot showing periodic boundary conditions in KDEpy implementations, here for 5000 particles and a bandwidth  $\Delta = 0.05$ . The samples are from simulations in the double gyre flow, after an integration time  $T = 8$ . The upper subplot shows both particles and density estimation for the entire area, while the lower subplot shows only the density estimation for the domain  $x \in [0, 2]$  and  $y \in [0, 1]$ .



**Figure C.2:** The convex closure (or *convex hull*) is the smallest convex set that encloses the shape, which here is the smallest square that encloses the grid points, excluding the parts of the grid cells that fall outside of this minimum square.

## C.2 Crank-Nicolson scheme 2D

### Explicit implementation

A thorough description of the scheme implementations in two dimensions is presented here, a continuation from Section 3.1.2.

The scheme is central in space, and hence result in a spatial five-point stencil, as seen in Figure A.6 of Appendix A.3, considering neighboring points in both directions. Thus, by inserting the approximations (3.1.5)-(3.1.9) into the two-dimensional advection-diffusion equation, we end up with the following:

$$\begin{aligned}
\frac{\mathcal{C}_{j,i}^{n+1} - \mathcal{C}_{j,i}^n}{\Delta t} &= \frac{1}{2} \left( \frac{\partial D}{\partial x} - u \right) \left[ \frac{\mathcal{C}_{j,i+1}^{n+1} - \mathcal{C}_{j,i-1}^{n+1}}{2\Delta x} + \frac{\mathcal{C}_{j,i+1}^n - \mathcal{C}_{j,i-1}^n}{2\Delta x} \right] \\
&+ \frac{1}{2} \left( \frac{\partial D}{\partial y} - v \right) \left[ \frac{\mathcal{C}_{j+1,i}^{n+1} - \mathcal{C}_{j-1,i}^{n+1}}{2\Delta y} + \frac{\mathcal{C}_{j+1,i}^n - \mathcal{C}_{j-1,i}^n}{2\Delta y} \right] \\
&+ \frac{D}{2} \left[ \frac{\mathcal{C}_{j,i+1}^{n+1} - 2\mathcal{C}_{j,i}^{n+1} + \mathcal{C}_{j,i-1}^{n+1}}{(\Delta x)^2} + \frac{\mathcal{C}_{j,i+1}^n - 2\mathcal{C}_{j,i}^n + \mathcal{C}_{j,i-1}^n}{(\Delta x)^2} \right] \\
&+ \frac{D}{2} \left[ \frac{\mathcal{C}_{j+1,i}^{n+1} - 2\mathcal{C}_{j,i}^{n+1} + \mathcal{C}_{j-1,i}^{n+1}}{(\Delta y)^2} + \frac{\mathcal{C}_{j+1,i}^n - 2\mathcal{C}_{j,i}^n + \mathcal{C}_{j-1,i}^n}{(\Delta y)^2} \right] \\
&+ \frac{1}{2} \left( -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) [\mathcal{C}_{j,i}^{n+1} + \mathcal{C}_{j,i}^n]
\end{aligned} \tag{C.1}$$

Next, the equation is rearranged putting all expressions at a future time  $n + 1$  to the left side, and all expressions at the time  $n$  to the right side:

$$\begin{aligned}
&\mathcal{C}_{j,i}^{n+1} \left( 1 + \frac{D\Delta t}{\Delta x^2} + \frac{D\Delta t}{\Delta y^2} - \left[ -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right] \frac{\Delta t}{2} \right) \\
&+ \mathcal{C}_{j,i+1}^{n+1} \left( -\left[ \frac{\partial D}{\partial x} - u \right] \frac{1}{2} \frac{\Delta t}{2\Delta x} - \frac{D\Delta t}{2(\Delta x)^2} \right) \\
&+ \mathcal{C}_{j,i-1}^{n+1} \left( +\left[ \frac{\partial D}{\partial x} - u \right] \frac{1}{2} \frac{\Delta t}{2\Delta y} - \frac{D\Delta t}{2(\Delta x)^2} \right) \\
&+ \mathcal{C}_{j+1,i}^{n+1} \left( -\left[ \frac{\partial D}{\partial y} - v \right] \frac{1}{2} \frac{\Delta t}{2\Delta y} - \frac{D\Delta t}{2(\Delta y)^2} \right) \\
&+ \mathcal{C}_{j-1,i}^{n+1} \left( +\left[ \frac{\partial D}{\partial y} - v \right] \frac{1}{2} \frac{\Delta t}{2\Delta y} - \frac{D\Delta t}{2(\Delta y)^2} \right) \\
&= \mathcal{C}_{j,i}^n \left( 1 - \frac{D\Delta t}{\Delta x^2} - \frac{D\Delta t}{\Delta y^2} + \left[ -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right] \frac{\Delta t}{2} \right) \\
&+ \mathcal{C}_{j,i+1}^n \left( +\left[ \frac{\partial D}{\partial x} - u \right] \frac{1}{2} \frac{\Delta t}{2\Delta x} + \frac{D\Delta t}{2(\Delta x)^2} \right) \\
&+ \mathcal{C}_{j,i-1}^n \left( -\left[ \frac{\partial D}{\partial x} - u \right] \frac{1}{2} \frac{\Delta t}{2\Delta y} + \frac{D\Delta t}{2(\Delta x)^2} \right) \\
&+ \mathcal{C}_{j+1,i}^n \left( +\left[ \frac{\partial D}{\partial y} - v \right] \frac{1}{2} \frac{\Delta t}{2\Delta y} + \frac{D\Delta t}{2(\Delta y)^2} \right) \\
&+ \mathcal{C}_{j-1,i}^n \left( -\left[ \frac{\partial D}{\partial y} - v \right] \frac{1}{2} \frac{\Delta t}{2\Delta y} + \frac{D\Delta t}{2(\Delta y)^2} \right).
\end{aligned} \tag{C.2}$$

