TRANSPORT MODELLING IN COASTAL WATERS USING STOCHASTIC DIFFERENTIAL EQUATIONS

Transport Modelling in Coastal Waters using Stochastic Differential Equations

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op donderdag 29 March 2007 te 10.00 uur

door

Wilson Mahera CHARLES,

Master of Science in Mathematics
The University of Zimbabwe, Harare.

Born in Musoma, Tanzania.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. A.W. Heemink

Toegevoegd promotor:Dr.ir. H.X.Lin

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. A. W. Heemink	EWI, Technische Universiteit Delft, promotor
Dr.ir. H. X. Lin	EWI, Technische Universiteit Delft, toegevoegd promotor
Prof. J. C. Chuncha	DoCS, The New University of Technology, Lisbon, Portugal
Prof. V. G. Masanja	DoM, University of Dar-es-salaam, Tanzania
Prof. dr.ir. G. S. Stelling,	CITG, Technische Universiteit Delft
Prof. dr. ir. H. J. Sips	EWI, Technische Universiteit Delft
Dr.ir. M. Verlaan	EWI, Technische Universiteit Delft, advisor

ISBN 978-90-9021758-1
Copyright © 2007 by Wilson Mahera Charles

This research was carried out in the section of Mathematical Physics at the Delft Institute of Applied Mathematics (DIAM), Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands

All rights reserved. No part of this publication may be reproduced in any form or by any means of electronic, mechanical, including photocopying, recording or otherwise, without the prior written permission from the author.
Typesetting system: $\LaTeX\ 2_{\varepsilon}$

Printed in The Netherlands by: Print Partners Ipskamp

To my family, parents, brothers, sisters, and all friends

Contents

List of Figures viii
List of Tables xi

List of Symbols and Notations **xv**

1 Introduction **1**
1.1 Research Objectives 2
1.2 Outline of the thesis 4

2 Stochastic modelling of advection-diffusion processes **7**
2.1 Introduction 7
2.2 Wiener Processes 8
2.3 White noise 9
2.4 Modelling 11
  2.4.1 Particle Tracking Models 11
  2.4.2 Stochastic differential equations 11
2.5 Stochastic integrals 12
  2.5.1 Itô differential rule 14
  2.5.2 The relation between Itô and Stratonovich stochastic differential equations 15
  2.5.3 Advection-diffusion process 19
  2.5.4 Consistence of particle model with the advection-diffusion equation 20
2.6 Stochastic Taylor expansion and derivation of stochastic numerical schemes 21
2.7 Numerical schemes 26
  2.7.1 Euler scheme 27
  2.7.2 Milstein Scheme 27
  2.7.3 Heun Scheme 28
  2.7.4 Strong convergence 28
  2.7.5 Weak convergence 29

3 Random flight model for transport of pollutants **33**

3.1 Introduction 33
3.2 The traditional particle model for dispersion in shallow waters 35
3.3 Coloured noise processes 36
  3.3.1 The scalar exponential coloured noise process 36
  3.3.2 The general vector coloured noise force 36
3.4 Modelling dispersion of pollution processes by coloured forcing noise . . . 38
  3.4.1 Random flight model 39
3.5 The spreading behaviour of a cloud of contaminants 40
  3.5.1 Long term behaviour of Brownian motion force 40
  3.5.2 Long term spreading behaviour of a cloud of contaminants subject to coloured noise forcing 40
  3.5.3 The analysis of short term spreading behaviour of a cloud of contaminants 41
  3.5.4 The general long term behaviour of a cloud of contaminants due to coloured noise 42
3.6 The traditional discrete particle model driven by Brownian motion 43
  3.6.1 Boundaries 43
3.7 Numerical Experiments 44
3.8 Application of both the traditional particle and random flight models . . 46
3.9 Conclusions 51
3.10 Appendix 51

4 The particle model for simulation of sediment transport in coastal waters 53
4.1 Introduction 53
4.2 Shallow water flow equations 54
4.3 Eulerian sediment transport model 55
  4.3.1 Determination of bedlevel changes by Eulerian transport model . . 56
4.4 A particle model for sediment transport in shallow waters 57
  4.4.1 Integration of particle movement 57
  4.4.2 Deposition of sediment particles 58
  4.4.3 Suspension of sediment particles 58
  4.4.4 Particle flux at open boundaries 59
  4.4.5 The connection between the Fokker-Planck equation and Eulerian transport model 59
  4.4.6 Determination of bedlevel changes using particle models 61
4.5 Numerical approximation of the particle model 62
4.6 Experiment of sediment transport in the two dimensional channel 62
4.7 Transport of heavy particles in shallow water 65
4.8 Application and results 68
4.9 Discussion and conclusions 69

5 Parallel and distributed simulation of sediment dynamics in shallow water using particle decomposition approach 71

5.1 Introduction 71
5.2 Sediment transport using Lagrangian particle models 73
5.2.1 Determination of bed level changes using Lagrangian models . . . 73
5.3 Parallel processing 74
5.3.1 Work decomposition 75
5.3.2 The process of sediment suspension 76
5.3.3 Sedimentation 80
5.3.4 Equilibrium determining the expected number of particles in the flow 80
5.3.5 Depth update 81
5.4 Implementation 82
5.4.1 Modules 82
5.5 Application 84
5.5.1 Experimental results 84
5.6 Discussion and conclusions 86
5.7 Summary of the simulation parameters 88

6 Variable time stepping in the parallel particle models for transport problems in shallow water 89
6.1 Introduction 89
6.2 The particle model for pollution transport 91
6.3 Higher order strong adaptive scheme for pollution transport using SDEs . 91
6.3.1 An explicit Order 1 Strong Scheme 92
6.3.2 An explicit Order 1.5 Strong Scheme 93
6.3.3 An explicit Order 2 Strong Scheme 94
6.3.4 Determination of variable time step sizes 97
6.4 Variable time stepping implementation in SDEs 98
6.5 Schematic summary of the adaptive particle model 99
6.6 Experiments of adaptive particle model parallel processing 100
6.6.1 Results of the test case 101
6.6.2 Results from the application of the model to the real data 101
6.7 Concluding remarks 107

7 General conclusion and recommendations 109
7.1 General conclusions 109
7.2 Recommendations 111

Bibliography 113

A 119

B 121
B.1 Approximation of the multiple stratonovich integral $J_{(1,1,0)}$ 125

Summary **127**

Samenvatting **129**

Acknowledgments **131**

Curriculum Vitae **133**

List of Figures

2.1 A Wiener path 9

2.2 Strong solution of an Itô SDE (2.25) for $a = 5, b = 2$, $\Delta t = 2^{-8}$ by Euler scheme (b) of its corresponding Stratonovich SDEs (2.26) for $a = 3, b = 2$, $\Delta t = 2^{-8}$ by Heun scheme. 29

2.3 Strong solution of (a) an Itô SDEs (2.25) for $a = 7, b = 2$, $\Delta t = 2^{-8}$ using Euler scheme (b) of Itô SDE(2.25) for $a = 7, b = 2$, $\Delta t = 2^{-8}$ due to the Milstein scheme. 30

2.4 (a) Results of Itô SDE (2.25) for $a = 2, b = 1$ due to Euler's 0.5 strong order (b) Strat.SDE (2.26) $a = 1.5, b = 1$ due to Heun 1 strong order. . . . 30

3.1 (a) Shows that the mean of the coloured noise processes tends to zero, (b) Shows that the variance of coloured noise processes started from non-stationary state to stationary state. 38

3.2 The auto-covariance of coloured noise processes started from non-stationary state for $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1$ and tends to zero at stationary state. . . 39

3.3 Domain showing flow velocity and a point at which particles are released. 45

3.4 (a) Dispersion of a cloud of 40, 000 particles released in the idealised channel domain by using Random flight for $t < T_L$, (b) Dispersion of a cloud of 40,000 particles released in the idealised channel domain by using Tradition particle model for $t < T_L$. 46

3.5 (a) Dispersion of a cloud of 40,000 particles released in the idealised channel domain by using Random flight for $t \gg T_L$, (b) Dispersion of a cloud of 40,000 particles released in the idealised channel domain by using Tradition particle model for $t \gg T_L$. 47

3.6 (a) Tracking of a single marked particle domain by using Random flight for $t < T_L$, (b) Tracking of a single marked particle domain by using Tradition particle model for $t < T_L$. 47

3.7 (a) Tracking of a single marked particle by using Random flight for $t \gg T_L$, (b) Tracking of a single marked particle by using Tradition particle model for $t \gg T_L$. 48

3.8 Distribution of a cloud of 20,000 particles in the idealized empty domain showing the two zones. 48

3.9 (a) Example of tidally averaged flow fields 3 hours after, (b) Example of tidally averaged flow fields 6 hours after. 49

3.10 (a) Dispersion of a cloud of $400,000$ particles released at $(x, y) = (135km, 570km)$ in the Wadden sea by using Random flight for $t < T_L$, (b) Dispersion of a cloud of 400,000 particles released at $(x, y) = (135km, 570km)$ in the Wadden sea by using Tradition particle model for $t < T_L$. 49

3.11 (a) Dispersion of a cloud of $400,000$ particles released at $(x, y) = (135km, 570km)$ in the Wadden sea by using Random flight for $t \gg T_L$, (b) Dispersion of a cloud of 400,000 particles released at $(x, y) = (135km, 570km)$ in the Wadden sea by using Tradition particle model for $t \gg T_L$. 50