The left and right hand side of Eq. (C.2) are, with opposite signs, almost identical. To simplify, the following variables are defined

$$\lambda^x = \frac{\Delta t}{(\Delta x)^2} \quad \lambda^y = \frac{\Delta t}{(\Delta y)^2} \quad \alpha^x = \frac{\Delta t}{4\Delta x} \quad \alpha^y = \frac{\Delta t}{4\Delta y}$$

We rewrite the equation in simplified terms:

$$\begin{aligned} & \mathcal{C}_{j,i}^{n+1} \left( 1 + D\lambda^x + D\lambda^y - \left[ -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right] \frac{\Delta t}{2} \right) \\ & + \mathcal{C}_{j,i+1}^{n+1} \left( - \left[ \frac{\partial D}{\partial x} - u \right] \alpha^x - \frac{1}{2} D\lambda^x \right) \\ & + \mathcal{C}_{j,i-1}^{n+1} \left( + \left[ \frac{\partial D}{\partial x} - u \right] \alpha^x - \frac{1}{2} D\lambda^x \right) \\ & + \mathcal{C}_{j+1,i}^{n+1} \left( - \left[ \frac{\partial D}{\partial y} - v \right] \alpha^y - \frac{1}{2} D\lambda^y \right) \\ & + \mathcal{C}_{j-1,i}^{n+1} \left( + \left[ \frac{\partial D}{\partial y} - v \right] \alpha^y - \frac{1}{2} D\lambda^y \right) \\ & = \mathcal{C}_{j,i}^n \left( 1 - D\lambda^x - D\lambda^y + \left[ -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right] \frac{\Delta t}{2} \right) \\ & + \mathcal{C}_{j,i+1}^n \left( + \left[ \frac{\partial D}{\partial x} - u \right] \alpha^x + \frac{1}{2} D\lambda^x \right) \\ & + \mathcal{C}_{j,i-1}^n \left( - \left[ \frac{\partial D}{\partial x} - u \right] \alpha^x + \frac{1}{2} D\lambda^x \right) \\ & + \mathcal{C}_{j+1,i}^n \left( + \left[ \frac{\partial D}{\partial y} - v \right] \alpha^y + \frac{1}{2} D\lambda^y \right) \\ & + \mathcal{C}_{j-1,i}^n \left( - \left[ \frac{\partial D}{\partial y} - v \right] \alpha^y + \frac{1}{2} D\lambda^y \right) \end{aligned} \tag{C.3}$$

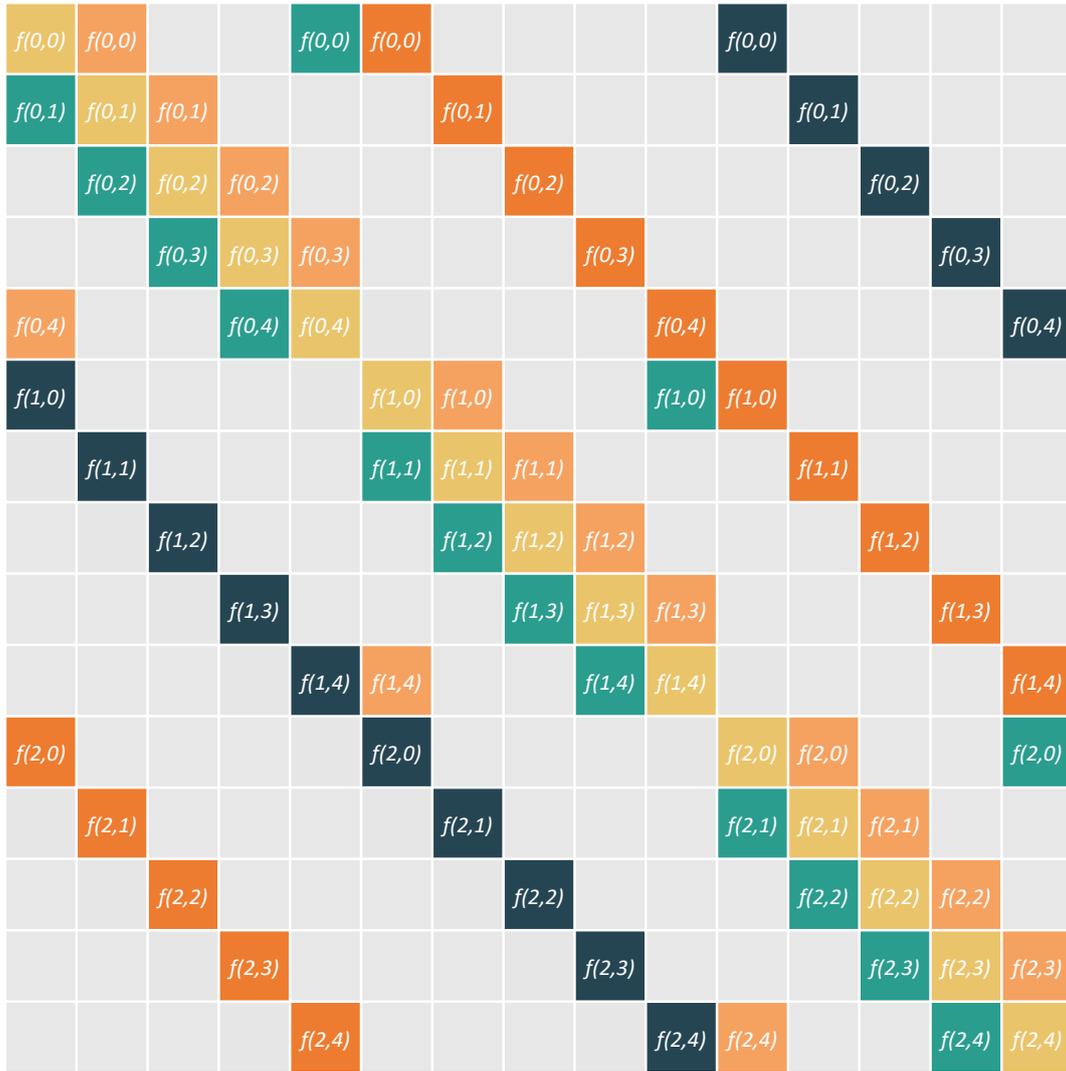
This can be written, for the left hand side matrix  $\mathbb{L}$  and right hand side matrix  $\mathbb{R}$ , and the row-ordered column vector form of  $\mathbf{C}$  (see Section (3.1.1)):