3.12 (a) Tracking of a single marked particle domain by using Random flight for $t < T_L$, (b) Tracking of a single marked particle by using Tradition particle model for $t < T_L$. 50

3.13 (a) Tracking of a single marked particle by using Random flight for $t \gg T_L$, (b) Tracking of a single marked particle by using Tradition particle model for $t \gg T_L$. 51

4.1 Two dimensional channel showing the inflow, main domain and outflow. . 63

4.2 Change of bed level in $m/year$ for a two dimensional channel (a) is due to the particle model, while (b) the result is computed by using the Eulerian approach. 64

4.3 Change of bed level in $m/year$ for a two dimensional channel (a) is due to the particle model, while (b) the result is computed by using the Eulerian approach. 64

4.4 Cumulative local changes in depth by using data of 90 days for two selected grid cells in the ideal two dimensional channel domain. 65

4.5 Cumulative local changes in depth by using data for 90 days for light particles in the two selected grid cells. 68

4.6 Change of bed level in $m/year$ (a) is the change computed by using the particle model for light particle, while (b) the result is computed by using the Eulerian approach. 69

4.7 Change of bed level in $m/year$ for 90 days, (a) is the change computed by using the particle model for Heavy particle, while (b) is computed by using the Eulerian approach 69

5.1 Cumulative local changes in depth by using data for 90 days for two selected grid cells. 74

5.2 (a) Domain decomposition with different parts of the domain assigned to different processor, (b) ghost regions for processor 1 for the same decomposition, indicated by the shaded boxes. 77

5.3 Assigning n_k mod np particles equally among processors for three successive cells k. For the first cell we have a remaining 3 particles to assign to 8 processors, which is indicated by the shaded blocks. For the second cell we have 4 which are assigned to processors 3–6. Finally, a remaining 2 particles are assigned to processors 7, and 0. 79
5.4 An overview of the software for the parallel particle model for sediment transport in shallow water. 82
5.5 (a) The total number of particles in each processor summed over all the iterations for each approach (b)The difference between the maximum and the minimum number of particles among the processors at every iteration for each approach;(c) The maximum number of particles in a processor at every iteration for each approach. 85
5.6 (a) Parallel speedups obtained by the three sediment suspension methods (b) Efficiency of the three methods. 86
5.7 Sedimentation in the Dutch North sea in m over 80 days. The effect is most pronounced in the regions between the islands. 87

6.1 A sample path of Brownian bridge. 96
6.2 An overview of the software for the parallel adaptive particle model for pollution dispersion in shallow water. 100
6.3 Simulation results (a) flow fields (b) Variations of step sizes long the domain (c) Snap shot of particles' position at every 5 minutes (d) Speed up measured on a Beowulf cluster. 102
6.4 (a) Fixed dt over # of steps for a single track 1 (b) Fixed dt over # of steps for a single track (c) The distribution of the number of # of steps for single track using fixed dt (d) The distribution of the number of # steps for single track using variable dt 104
6.5 Simulation results (a) variations of step sizes at different locations along x direction (b) variations of step sizes at different locations along y direction (c) variations of step sizes over all iterations (d) Speed up measured on a Beowulf cluster. 105
6.6 (a) Snapshot taken after 0 second (b) Snapshot taken after 20000 seconds (c) Snapshot taken after 40000 seconds (d) Snapshot taken after 60000 seconds 106

List of Tables

3.1 The simulation parameters of the particle model for the dispersion of pollutants in shallow waters. 52

4.1 Parameters used by particle model for the sediment transport in the test case 66
4.2 Parameters used by the particle model for real data. 70

5.1 Simulation parameters of the parallel particle model for the sediment transport model. 88

6.1 Summary of the simulation data using fixed dt and variable dt in the shallow waters using the Lagrangian approach 103

List of Symbols and Notations

Symbols and notations transport model

List of symbols and notations

- SDEs: Stochastic Differential Equations.
- FPE: Fokker-Planck Equation.
- PDE: Partial Differential Equation.
- ADE: Advection Diffusion Equation.
- PMs: Particle Models.
- APMs: Adaptive Particle Models.
- LPMs: Lagrangian Particle Models.
- CN: coloured noise process.
- BM: Brownian motion process.
- WN: Wiener process.

Variable/ Parameter	**Dimension**	**Description**
C	$\left[\frac{kg/m^3}{s}\right]$	Concentration of particles
D	$\left[\frac{m^2}{s}\right]$	Diffusion Coefficient
H	$[m]$	Depth with respect to water level
U	$[m/s]$	Water velocities along x direction

V	$[m/s]$	Water velocities along y direction
T	$[s]$	Tidal period
T_L	$[s]$	Lagrangian time scale
x	$[m]$	Position in horizontal along x direction
y	$[m]$	Position in horizontal along y direction
$X_0 =$	$[m]$	Initial position of a particle at $t = t_0$
X_k	$[m]$	Position of a particle in the discrete form
Y_t	$[m]$	Position of a particle along y direction in the model
$Y_0 =$	$[m]$	Position of a particle at $t = t_0$ along y direction
Y_k	$[m]$	Position of a particle in the discrete along y direction
$Y(t)$	$[m]$	Position of a particle along y direction in the model
B_k	$[\sqrt{s}]^2$	Brownian motion process in the discrete model
$B(t)$	$[\sqrt{s}]^2$	Brownian motion process in the model
γ	$[s^{-1}]$	Deposition Coefficient
λ_s	$[kgsm^{-4}]$	Erosion Coefficient of sediments particle
δ_b	$[-]$	Sea bed slope coefficient
po	$[-]$	Sea bed porosity
$\mathcal{M}_p$	$[Kg]$	Mass of each single particle
$enp_{(i,j,t)}$	$[-]$	Expected number of particles suspended in each grid cell
$F(x)$, $F_X(x)$	$[-]$	Distribution function
$p(x)$, $p_X(x)$	$[-]$	Probability density function
$p(x\|y)$	$[-]$	Conditional probability density function
$p(x_1, \ldots, x_n)$	$[-]$	Joint probability density function
$p(t, x, \tau, y)$	$[-]$	Transition density function
$\mathbb{E}\{\mathbb{X}\}$	$[-]$	Expectation of X
$\text{var}\{\text{X}\}$	$[-]$	Variance of X
$\text{std}\{\text{X}\}$, $\sigma\{X\}$	$[-]$	Standard deviation of X
$\text{cov}\{\text{X}\}$	$[-]$	Covariance of X
P_X	$[-]$	Covariance matrix of X
μ	$[-]$	Mean
σ	$[-]$	Standard deviation
$\text{N}(\mu, \sigma^2)$	$[-]$	Normal distribution with mean μ and standard deviation σ
$W(t)$	$[-]$	White noise process
$S(p)$	$[-]$	Speedup
$E(p)$	$[-]$	Efficiency

Chapter 1

Introduction

All over the world, ecosystem in shallow waters is being threatened because of pollution and human activities such as dredging which are constantly increasing throughout the years. This can be disastrous and have indirect and direct negative impacts on ecosystem. The increasing importance of this issue has contributed to the global awareness of the importance the environment to the societies. The public opinion awareness has been fostering the need for a better understanding of how these processes behave in the shallow waters. Numerical simulation of how particles or pollutants behave in a water body is fundamental. This helps to locate for instance, a waste water discharge for accurate predictions of these processes and how to optimise dredging activities in the harbour for example. Therefore, the theoretical foundations to support the transform and transport in water bodies are of paramount importance.

In general, a coastal zone has a number of different functions such as housing, fishing agriculture, water supply navigation and recreation. All these functions are interconnected and often are the cause of conflicting interests. Coastal zone is not limited to just the boundary between the land and sea, but extends considerably seaward and landwards of the coastline. There are many socio-economic activities taking place day to day hence the coastal study is very important. Some of the activities are human influenced such as the need for continuous dredging of natural obstructions in the harbours which is an economic necessity to provide passage for large container ships. But this can in one way cause siltation and deposition of sediments to unwanted locations which might lead to an environmental loss.

Furthermore, growing industrialisation in the coastal zone has increased the pollution in rivers, sea and lakes. Therefore the fact that shallow waters such as that in coastal zone is important to humankind, it is therefore necessary to carry out research that can contribute to answering some question so as to make it sustainable. The dissolved materials in shallow waters need to be accurately predicted, so numerical simulation plays a vital role.

Transport of dissolved and suspended matter in shallow waters is generally described by the advection-diffusion equations(ADEs). This type of mathematical model is in the form of partial differential equations(PDEs). They describe the changes and variations of the concentrations of pollutants or sediment transport in space and time.

On the other hand, an alternative way to describe transport processes in shallow waters is to consider the matter consisting of a large number of particles whose movement is partly random due to the small turbulence in the fluid [32, 21]. This type of mathematical model is stochastic in nature and is in the form of stochastic differential equations which is often known as *particle model.* Often Lagrangian numerical technique solves the particle model numerically by the following tracks of particles.

The behaviour of the movement of each individual particle is simulated by decomposing the movement of a particle into a drift and a diffusion part [29, 51, 4, 15, 40]. The diffusion part is a noise term in which the rapidly fluctuating physical processes are modelled by Brownian process or coloured noise processes [28, 32]. Therefore, the implementation of a numerical method for the stochastic differential equations requires the sampling of the Wiener process to approximate the white noise by means of computer generations of pseudo-random numbers [32, 34, 35]. From the theoretical point of view this method does not have spatial error instabilities.

For a particle model to give a better solution, a large number of samples is required. This demand considerably increase in computational costs in the transport models. There are several ways of mitigating the problem, one way is by using adaptive schemes with different resolutions. Variable time step size is implemented in the Lagrangian particle tracking so as to avoid unnecessary smaller step size. Another, more general approach for reducing the required computer time is to use more powerful computers or distributed computing to speed up the computations. The use of cluster is there to reduce this problems at a very low cost. The parallel computing in both hydrodynamic and transport models in shallow waters often resort to the domain decomposition techniques [11, 14, 38, 60, 13]. However, the dynamical effects of tides can lead to a very high load imbalances when domain decomposition is implemented. There are ways that can be found to be useful in minimizing of load balance problem, for example in his thesis [14] Costa used both the mixture of the so called scattered partitioning and dynamic load balance. However, in this thesis particle decomposition parallel approach is implemented. Regardless of the dynamics of the hydrodynamics, if well designed, the use of particle decomposition approach can be found to be useful to minimise load imbalances. Furthermore, the Dutch region constitutes the main real-life application of the model. In this area, the North sea receives pollution, for example released from the rivers such as the Rhine, Meuse and so forth. The main contributor of pollution is the river Rhine [20]. Many studies have been conducted on pollution in this part [28, 51, 20].

1.1 Research Objectives

This thesis intends to address two distinct areas concerning the simulations of the transport of dissolved but inactive materials in the shallow waters. The first part focuses on the modelling aspects such as

(I.1) Designing/formulation of the particle model for pollutants transport that corrects the short term behaviour of the movements of the particles shortly after the deployment of the pollutants in shallow waters. The displacement of the particle shortly after the deployment is correlated with time. But the Wiener process assumes that the increments are independent throughout. Therefore, instead of using the Wiener process, we correct this by using the colored noise where the velocity of the particle is as well considered. The model is designed in such a way that eventually it is consistent with the advection-diffusions when the time scale is larger than the smallest time scale known as the Lagrangian time.

(I.2) To modify the particle model in part (I.1) to suite the simulation of sediment transport in shallow waters. It is done by using probabilistic concept to include in the particle model the modelling of the deposition and erosion processes. In this case the noise term is modelled by using the Wiener process. This is followed by showing the consistency of the designed particle model with the Eulerian sediment transport model. Therefore, the simulation parameters for the particle model will be adapted from the Eulerian transport model. Because of the relative easiness of the numerical implementation and modification of the particle model, in this thesis the particle model for sediment transport is implemented.

The second part focuses on parallel and distributed simulation and variable time step size.

(II.1) The implementations of the numerical scheme of the particle model sequentially for both pollutants and / or sediments transport in shallow waters. The inputs such as water flow fields and depths are computed separately by the two dimensional hydrodynamic model [49]. From the theoretical point of view it is known that for accurate solutions of particle concentrations to be attained, a large number of samples is required. The accuracy is inversely proportional to the square root of the number of samples [28, 35, 2, 4]. Thus, sometimes a computer with one processor fails. Therefore, this thesis resorts to designing a parallel implementations of particle model so as to come up with an efficient and robust model for both pollutants and sediment transport processes.

(II.2) In part (I.1) and (I.2) above, a fixed time step size was used in the stochastic differential equations for the simulation of particle transport using realistic data from the Dutch coastal waters. Nevertheless, when using fixed step size in the stochastic differential equations, the accuracy of a numerical solution is not always guaranteed. Fixed time step size implementations for the numerical solutions of SDE have limitations for instance, when the SDE being solved is stiff. This will force the use of very small time step sizes in the simulations [6, 24].

In addition, the use of fixed smaller time step sizes may become unnecessary and might lead to high computations costs. Therefore, this thesis also designs a particle model that uses dynamic variable time step sizes to get the accurate solution at minimum costs. The implementation of adaptive time step sizes requires the use

> of schemes of higher strong order of convergence [6]. Therefore, in this thesis, two explicit strong order schemes are embedded in the implementation of dynamic variable time step sizes. This makes it possible to compute an error efficiently at each time step. The improvement of the higher order in stochastic schemes is mainly for the correction of the diffusion part [34]. To achieve this, a diffusion coefficient function that varies with the locations of the domain is introduced in the particle model. In this thesis the diffusion coefficient function is modelled in such a way it is constant when the particle is far from the boundary and it goes to zero when the particle approaches the boundary. This way it is guaranteed that the particle will not cross the boundary. Again to further speed up the computations, a parallel version of the particle model with variable time step sizes is implemented. The speedup and efficiency are measured on a multi-computer cluster.

This thesis uses the Lagrangian particle model approach. By designing an efficient particle model for transport of materials in coastal waters, we hope to contribute to solving the transport problems in shallow waters.

1.2 Outline of the thesis

Following this general introduction, chapter 2 gives a detailed account of the necessary theory of stochastic differential equations(SDEs) which is going to be used in this research. The theory of SDEs is described here and the connection between the PDEs and the SDEs is described too.

In chapter 3, a new driving force known as coloured noise is introduced to drive the SDEs. Unlike the classical Brownian motions which are usually used as driving forces whose increments are independent, the coloured noise forces are correlated in time for time scales that are very short [42, 32]. By using coloured noise forces it becomes possible to predict accurately the spreading of a cloud of pollutants shortly after its deployment. The developed general particle model that is driven by the coloured noise is called Random flight model (RFM). It is eventually applied to the Wadden sea for prediction of the dispersion of pollutants. Besides, the RFM turns out to behave the same manner as the Brownian motions when the simulation period is long enough, therefore it is only recommended for short term scale. The results were presented in the conference [8] and the extended paper is submitted to the journal.

In chapter 4, the two-dimensional SDEs for simulating sediment transport is developed. We use probabilistic concepts to incorporate in the model the extra equations for sedimentation and suspension of sediment particles. This is followed by showing that the particle model is consistent with the well-known two dimensional Eulerian transport model with the source and sink terms included. The fact that the particle model is relatively easy to implement [28, 4, 56], in this chapter, the parameters from Eulerian transport model are adapted in the numerical implementation of the particle model.

The North sea provides the real flow data that are used as inputs in the particle transport model. These real data of water velocities and water depth are computed by the two-dimension hydrodynamic model known as WAQUA [49], are used as the input in the particle model. This work was published in the Journal [9].

Chapter 5 specifically considers the improvement of the numerical implementation of the particle model and deals with aspect of parallel processing in the particle model. It is well known that when using particle models, a large number of particles is required in order to get accurate results [21, 31, 40]. In that respect, it is required to use a large number of particles in the simulation of sediment transport. However, the computing time usually increases with a number of particles. Consequently the simulation effort is slowed. Fortunately, particles are assumed to be independent of one another. This property favours the application of *parallel computing* [37]. A parallel computer is a set of processors that work cooperatively to solve a computational problem. This definition is broad enough to include parallel supercomputers that have hundreds or thousands of processors, networks of stations, multiple processor workstations, and embedded systems [22]. Thus, this chapter is wholly devoted to describing and implementing several experiments of parallel simulation of sediment transport. Furthermore, the speedups, efficiency and load balance are measured on a Beowulf cluster. The results were presented in [11], and the extended paper is submitted to the Journal of Parallel and Distributed Computing.

Chapter 6 deals with more realistic aspects of the physical processes and introduces the adaptive scheme that uses variable time step sizes in the particle model. The main focus of this chapter is the derivation and efficient implementation of an adaptive scheme for numerical integration of the underlying sets of SDEs. The size of each integration time step near the boundary is determined using two numerical schemes, one with strong order 1 convergence, the other of strong order 1.5, otherwise we use the aforementioned order 1 scheme and another scheme of strong order 2. By comparing the differences, it is possible to estimate the error and therewith the optimal step size satisfying a given error tolerance criteria. Moreover, the algorithm is developed in such away that it allows for a completely flexible change of the time step size while guaranteeing correct Brownian paths. Part of the results has been presented and published in [12].

Experience has shown that parallel computing works for many real life scientific and engineering applications. Therefore, the particle model parallel computing on multicluster is also designed in this chapter so as to reduce the computing time. The software implementation uses the MPI library and allows for parallel processing. By making use of internal synchronisation points it allows for snap shots and particle counts to be made at given times, despite the inherent asynchronicity of the particles with regard to time. Several experiments of parallel computers are carried out and the analysis of the performances such as speedup and efficiency on a DAS2 Beowulf clusters. The SDEs under consideration model transport of pollutants in shallow waters forced by Brownian motion and were used in a realistic simulation carried out in the North-sea waters.

The last chapter of the thesis, Chapter 7, presents the conclusions of this research. Furthermore, it also gives recommendation as well as directions for further research.

Most chapters in this thesis have been previously published as conference /Journal papers. Consequently, a slight overlap between chapters, especially in the introductory sections is inevitable.

Chapter 2

Stochastic modelling of advection-diffusion processes

2.1 Introduction

The discussions about the deterministic models and stochastic models are succinctly described in this chapter. The connection between the advection-diffusion equation and the partial differential equation known as Fokker-Planck equation is discussed. By doing so we have derived the underlying stochastic differential equations (SDEs). The difference between the Itô and Stratonovich calculus is briefly examined as well as the concepts of strong and weak convergence. The numerical schemes such as Euler and Heun as well as the transformation between the Itô and Stratonovich integration approach are discussed [42, 3, 32, 35, 34].