$$\mathbb{L} \cdot \mathbf{C}^{n+1} = \mathbb{R} \cdot \mathbf{C}^n. \tag{C.4}$$

The matrices of the scheme are square  $(Nx \cdot Ny) \times (Nx \cdot Ny)$ , and the concentration vector  $\mathbf{C}$  is of shape  $(Nx \cdot Ny)$ . The matrices are explicitly expressed in Eqs. (3.1.11) and (3.1.12).

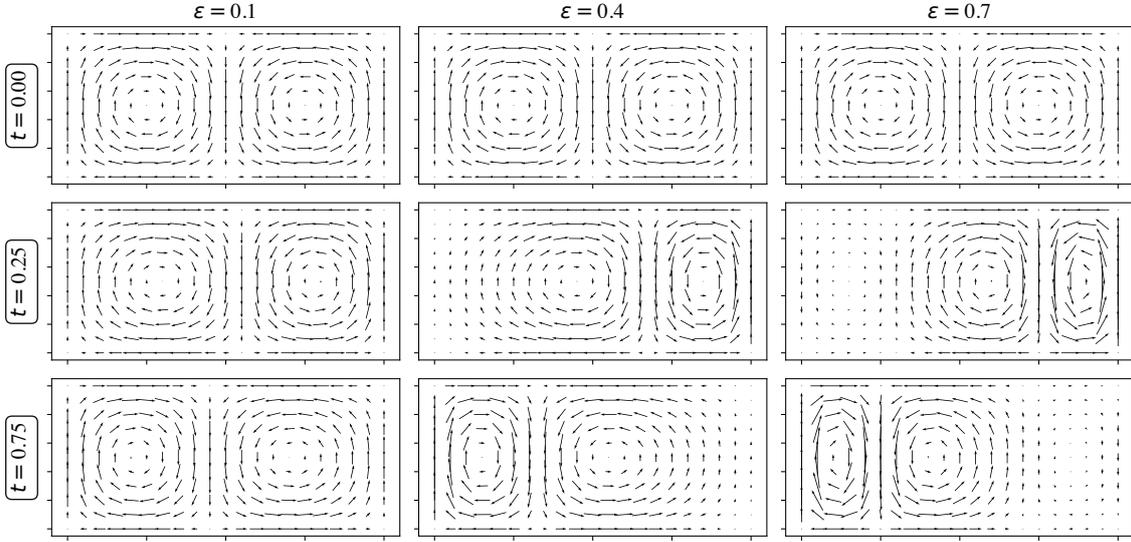
**Scheme matrices**

Each row in the scheme matrices are functions of the same point in space. In other words, for any function within the creation of the matrix diagonals (see (3.1.12) and (3.1.11)), one must ensure that the entry is a function of the correct spatial point. In the illustration below, the function  $f$  can be any field  $\mathcal{U}$ ,  $\mathcal{V}$ ,  $\partial_x \mathcal{U}$ ,  $\partial_y \mathcal{V}$ ,  $\mathcal{D}$ ,  $\partial_x \mathcal{D}$ ,  $\partial_y \mathcal{D}$ . Note how each row is a function of a particular point  $(y_j, x_i)$ , written as  $(j, i)$ . A pitfall here could be assuming that the functions accessed column-wise. The main diagonal is equal in both approaches.



**Figure C.3:** Illustration of the dependency of the fields within the matrices  $\mathbb{R}$  and  $\mathbb{L}$  in the Crank-Nicolson scheme. Colors are presented in Figure 3.1.5, displaying the five-point stencil.

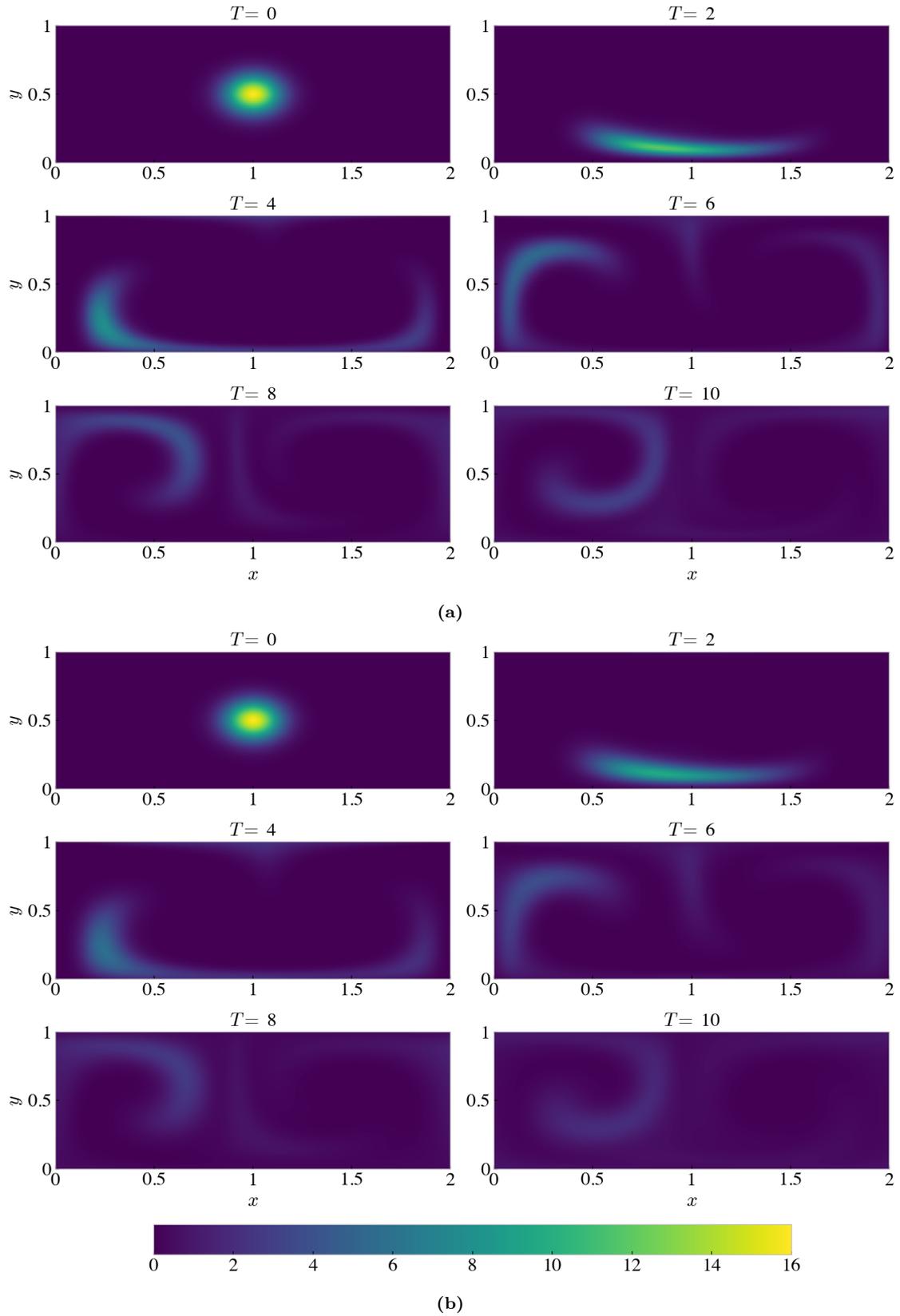
## D ADDITIONAL FIGURES AND TABLES



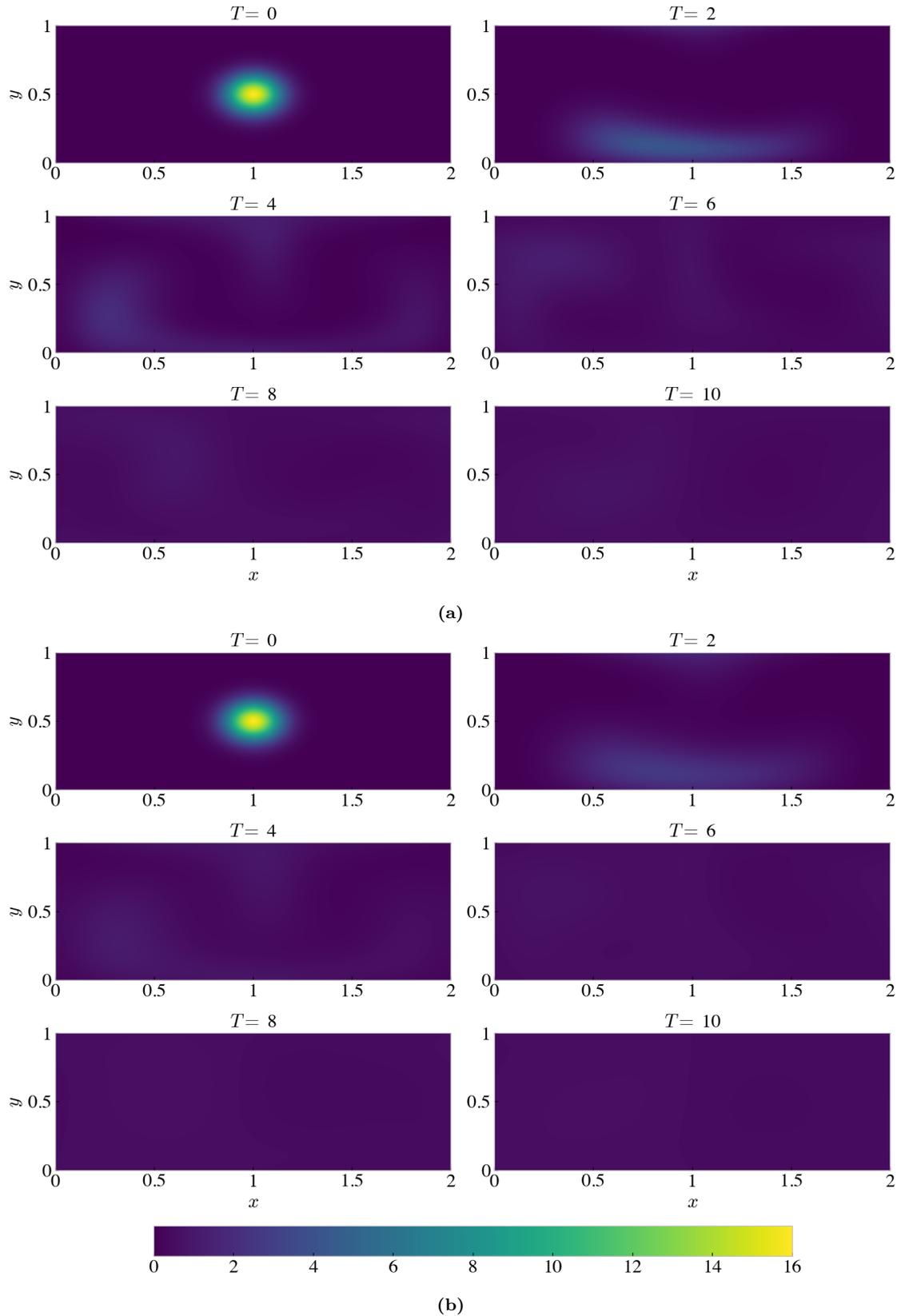
**Figure D.1:** The double gyre system for different values of  $\epsilon$  (see Section 1.1.1). The first row describes the flow at times  $t = 0, 0.5, 1$ . The parameters of the flow are  $A = 0.1$ , and  $\omega = 2\pi$ . The dimensions of each plot is  $x \in [0, 2]$  and  $y \in [0, 1]$ .