The numerical simulation of the transport of dissolved substances has become an increasingly important tool. It is used for prediction of pollution in shallow waters or in the atmospheric environment, for example. In addition, water quality management and environmental impact assessment of engineering facilities require numerical models. In general, the resolution of such models is based on the deterministic solution of the ADE through the use of Eulerian models [19, 13] and Eulerian Lagrangian models [19, 13, 56].

In ADE, often an Eulerian approach is used, it can discretise the computational domain by either dividing it in uniform or unstructured grids, the simplest versions for this type of approach are well reported [19].

These methods do not make explicit use of stochastic concepts, which can be seen as a disadvantage in the understanding of the underlying physical processes involving randomness.

On the other hand, Lagrangian particle models (LPMs) have been applied to the simulations of pollutants in both underground water [54, 47] and shallow water bodies such as river or estuary [28, 51, 17]. One of the main advantages of the particle tracking methods comes from the direct use of stochastic concepts, by explicitly assuming that the motions of a particle in a water body is a Markov process. Direct use of stochastic concept to analyse numerical formulation can bring a physical meaning to the associated numerical errors [14].

Stochastic differential equations (SDEs) play a paramount role in a wide range of application areas, including biology, chemistry, epidemiology, mechanics, micro-electronics, economics. More detail on SDEs calculus can be found in standard textbooks such as [42, 3, 34, 41, 26].

2.2 Wiener Processes

A Wiener process can be defined as follows:

Definition 1 *Wiener Process*
A Wiener stochastic process $W(t), t \in [0, \infty)$, is said to be Gaussian with the following properties

1. $W(0) = 0$
2. *For* $0 < t_0 < t_1 < t_2 <, \ldots < t_n$, *the increments* $W(t_1) - W(t_0), \ldots, W(t_n) - W(t_{n-1})$ *are independent.*
3. *For arbitrary t and $\Delta t > 0$, $W(t + \Delta t) - W(t)$ has a Gaussian distribution with mean 0 and variance Δt. That is* $\Delta W(t) \sim \mathrm{N}[0, \Delta \mathrm{t}]$.

Standard Wiener process sometimes is called Brownian motion in general has the properties:

$$W(0) = 0, w \cdot p \cdot 1, \quad \mathbb{E}(W(t)) = 0, \quad \mathit{Var}[W(t) - W(s)] = t - s, \qquad \textit{for all} \;\; 0 \leq s \leq t$$

The properties $\mathbb{E}[W(t)(W(s)] = \min(s, t)$ can be used to demonstrate the independence of the Wiener increments. Let us assume that the time interval : $0 \leq t_0 < \ldots < t_{i-1} < t_i < \ldots t_{j-1} < t_j \ldots < t_n$, thus

$$\begin{aligned} \mathbb{E}[(W(t_i) - W(t_{i-1}))(W(t_j) - W(t_{j-1}))] &= \mathbb{E}(W(t_i)W(t_j)) - \mathbb{E}(W(t_i)W(t_{j-1})) \\ &- \mathbb{E}(W(t_{i-1})W(t_j)) + \mathbb{E}(W(t_{i-1})W(t_{j-1})) \\ &= t_i - t_i - t_{i-1} + t_{i-1} = 0. \end{aligned}$$

Where the increments $(W(t_i) - W(t_{i-1}))$ and $(W(t_j) - W(t_{j-1}))$ are independent. More information on this concept is found in [16, 42, 3, 32].

For computation purposes, it is worthy to consider a discretised version of Wiener motion [16], for example, where $W(t)$ is specified at discrete t values. Hence for a fixed time step size, we set $\Delta t = \frac{T}{N}$ for some positive integers N and let W_k denote $W(t_k)$ with $t_k = k\Delta t$. Thus, an initial Wiener process at $t = 0$ is zero, that is, $W(0) = 0$ with probability 1, and condition 2 and 3 in section 2.2, tell us that

$$W_k = W_{k-1} + \Delta W_k, \quad k = 1, 2, \ldots, N,$$

where the increment ΔW_k is an independent random variable of the form $\sqrt{\Delta t}\mathrm{N}[0,1]$. Let us do one simulation of the discretised Wiener motion over $[0,1]$ with $N = 1000$ computed. The result of a typical Brownian motion path is illustrated in Figure 2.1.

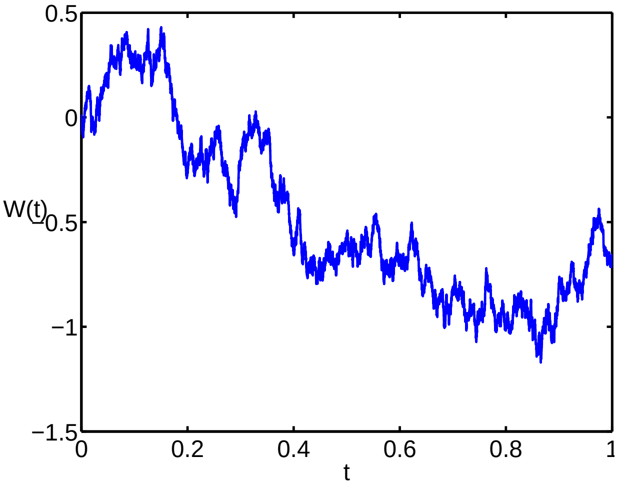


Figure 2.1: A Wiener path

2.3 White noise

White noise $\xi(t)$ is an important example of a stochastic process which is of purely random process. It is Gaussian, stationary process with 0 mean and constant spectral density. It is can be interpreted as the derivative of the Wiener process, but only in generalized sense because Wiener process is nowhere differentiable (see [3, 32, 34, 42], for example):

$$\begin{aligned} \xi(t) &= \frac{dW(t)}{dt} \qquad (2.1) \\ W(t) &= \int_0^t \xi(s)ds. \end{aligned}$$

The variance of a Wiener process satisfies $\mathrm{Var}(W(t)) = t$ and its mean remains at 0. Thus, typical sample paths of the Wiener process attain larger values in magnitude as time increases, and consequently the sample paths of the Wiener process are not of bounded variation. Based on many engineering situations for example, one is led to assume that $\xi(t)$ has at least approximately, these properties

(i) $t_1 \neq t_2 \Rightarrow \xi(t_1)$ and $\xi(t_2)$ are independent.

(ii) $\{\xi(t)\}$ is stationary that is the distribution $\{\xi(t_1+\Delta t), \xi(t_2+\Delta t), \ldots \xi(t_k+\Delta t)\}$ does not depend on Δt. Thus, the white noise is considered to be a stationary Gaussian stochastic process.

(iii) $\mathbb{E}[\xi(t)] = 0$ for all t.

It turns out that there does not exist any reasonable stochastic process satisfying (i) and (ii) thus a process $\xi(t)$ can not have continuous paths [32, 3]. Nevertheless, it is possible to represent $\xi(t)$ as a generalized stochastic process. Note that, the assumption $(i), (ii)$ and (iii) on $\xi(t)$ suggest that $W(t)$ has independent increments with zero mean and has a continuous paths and it is nowhere differentiable ([3, 42]).

As in [3], for example let us consider 1-dimensional white process. White noise has a constant spectral density $f(\lambda)$ on the entire real axis. More detailed information on this concept can be found for example in [42]. If $\mathbb{E}[\xi(s)\xi(t+s)] = C(t)$ is the covariance function of $\xi(t)$, then, the spectral density is given:

$$f(\lambda) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\lambda t} C(t)dt = \frac{c}{2\pi}, \quad \forall \;\; \lambda \in \Re^1. \tag{2.2}$$

The positive constant c without loss of generality can take a value equals 1. Nevertheless, the white noise does not exist in the traditional sense because equation (2.2) is compatible only with the choice $C(t) = \delta(t)$ where δ is the Dirac's delta function [3]. Notably, we would have

$$C(0) = \mathbb{E}[W^2(t)] = \int_{-\infty}^{\infty} f(\lambda)d\lambda = \infty.$$

Since $C(t) = 0$ for $t \neq 0$ the values $\xi(s)$ and $\xi(s+t)$ would be uncorrelated for arbitrary small values of t a fact that explains the name purely random process. Because of the independence of the values at every point, white noise is useful mathematically for describing rapidly fluctuating random phenomena for which the correlation of the state at time s when $|t-s|$ is increasing becomes small very rapidly [3]. White noise $\xi(t)$ can be approximated by an ordinary stationary Gaussian process $X(t)$, for example one with covariance:

$$\mathrm{C(t) = ae^{-b|t|}, \;\; (a>0, b>0)},$$

it can be shown (see Appendix **A**) that such a process has a spectral density.

$$f(\lambda) = \frac{ab}{\pi(b^2+\lambda^2)}.$$

If we now let a and b approach ∞ in such a way that $\frac{a}{b} \to \frac{1}{2}$, we get

$$f(\lambda) \to \frac{1}{2\pi}, \qquad \forall \;\; \lambda \in \Re', \qquad \mathrm{C(t)} = \begin{cases} 0 & t \neq 0, \\ & \\ \infty & t = 0 \end{cases}, \qquad \int_{-\infty}^{\infty} \mathrm{C(t)dt} \to 1,$$

so that $\mathrm{C(t)} \to \delta(\mathrm{t})$, that is, $X(t)$ converges to $\xi(t)$ [3, 32, 35].