**Table D.1:** The optimal bandwidth  $\Delta_{optimal}$  for  $N = 100000$ , for four different values of diffusion.

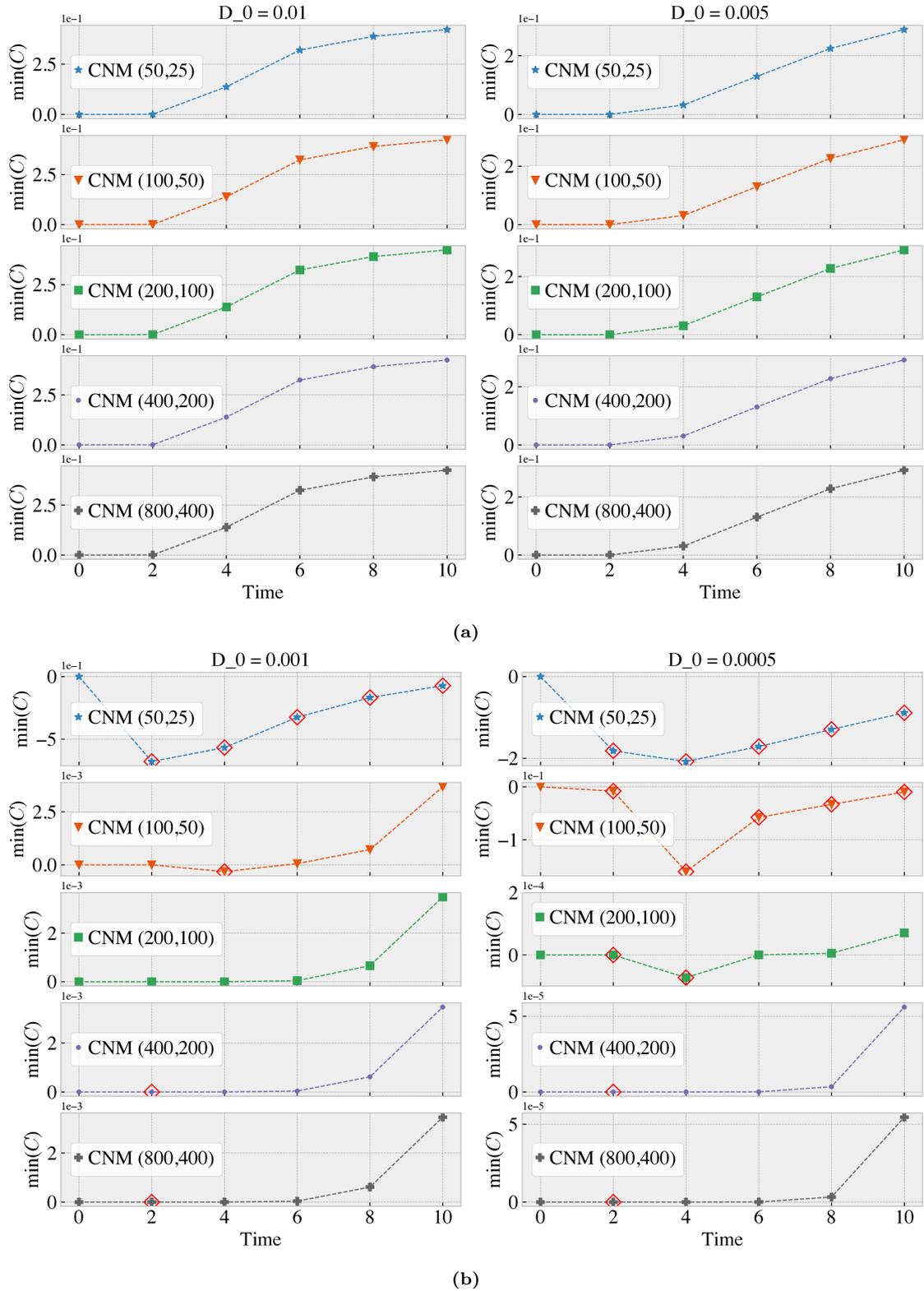
$T \setminus D$		0.01	0.005	0.001	0.0005
2	$\Delta_{optimal}$	0.0029	0.0207	0.0109	0.0093
	ISE	0.0048	0.0137	0.0601	0.0863
4	$\Delta_{optimal}$	0.0629	0.0531	0.0381	0.0375
	ISE	0.0197	0.0705	0.3929	0.7688
6	$\Delta_{optimal}$	0.0655	0.0591	0.0303	0.0285
	ISE	0.00032	0.0247	0.1893	0.4089
8	$\Delta_{optimal}$	0.1200	0.1400	0.0755	0.0747
	ISE	0.0012	0.0195	0.3277	0.6495
10	$\Delta_{optimal}$	0.1000	0.0997	0.0505	0.0463
	ISE	0.0003	0.0072	0.2085	0.4361