2.4 Modelling

2.4.1 Particle Tracking Models

Let us denote the state of the physical system as $x(t)$ in the following ordinary differential equation:

$$\frac{dx}{dt} = f(x,t), \;\; x(t_0) = x(0), \tag{2.3}$$

The differential equation (2.3) can be written as

$$dx = f(x,t)dt, \rightarrow \qquad x(t) = x(0) + \int_{t_0}^{t} f(x(s),s)ds,$$

Thus $x(t) = x(t|x_0,t_0)$ is a solution with initial condition $x(t_0) = x_0$. ODE becomes SDEs when we consider the uncertainty in the model, the SDEs governs the time evolution of the process X satisfies the following equation in some sense.

$$\frac{dX(t)}{dt} \;=\; f(X(t),t) + g(X(t),t)\xi(t), \quad X(t_0) = x_0. \tag{2.4}$$

The stochastic process $\xi(t)$ models uncertainties in the underlying deterministic differential equation. The initial condition is also assumed to be a random variable and independent of $\xi(t)$, essentially, equation (2.4) should be a Markov. This implies that future behaviour of the state of the systems depends only on the present state and not on its past [3, 42].

2.4.2 Stochastic differential equations

The stochastic differential equations(SDEs) are often connected to the partial differential equations (PDEs). The random movement and fluctuations of the displacement of the particle are often described by the Langevin equation [3, 32]) written as a Markovian stochastic differential equation:

$$\frac{d}{dt}X_i(t) = \underbrace{f_i(X_i(t),t)}_{deterministic} + \sum_{j=1}^{2} \underbrace{g_{i,j}(X_i(t),t)\xi_j(t)}_{stochastic}, \qquad i = 1,2. \tag{2.5}$$

Where $(X_1,X_2)^{\mathrm{T}}$ is the position of a particle, $\underline{f}$ is the drift vector and g the noise tensor. It is now convenient to rewrite the stochastic differential equation (2.5) by the aid of Eqn. (2.1):

$$dX_i(t) = f_i(X_i(t),t)dt + \sum_{j=1}^{2} g_{i,j}(X_i(t),t)dW_j(t), \qquad i = 1,2. \tag{2.6}$$

Where the Wiener increment $dW_j(t) = \xi_j(t)dt$ expresses the stochastic influence of the process [32]. The increments of Brownian motions possess these properties:

$$\mathbb{E}[dW_j(t)] \quad = \quad 0$$

$$\mathbb{E}[dW_i(t_1)dW_j(t_2)] = \begin{cases} 0 & \text{if i} \neq \text{j or } \mathrm{t}_1 \neq \mathrm{t}_2 \\ \\ dt_1 & \text{if i} = \text{j and } \mathrm{t}_1 = \mathrm{t}_2. \end{cases}$$

In one dimensional case Eqn. (2.6) can be simplified into the following stochastic differential equation, for $0 \leq t_0 \leq t \leq \infty$.

$$dX(t) \quad = \quad f(X(t),t)dt + g(X(t),t)dW(t), \quad X(t_0) = x_0. \tag{2.7}$$

Or the integral form of (2.7) can be represented as follows:

$$X(t) = X(t_0) + \int_{t_0}^{t} f(X(s),s)ds + \int_{t_0}^{t} g(X(s),s)dW(s). \tag{2.8}$$

To find the solution of the SDE (2.8) careful attention is required unlike in the classical calculus. Note that the first integral on the right hand side of SDEs (2.8) is a Riemann integral while the second integral is called stochastic integral. Therefore, approximating the solution of SDEs (2.8) requires not only the evaluation of the Riemann integral but also the evaluation of a stochastic integral of the type:

$$\int_{t_0}^{t} g(X(s),s)dW(s), \tag{2.9}$$

this integral can be evaluated by using an Itô or a Stratonovich concept depending on the position of the time in which the function $g(X(s),s)$ is evaluated. More discussion will come shortly in Section 2.5. Where $g(X(s),s)$ is the general stochastic process and $W(s)$ is the Wiener process. More detailed information on the interpretations of the stochastic integral can be found e.g., in [25, 41, 42].

2.5 Stochastic integrals

Let us consider an approximation of the stochastic integral (2.9) by the sums:

$$\sum_{i=1}^{N} g(X(\theta_i),\theta_i)\left[W(t_i) - W(t_{i-1})\right]$$

converges in the mean square sense to different values of the integral (see [42], for example), where $\theta_i \in [t_{i-1}, t_i]$.

A value for the integral $\int_{tI}^{tF} W(t)dW(t)$ can be evaluated by approximating $W(t)$ by the function $\psi_n^\kappa(t)$:

$$\psi_n^\kappa(t) = \kappa W(t_k^{(n)}) + (1-\kappa)W(t_{k-1}^{(n)}), \qquad t_{k-1}^{(n)} < t < t_k^{(n)}$$

for any κ, $0 \leq \kappa \leq 1$, and the approximate of the integral by the sums:

$$\int_{tI}^{tF} \psi_n^\kappa(t)dW(t) = \sum_{k=1}^{n} \psi_n^\kappa(t_{k-1}) \left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right]. \tag{2.10}$$

Thus the right-hand side of (2.10) can be presented in the following form:

$$\kappa \sum_{k=1}^{n} W(t_k^{(n)}) \left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right] \\ + (1-\kappa) \sum_{k=1}^{n} W(t_{k-1}^{(n)}) \left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right].$$

By rearranging the terms algebraically,

$$\begin{aligned} \sum_{k=1}^{n} W(t_{k-1}^{(n)}) \left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right] &= \frac{1}{2}W(t_k^{(n)})^2 \\ &- \frac{1}{2}W(t_0^{(n)})^2 - \frac{1}{2}\sum_{k=1}^{n}\left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right]^2 \end{aligned}$$

and

$$\begin{aligned} \sum_{k=1}^{n} W(t_k^{(n)}) \left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right] &= \frac{1}{2}W(t_k^{(n)})^2 \\ &- \frac{1}{2}W(t_0^{(n)})^2 + \frac{1}{2}\sum_{k=1}^{n}\left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right]^2. \end{aligned}$$

It follows that

$$\sum_{k=1}^{n} \psi_n^\kappa(t_{k-1}) \left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right] = \frac{1}{2}W(t_k^{(n)})^2 - \frac{1}{2}W(t_0^{(n)})^2 \\ + \frac{1}{2}(2\kappa - 1)\sum_{k=1}^{n}\left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right]^2. \tag{2.11}$$

The interval $[tI, tF]$ has been partitioned into n equal sub-intervals of length $\frac{tF-tI}{n}$, so for each k the expected valued of $\left[W(t_k^{(n)}) - W(t_{k-1}^{(n)})\right]^2$ is $t_k^{(n)} - t_{k-1}^{(n)}$ which equals $\frac{tF-tI}{n}$. Consequently, the mean-square limit of (2.11) as $\delta_n = t_k^{(n)} - t_{k-1}^{(n)} \to 0$,

$$\int_{tI}^{tF} W(t)dW(t) = \frac{1}{2}\left[W^2(tF) - W^2(tI)\right] + (\kappa - \frac{1}{2})(tF - tI). \tag{2.12}$$

Thus for any choice of κ there is a different result. In particular, if $\kappa = 0$ (which is equivalent to $\theta_i = t_{i-1}$, the left hand side end point of the interval), the integral is known as the Itô integral. This leads to the calculus based on Itô's chain rule [32, 3]. On the other hand if $\kappa = \frac{1}{2}, \theta_i = \frac{1}{2}(t_i + t_{i-1})$, then the resulting integral is the Stratonovich integral. The Stratonovich calculus follows the same rules as for the regular Riemann-Stieltjes calculus.

The evaluation of the Itô integral lead to the following results:

$$\int_{tI}^{tF} W(t)dW(t) \overset{\text{Itô}}{=} \frac{1}{2}\left(W^2(tF) - W^2(tI)\right) - \frac{1}{2}(tF - tI) \tag{2.13}$$

while the following Stratonovich integral leads into

$$\int_{tI}^{tF} W(t)dW(t) \overset{\text{Str}}{=} \frac{1}{2}\left[W^2(tF) - W^2(tI)\right]. \tag{2.14}$$

2.5.1 Itô differential rule

Having defined an Itô integral we will now study the Itô's differential rule. Suppose $X(t)$ is the unique solution of the Itô SDE (2.7). Let $\phi(x, t)$ be a scalar-valued real function, monotone (in x) continuous for $t \in [0, T]$, $x \in (-\infty, \infty)$ for which the derivative $\frac{\partial \phi}{\partial t}, \frac{\partial \phi}{\partial x}, \frac{\partial^2 \phi}{\partial x^2}$ exist and are continuous. With the aid of the Taylor stochastic expansion, it can be shown [26], that the process $\phi(x, t)$ has a SDE given by;

$$d\phi \;=\; \frac{\partial \phi}{\partial t}|_{X(t),t}dt + \frac{\partial \phi}{\partial x}|_{X(t),t}dX(t) + \frac{1}{2}\frac{\partial^2 \phi}{\partial x^2}|_{X(t),t}(dX(t))^2, \tag{2.15}$$

where $(dX(t))^2 = dX(t) \cdot dX(t)$ is computed based on the rules (see e.g.,[42, 3]), $dt \cdot dtdt \cdot dW(t) = dW(t) \cdot dt = 0$ while $(dW(t) \cdot dW(t) = dt$ and with the aid of Eqn. (2.7), we get

$$\begin{aligned} d\phi &= \frac{\partial \phi}{\partial t}dt + \frac{\partial \phi}{\partial x}\left(f(X(t), t)dt + g(X(t), t)dW(t)\right) + \frac{1}{2}g^2(X(t), t)\frac{\partial^2 \phi}{\partial x^2}dt \\ d\phi &= \left(\frac{\partial \phi}{\partial t} + f(X(t), t)\frac{\partial \phi}{\partial x} + \frac{1}{2}g^2(X(t), t)\frac{\partial^2 \phi}{\partial x^2}\right)dt + g(X(t), t)\frac{\partial \phi}{\partial x}dW(t). \end{aligned} \tag{2.16}$$

Itô's differential rule is a direct consequence of the definition of the Itô integral. The extension of the Itô's differential rule into a multi-dimensional case follows similar lines, for example in [3, 32, 42] and references therein where the proof of Itô's differential rule can also be found.

Let the 2-dimensional process $\boldsymbol{X} = (X_1(t), X_2(t))$ have the dynamics given by the Itô SDE (2.6). Its formula is represented by the following equation (see [5], for example):

$$d\phi = \left[\frac{\partial \phi}{\partial t} + \sum_{i=1}^{2} f_i \frac{\partial \phi}{\partial x_i} + \frac{1}{2} \sum_{i=1}^{2} \sum_{j=1}^{2} L_{i,j} \frac{\partial^2 \phi}{\partial x_i \partial x_j} \right] dt + \sum_{i,j=1}^{2} g_{i,j} \frac{\partial \phi}{\partial x_j} dW_j(t), \tag{2.17}$$

where $L_{i,j} = g \cdot g^{\mathrm{T}}$ and let us define the differential operators:

$$\mathcal{A} = \left[\sum_{i=1}^{2} f_i \frac{\partial}{\partial x_i} + \frac{1}{2} \sum_{i=1}^{2} \sum_{j=1}^{2} L_{i,j} \frac{\partial^2}{\partial x_i \partial x_j} \right] \tag{2.18}$$

and

$$\mathcal{A}^0 = \sum_{i,j=1}^{2} g_{i,j} \frac{\partial}{\partial x_j}. \tag{2.19}$$

Consequently, the Itô's formula for 2-dimensional case is written in this form:

$$d\phi = \frac{\partial \phi}{\partial t} + \mathcal{A}\phi dt + \mathcal{A}^0 \phi dW_j(t). \tag{2.20}$$

2.5.2 The relation between Itô and Stratonovich stochastic differential equations

The relation between the Itô and Stratonovich SDEs are given in this section. If a physical process on one hand is described by the Itô SDEs,

$$dX_i(t) \overset{\text{Itô}}{=} f_i(X_i(t), t)dt + \sum_{j=1}^{2} g_{i,j}(X_i(t), t) dW_j(t), \qquad i = 1, 2, \tag{2.21}$$

then the same process can be described also with the Stratonovich equation:

$$\begin{aligned} dX_i(t) \overset{\text{Str}}{=} & \left(f_i(X_i(t), t) - \frac{1}{2} \sum_{k=1}^{2} \sum_{j=1}^{2} g_{k,j}(X_i(t), t) \frac{\partial g_{i,j}(X_i(t), t)}{\partial x_k} \right) dt \\ & + \sum_{j=1}^{2} g_{i,j}(X_i(t), t) dW_j(t), \qquad i = 1, 2. \end{aligned} \tag{2.22}$$

On the other hand, if a physical process is described by the Stratonovich stochastic differential equation:

$$dX_i(t) \overset{\text{Str}}{=} f_i(X_i(t), t)dt + \sum_{j=1}^{2} g_{i,j}(X_i(t), t) dW_j(t), \qquad i = 1, 2, \tag{2.23}$$

then the same process can be described also with the Itô equation: or in general,

$$dX_i(t) \overset{\text{Itô}}{=} \left(f_i(X_i(t),t) + \frac{1}{2}\sum_{k=1}^{2}\sum_{j=1}^{2} g_{k,j}(X_i(t),t)\frac{\partial g_{i,j}(X_i(t),t)}{\partial x_k} \right) dt + \sum_{j=1}^{2} g_{i,j}(X_i(t),t)dW_j(t), \qquad i = 1,2. \quad (2.24)$$

Note that as long as the function $g(X(t),t) = g(t)$ is only time dependent both interpretations will produce the same results. From now in this thesis $X(t) = X_1(t)$, $Y(t) = X_2(t)$, $W(t) = W_1(t)$, $g = g_1$ and $f = f_1$. It is essential to note that the Stratonovich formula agrees well with the classical differential formula unlike in the Itô formula there is no additional term ([34]). Most physics are interpreted into Stratonovich SDE while on one hand, mathematicians prefer the Itô SDE for it is mathematically tractable, for instance the mean of the Itô integral (2.13) is zero (see e.g., [3]).

Example 1 *Let us consider the following geometric Brownian processes that is often applied in finance as models for stochastic prices [42] such that Itô SDE is written as follows;*

$$dX(t) \overset{\text{Itô}}{=} aX(t)dt + bX(t)dW(t), \qquad X_{t_0} = 1, \quad (2.25)$$

with the aid of Itô's differential rule (2.16), and the function $\phi(x,t) = \ln(x), x > 0$ *the following general Itô solution can be obtained [34].*

$$X(t) = e^{(a-\frac{b^2}{2})t+bW(t)}, \qquad X(t_0) = 1, W(0) = 0,$$

where a, b *are positive constants. While*

$$dX(t) \overset{\text{Str}}{=} aX(t)dt + bX(t)dW(t), \qquad X_{t_0} = 1, \quad (2.26)$$

by using equation (2.24), we obtain

$$dX(t) \overset{\text{Itô}}{=} X(t)(a + \frac{b^2}{2})dt + bX(t)dW(t), \qquad X_{t_0} = 1. \quad (2.27)$$

Again now with the aid of the Itô's differential rule (2.16), and the function $\phi(x,t) = \ln(x)$ *the following general Itô solutions can be obtained([34]).*

$$X(t) \overset{\text{Itô}}{=} e^{at+bW(t)}, \qquad X(t_0) = 1, \quad (2.28)$$

While the Stratonovich Eqn. (2.26) has the Stratonovich solution:

$$X(t) \overset{\text{Str}}{=} e^{at+bW(t)}, \qquad X(t_0) = 1, \quad (2.29)$$

which is the same solution.

The evolution of the probability density function of the position of the particle $X(t)$ due to space and time is often described by the Kolmogorov forward equation.

One dimensional Kolmogorov forward equation

Assume that the solution $X(t)$ of (2.7) has a transition density $p(s,y;t,x)$, then p will satisfy the Kolmogorov forward equation

$$\frac{\partial}{\partial t}p(s,y;t,x) = \mathcal{L}^* p(s,y;t,x),\ (t,x) \in (0,T) \times \Re \qquad (2.30)$$
$$p(s,y;t,x) \to \delta_y, \quad \text{as} \quad t \downarrow s.$$

where

$$\mathcal{L}^* = -\frac{\partial}{\partial x}[f(x,t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[g^2(x,t)].$$

This equation is also know as the Fokker-Planck equation.

Proof of the 1-dimensional Itô Fokker-Planck equation

The one dimensional forward Kolmogorov's equation can be derived as follows.

Proof 1 *We assume that $X(t)$ has a transition density and is a solution of the Itô 1 dimensional SDE (2.7)*

$$dX(t) \overset{\text{Itô}}{=} f(X(t),t)dt + g(X(t),t)dW(t), \qquad X(t_0) = x_0.$$

Let us fix two points in time s and T with $s < T$. Now consider an arbitrary "test function" .i.e. an infinite differentiable function $\phi(x,t)$ with compact support in the set $(s,T) \times \Re$ [5]. From the Itô's formula we have

$$\phi(X(T),T) = \phi(X(s),s) + \int_s^T \left(\frac{\partial \phi}{\partial t} + \mathcal{L}\phi\right)(X(t),t)dt + \int_s^T \frac{\partial \phi}{\partial x}(X(t),t)dW(t), \quad (2.31)$$

where

$$\mathcal{L} = \frac{\partial}{\partial x}[f(x,t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[g^2(x,t)].$$

Applying the expectation operator $\mathbb{E}_{s,y}[\cdot]$, to Eqn. (2.31) and using the fact that, because of the compact support, $\phi(x,T) = \phi(x,s) = 0$, we obtain

$$\mathbb{E}_{s,y}\left[\phi(X(T),T) - \phi(X(s),s)\right] = \mathbb{E}_{s,y}\left[\int_s^T \left(\frac{\partial \phi}{\partial t} + \mathcal{L}\phi\right)(X(t),t)\right]dt + \mathbb{E}_{s,y}\left[\int_s^T \frac{\partial \phi}{\partial x}(X(t),t)dW(t)\right].$$

$$0 = \mathbb{E}_{s,y}\left[\int_s^T \left(\frac{\partial \phi}{\partial t} + \mathcal{L}\phi\right)(X(t),t)\right] dt + 0,$$

thus

$$\mathbb{E}_{s,y}\left[\int_s^T \left(\frac{\partial \phi}{\partial t} + \mathcal{L}\phi\right)(X(t),t)\right] dt = 0.$$