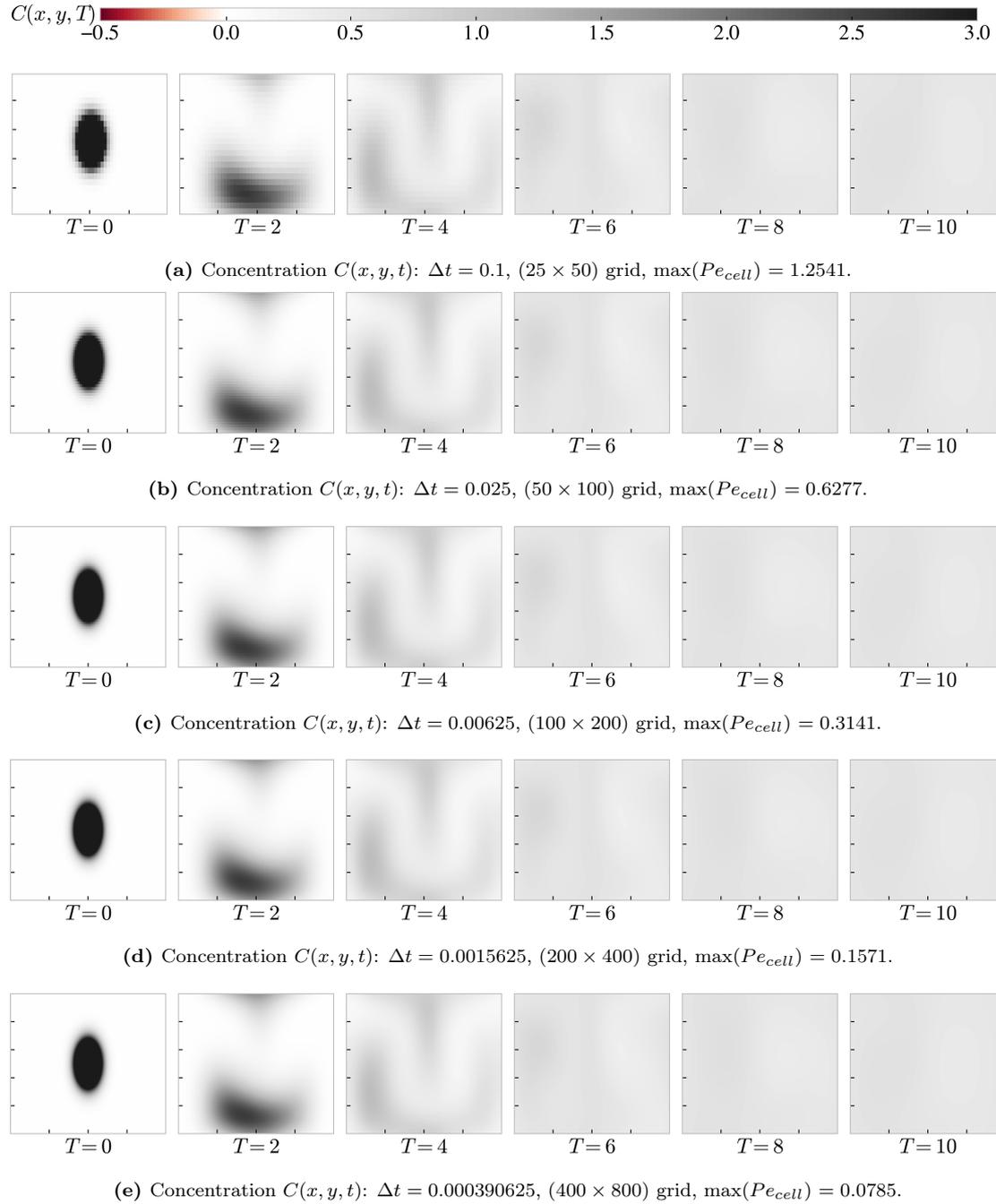
**Figure D.2:** Contour plot of the fluid method solution, in a system of (a)  $D_0 = 0.0005$  and (b)  $D_0 = 0.001$ . Computations on a  $(200 \times 400)$ -grid with a time step of  $\Delta t = 0.0001953125$ .



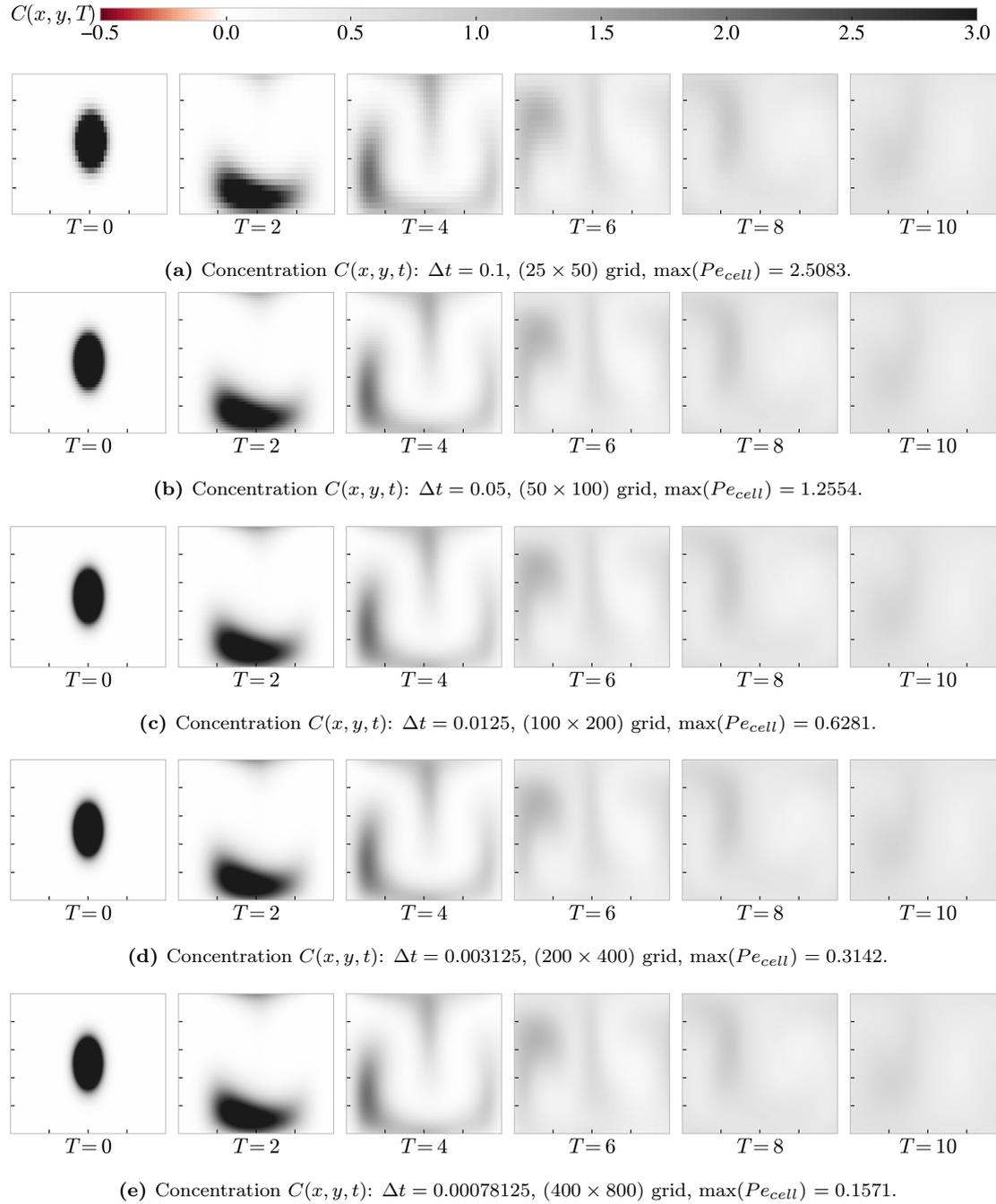
**Figure D.3:** Contour plot of the fluid method solution, in a system of (a)  $D_0 = 0.005$  and (b)  $D_0 = 0.01$ . Computations on a  $(200 \times 400)$ -grid with a time step of  $\Delta t = 0.0001953125$ .



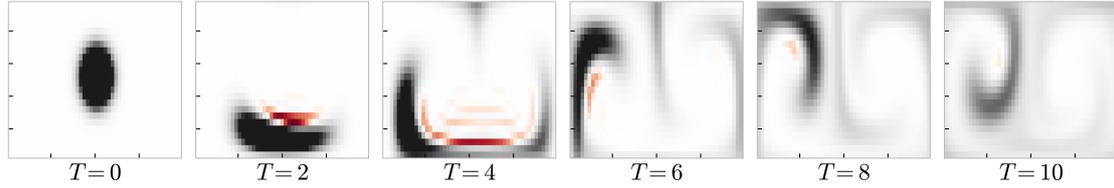
**Figure D.4:** Minima of Crank-Nicolson method (CNM) solution over the grid of size  $(N_x, N_y)$ . (a) No negative values are found, and cell Peclet numbers in each system satisfy  $Pe_{cell} \leq 2$  (see limit (3.1.17)), except in one single case where  $Pe_{cell} = 2.5$  ( $D = 0.005$  for grid  $(N_x, N_y) = (50, 25)$ ). (b) High cell Peclet numbers ( $Pe_{cell} > 2$ ) give rise to unphysical negative numbers, or "oscillating solutions", marked in red  $\diamond$ . See also, Figures D.5-D.8 for reference to distributions, showing a clear oscillating feature in the most severe cases. See Table 3.1.1 for a list of maximum cell Peclet numbers for different systems.



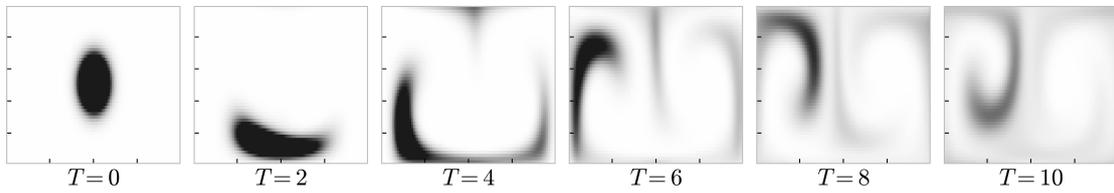
**Figure D.5:** The solution using Crank-Nicolson for different times  $T$ , for the constant diffusion  $D = 0.01$ . Each subplot shows the domain  $x \in [0, 2], y \in [0, 1]$  (cont.)



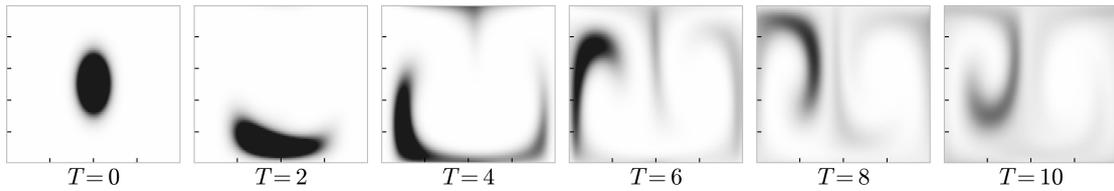
**Figure D.6:** The solution using Crank-Nicolson for different times  $T$ , for the constant diffusion  $D = 0.005$ . Each subplot shows the domain  $x \in [0, 2], y \in [0, 1]$  (cont.)



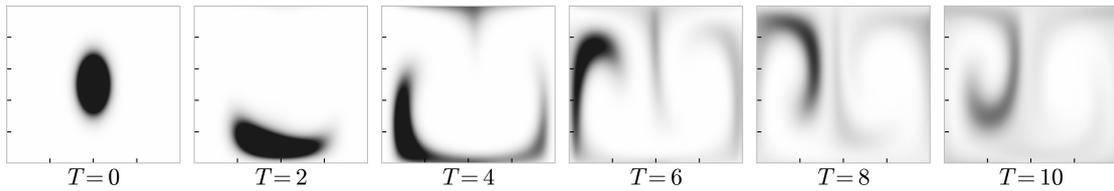
(a) Concentration  $C(x, y, t)$ :  $\Delta t = 0.1$ ,  $(25 \times 50)$  grid,  $\max(Pe_{cell}) = 12.5416$ .