Hence,

$$\int_{-\infty}^{\infty}\int_s^T \left(\frac{\partial}{\partial t} + \mathcal{L}\right)\phi(t,x)p(s,y;t,x)dxdt = 0,$$

By using partial integration with respect to (t for $\left(\frac{\partial}{\partial t}\right)$ *and with respect to* x *for* $\mathcal{L}$ *gives*

$$\int p(s,y,t,x)\phi'(x,t)dt = -\int \phi(x,t)\frac{\partial}{\partial t}[p(s,y,t,x)]dt$$

$$\int p(s,y,t,x)f(x,t)\phi'(x,t)dx = -\int \phi(x,t)\frac{\partial}{\partial x}\left(f(x,t)p(s,y,t,x)\right)dx$$

$$\int p(s,y,t,x)g^2(x,t)\phi''(x,t)dx = \int \phi(x,t)\frac{\partial^2}{\partial x^2}\left(g^2(x,t)p(s,y,t,x)\right)dx.$$

Where density function $p(s,y,t,x)$ *approaches zero* $x \to \infty$ *and* $x \to -\infty$ *thus,*

$$\mathbb{E}_{s,y}\left[\int_s^T \left(\frac{\partial \phi}{\partial t} + \mathcal{L}\phi\right)(X(t),t)\right] dt = \int_{-\infty}^{\infty}\int_s^T \phi(x,t)\left\{\frac{-\partial p(s,y;t,x)}{\partial t} + \mathcal{L}^* p(s,y;t,x)\right\} dxdt,$$

$$\int_{-\infty}^{\infty}\int_s^T \phi(x,t)\left(\frac{-\partial}{\partial t} + \mathcal{L}^*\right)p(s,y;t,x)dxdt = 0,$$

where the operator $\mathcal{L}^*$

$$(\mathcal{L}v)^* = -\frac{\partial}{\partial x}[f(x,t)v(x,t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[g^2(x,t)v(x,t)].$$

Since this equation holds for all test functions, the proof is completed.

The corresponding multi-dimensional Itô -FPE can be derived by following similar lines to those of the one dimensional case.

Two dimensional Kolmogorov forward equation

Assume that the solution $(X_1(t), X_2(t))$ of a two-dimensional SDE (2.6) has a transition density $p(s,\boldsymbol{y};t,\boldsymbol{x})$, then p will satisfy the Kolmogorov forward equation (the Fokker-Planck equation)

$$\frac{\partial}{\partial t}p(s,\boldsymbol{y};t,\boldsymbol{x}) = \mathcal{A}^* p(s,\boldsymbol{y};t,\boldsymbol{x}),\ (t,\boldsymbol{x}) \in (0,T)\times\Re^2 \qquad (2.32)$$

$$p(s,\boldsymbol{y};t,\boldsymbol{x}) \to \delta(\boldsymbol{x}-\boldsymbol{y}), \quad \text{as} \quad t \downarrow \boldsymbol{s}.$$

$$\mathcal{A}^* = \left[-\sum_{i=1}^{2} f_i(x,t)\frac{\partial}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2} L_{i,j}(x,t)\frac{\partial^2}{\partial x_i \partial x_j} \right]. \qquad (2.33)$$

For the proof the reader is referred to [26, 5], for example.

To each different process with the coefficient f and L is assigned second order differential operator. The conclusive behaviour of a diffusion process $X(t)$ is that their transition probability $p(s,y,t,x)$ is, under certain assumption uniquely dictated by the *drift vector* and the *diffusion matrix* $L = g \cdot g^{\mathrm{T}}$. This is a surprise since, on the basis of the Itô-SDE (2.6), the functions f and g are obtained only for the first two moments of the $p(s,y,t,x)$ (e.g., see [3]).

2.5.3 Advection-diffusion process

The mathematical description of the transport processes developed in this thesis will be founded on the relationship between the advection-diffusion equations and Kolmogorov's forward equation known as Fokker-Planck equations(FPE)[48]. The transport of substances in shallow waters is often described by the depth averaged advection-diffusion equation:

$$\frac{\partial HC}{\partial t} = -\sum_{i=1}^{2}\frac{\partial}{\partial x_i}(U_i HC) + \sum_{i=1}^{2}\sum_{j=1}^{2}\frac{\partial}{\partial x_i}\left(HD_{i,j}\frac{\partial C}{\partial x_j}\right) + S + Q, \qquad (2.34)$$

where H is water depth; C concentration; U_i is the flow velocity in the x_i direction, $D_{i,j}$ is the dispersion coefficient in x_i-direction due to the component of concentration in x_i gradient-direction, S and Q are terms catering for sinks and sources. The advection-diffusion is widely applied in a variety of engineering problems. For example with $S = Q = 0$, equation (2.34) can be used for prediction of the dispersion of pollutants in shallow waters [28, 4], for example.

In addition to Eqn. (2.34) the boundary conditions are needed. Boundaries in transport models are defined by physical boundaries such as banks, shores, water level and bed or by numerical (open) positioned at for instance tidal inlets. At closed boundary a Neumann boundary condition is often prescribed which excludes mass transfer through such a boundary, mathematically this is denoted by $\frac{\partial C}{\partial \boldsymbol{n}} = 0$, with $\boldsymbol{n}$ the normal vector to the boundary. Dirichlet boundary conditions are often imposed at for instance bottom or open boundaries to prescribe a fixed concentration [19]. In regions far away from the discharge location it is sometimes justified to prescribe $C = 0$ at open boundaries. An equilibrium bed concentration C_e is assumed a bottom boundary condition in sediment transport problems may be $C = C_e$. Initial conditions address a concentration distribution measured at initial state and account for instantaneous discharges of for instance, waste material. More information on the initial and boundary condition can be found in [19, 28]. In many practical situations the analytical solution of equation (2.34) cannot be easily obtained creating the need to numerically approximate it [30, 36].

2.5.4 Consistence of particle model with the advection-diffusion equation

The position of a particle $(X(t), Y(t))$ at time t is assumed to be a Markov process. Thus a 2-dimensional Itô SDEs to describe the position of a particle is given by the following equations;

$$dX(t) \overset{\text{Itô}}{=} [U + (\frac{\partial H}{\partial x}D)/H + \frac{\partial D}{\partial x}]dt + \sqrt{2D}dW_1(t) \tag{2.35}$$

$$dY(t) \overset{\text{Itô}}{=} [V + (\frac{\partial H}{\partial y}D)/H + \frac{\partial D}{\partial y}]dt + \sqrt{2D}dW_2(t). \tag{2.36}$$

$D(x, y)$ stands for a dispersion coefficient and $W(t)$ is a Wiener process with independent increments that are normally distributed with the zero mean and variance Δt.

The probability density function $f(x, y, t)$ for variation in time and space of the positions of particles in two dimension is described by the Fokker-Planck equation. Thus, the probability density function $f(x, y, t),\quad t \geq t_0$ is determined by the following Itô Fokker-Planck equation [28]

$$\frac{\partial f}{\partial t} = -\frac{\partial}{\partial x}[(U + (\frac{\partial H}{\partial x}D)/H + \frac{\partial D}{\partial x})f] - \frac{\partial}{\partial y}[(V + (\frac{\partial H}{\partial y}D)/H + \frac{\partial D}{\partial y})f] + \frac{1}{2}\frac{\partial^2}{\partial x^2}(2Df) + \frac{1}{2}\frac{\partial^2}{\partial y^2}(2Df). \tag{2.37}$$

With the initial condition:

$$f(x, y, t_0) = \delta(x - x_0)\delta(y - y_0) \tag{2.38}$$

due to Itô SDE (2.44). If we relate the particle concentration to the probability density function f:

$$C(x, y, t) = f(x, y, t)/H(x, y, t) \tag{2.39}$$

and substitute Eqn. (2.39) into the Fokker-Planck Eqn. (2.37), the resulting equation is called the advection-diffusion equation. It was shown in [28] that the underlaying SDEs (2.35)-(2.36) is consistent with the 2-dimensional advection-diffusion equation (2.34) where $S = Q = 0$:

$$\frac{\partial(HC)}{\partial t} = -\frac{\partial(HUC)}{\partial x} - \frac{\partial(HVC)}{\partial y} + \frac{\partial}{\partial x}(D\frac{\partial}{\partial x}CH) + \frac{\partial}{\partial y}(D\frac{\partial}{\partial y}CH). \tag{2.40}$$

The coefficient of the Fokker-Planck Eqn. (2.32),.i.e., A_i and $L_{i,j}$ become:

$$\begin{cases} A_i &= f_i \\ L_{i,j} &= \frac{1}{2}\sum_{k=1}^{2} g_{i,k}g_{j,k}, \end{cases} \tag{2.41}$$

with this relation equation (2.32) should be referred to as Itô Fokker-Planck equation. However, the Stratonovich integration rule leads to different values of A_i and $L_{i,j}$ in terms of f_i and $g_{i,j}$. Both rules are correct in the sense that both can be used in the simulation processes (see e.g., [34]).