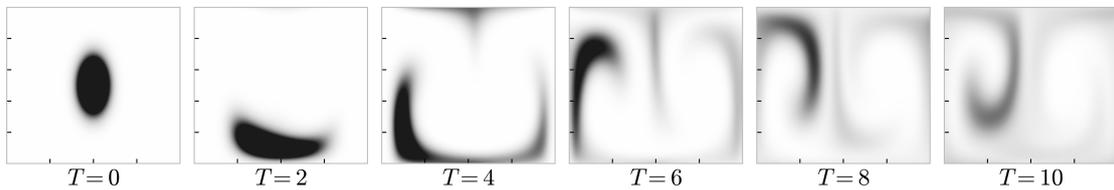
(b) Concentration  $C(x, y, t)$ :  $\Delta t = 0.1$ ,  $(50 \times 100)$  grid,  $\max(Pe_{cell}) = 6.2770$ .



(c) Concentration  $C(x, y, t)$ :  $\Delta t = 0.1$ ,  $(100 \times 200)$  grid,  $\max(Pe_{cell}) = 3.1408$ .

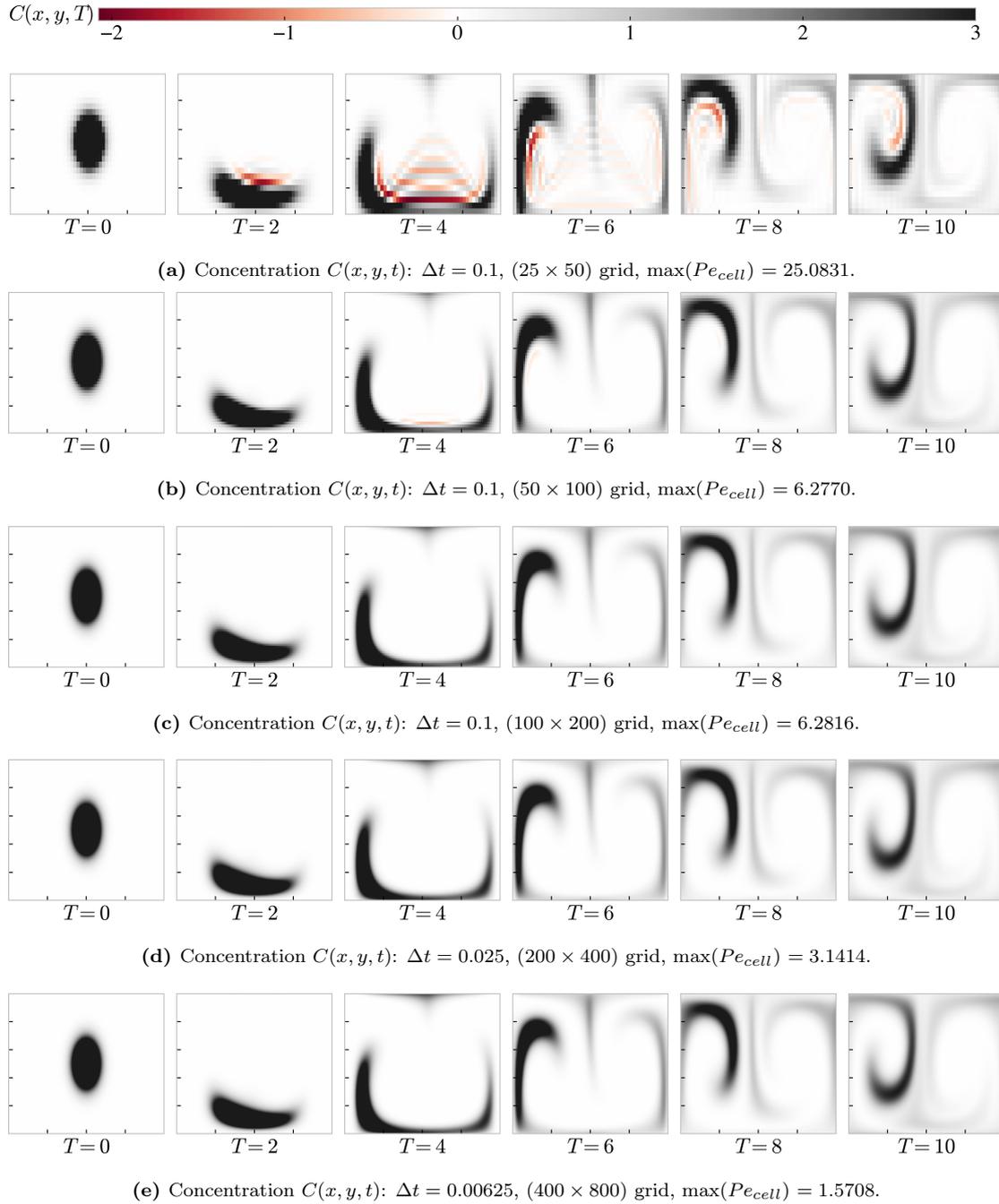


(d) Concentration  $C(x, y, t)$ :  $\Delta t = 0.0125$ ,  $(200 \times 400)$  grid,  $\max(Pe_{cell}) = 1.5707$ .



(e) Concentration  $C(x, y, t)$ :  $\Delta t = 0.003125$ ,  $(400 \times 800)$  grid,  $\max(Pe_{cell}) = 0.7854$ .

**Figure D.7:** The solution using Crank-Nicolson for different times  $T$ , for the constant diffusion  $D = 0.001$ . Each subplot shows the domain  $x \in [0, 2], y \in [0, 1]$  (cont.)



**Figure D.8:** The solution using Crank-Nicolson for different times  $T$ , for the constant diffusion  $D = 0.0005$ . Each subplot shows the domain  $x \in [0, 2], y \in [0, 1]$ . (cont.)