By matching the Fokker-Planck equation with the advection-diffusion equation, the underlying particle model is shown to be consistent with the ADE. Where,

$$\begin{cases} A_i &= U_i + \sum_{j=1}^{2}(\frac{\partial H}{\partial x_j}D_{i,j})/H + \sum_{j=1}^{2}\frac{\partial D_{i,j}}{\partial x_j} \\ L_{i,j} &= D_{i,j} \\ p &= CH. \end{cases}$$

The drift and the noise components with the local flow velocity and the diffusion component are related by using equation (2.41)

$$\begin{cases} f_i &= U_i + \sum_{j=1}^{2}(\frac{\partial H}{\partial x_j}D_{i,j})/H + \sum_{j=1}^{2}\frac{\partial D_{i,j}}{\partial x_j} \\ \frac{1}{2}\sum_{k=1}^{2} g_{i,k}g_{j,k} &= D_{i,j} \end{cases}$$

The drift component f_i consists of a contribution due to the local flow velocity and a contribution due to a correction term [28, 4, 57, 56]. The hydrodynamic flow model provides the inputs to the particle model. It also provides a diffusion tensor which is isotropic in the horizontal plane, for instance. In this case the off-diagonal elements of the diffusion tensor are set to zero while the diagonal elements equal $D_{11} = D_{22} = D_H$.

The approximation of the numerical solutions of the SDEs (2.35)-(2.36), for example, requires the stochastic numerical schemes. But the derivation of the stochastic schemes can be done in several ways such as an expansion of the stochastic Taylor series or the use of derivative free schemes [34].

2.6 Stochastic Taylor expansion and derivation of stochastic numerical schemes

Numerical schemes can be constructed in several ways. The most common schemes that are often implemented in the approximation of SDEs are based on the stochastic Taylor expansion. The concept is quite similar to that of the deterministic differential equation. The more terms of Taylor series expansion you include in the series, the higher the order of convergence you attain and thus more accurate scheme. Both the Stratonovich and the Itô sense can be derived but let us consider only the expansion of the following Itô SDE:

$$dX(t) \overset{\text{Itô}}{=} f(X(t),t)dt + g(X(t),t)dW(t), \qquad X(t_0) = x_0, \tag{2.42}$$

with the solution such as

$$X(t) \overset{\text{Itô}}{=} X(t_0) + \int_{t_0}^{t} f(X(s),s)ds + \int_{t_0}^{t} g(X(s),s)dW(s). \tag{2.43}$$

Let us assume that v is sufficiently smooth function and by the help of 1-dimensional Itô SDE (2.42), the differential of $v(X(t),t)$ is evaluated and leads to the following Itô's formula:

$$\begin{aligned} d[v(X(t),t)] = \frac{\partial v}{\partial t}|_{X(t),t}dt + f(X(t),t)\frac{\partial v}{\partial x}|_{X(t),t}dt + \frac{1}{2}g^2(X(t),t)\frac{\partial^2 v}{\partial x^2}|_{X(t),t}dt \\ + g(X(t),t)\frac{\partial v}{\partial x}dW(t) + odt. \end{aligned}$$

Consequently;

$$\begin{aligned} dv(X(t),t) = \left[\frac{\partial v}{\partial t} + f(X(t),t)\frac{\partial v}{\partial x} + \frac{1}{2}g^2(X(t),t)\frac{\partial^2 v}{\partial x^2}\right] dt. \\ + g(X(t),t)\frac{\partial v}{\partial x}dW(t) \end{aligned} \tag{2.44}$$

$$dv = \mathcal{L}^0 v dt + \mathcal{L}^1 v dW(t),$$

with the following partial operators;

$$\begin{aligned} \mathcal{L}^0 &= \frac{\partial}{\partial t} + f(X(t),t)\frac{\partial}{\partial x} + \frac{1}{2}g^2(X(t),t)\frac{\partial^2}{\partial x^2} \\ \mathcal{L}^1 &= g(X(t),t)\frac{\partial}{\partial x}. \end{aligned}$$

By applying the differentiation rule to the function $f(X(s),s)$ in equation (2.43), it yields

$$d[f(X(s),s)] = \mathcal{L}^0 f dt + \mathcal{L}^1 f dW(t),$$

whose solution

$$f(X(s),s) = f(X(t_0),t_0) + \int_{t_0}^{s} \mathcal{L}^0 f dz + \int_{t_0}^{s} \mathcal{L}^1 f dW(z). \tag{2.45}$$

Similarly for $g(X(s),s)$ we get

$$d[g(X(s),s)] \quad = \quad \mathcal{L}^0 g dt + \mathcal{L}^1 g dW(t),$$

whose solution is given by

$$g(X(s),s) = g(X(t_0),t_0) + \int_{t_0}^{s} \mathcal{L}^0 g dz + \int_{t_0}^{s} \mathcal{L}^1 g dW(z). \tag{2.46}$$

By substituting equations (2.45) and (2.46) into (2.43), we get

$$\begin{aligned} X(t) = X(t_0) &+ \int_{t_0}^{t} \left\{ f(X(t_0),t_0) + \int_{t_0}^{s} \mathcal{L}^0 f(X(z),z)dz + \int_{t_0}^{s} \mathcal{L}^1 f(X(z),z)dW(z) \right\} ds \\ &+ \int_{t_0}^{t} \left\{ g(X(t_0),t_0) + \int_{t_0}^{s} \mathcal{L}^0 g(X(z),z)dz + \int_{t_0}^{s} \mathcal{L}^1 g(X(z),z)dW(z) \right\} dW(s). \end{aligned}$$

$$\begin{aligned} X(t) = X(t_0) + f(X(t_0),t_0) \int_{t_0}^{t} ds + g(X(t_0),t_0) \int_{t_0}^{t} dW(s) \\ + \int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^0 f dz ds + \int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^1 f dW(z) ds \\ + \int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^0 g dz dW(s) + \int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^1 g dW(z) dW(s). \end{aligned}$$

This leads to a first approximation of the form;

$$\begin{aligned} X(t) = X(t_0) + f(X(t_0),t_0)[t-t_0] + g(X(t_0),t_0)[W(t)-W(t_0)] \\ + \int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^0 f(X(z),z) dz ds + \int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^1 f(X(z),z) dW(z) ds \\ + \int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^0 g(X(z),z) dW(z) ds + \int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^1 g(X(z),z) dW(z) dW(s). \end{aligned} \tag{2.47}$$

$$\begin{aligned} X(t) = X(t_0) + f(X(t_0),t_0)[t-t_0] + g(X(t_0),t_0)[W(t)-W(t_0)] \\ + \mathrm{Errt1}, \end{aligned} \tag{2.48}$$

$$X(t+\Delta t) = X(t) + f(X(t),t)\Delta t_n + g(X(t),t)[W(t+\Delta t) - W(t)],$$

or with $t = n\Delta t$:

$$X_{n+1} \quad = \quad X_n + f(X_n,t_n)\Delta t_n + g(X_n,t_n)\Delta W_{t_n}. \tag{2.49}$$

This is the simplest non-trivial stochastic Taylor expansion. In this derivation we have assumed that the coefficient functions f and g are sufficiently smooth. We again have applied the Itô's rule to the higher order terms of the integrand in (2.47) to obtain schemes with higher order of convergence. Note that the first three terms of (2.47) lead to the stochastic Euler scheme where,

$$\begin{aligned}\text{Errt1} = &\int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^0 f(X(z),z)dzds + \int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^1 f(X(z),z)dW(z)ds\\ &+\int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^0 g(X(z),z)dzdW(s) + \int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^1 g(X(z),z)dW(z)dW(s). \end{aligned} \tag{2.50}$$

Furthermore, we have analysed the next error term (with the lowest order) from equation (2.50)

$$\int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^1 g(X(z),z)dW(z)dW(s). \tag{2.51}$$

The next higher order approximation can be obtained by applying the Itô differentiation formula to the function $\mathcal{L}^1 g$ and get the following

$$d[\mathcal{L}^1 g] \quad = \quad \mathcal{L}^0\mathcal{L}^1 g dz + \mathcal{L}^1\mathcal{L}^1 g dW(z), \tag{2.52}$$

or in the integral form:

$$\mathcal{L}^1 g(X(r),r) = \mathcal{L}^1 g(X(t_0),t_0) + \int_{t_0}^{z}\mathcal{L}^0\mathcal{L}^1 g dr + \int_{t_0}^{z}\mathcal{L}^1\mathcal{L}^1 g dW(r), \tag{2.53}$$

the substitution of Eqn. (2.53) into Eqn. (2.47), yields

$$\begin{aligned}X(t) = X(t_0) &+ f(X(t_0),t_0))(t-t_0) + g(X(t_0),t_0)[W(t)-W(t_0)]\\ &+\mathcal{L}^1 g(X(t_0),t_0)\int_{t_0}^{t}\int_{t_0}^{s}dW(z)dW(s) + \text{Errt2}. \end{aligned} \tag{2.54}$$

$$\begin{aligned}\text{Errt2} = &\int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^0 f(X(z),z)dzds + \int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^1 f(X(z),z)dW(z)ds\\ &+\int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^0 g(X(z),z)dzdW(s) + \int_{t_0}^{t}\int_{t_0}^{s}\int_{t_0}^{z}\mathcal{L}^0\mathcal{L}^1 g(X(r),r)drdW(z)dW(s)\\ &+\int_{t_0}^{t}\int_{t_0}^{s}\int_{t_0}^{z}\mathcal{L}^1\mathcal{L}^1 g(X(r),r)dW(r)dW(z)dW(s). \end{aligned} \tag{2.55}$$

Now let us analyze the error in the fourth term of Eqn. (2.54) as follows;

$$\int_{t_0}^{t}\int_{t_0}^{s}\mathcal{L}^1 g(X(t_0),t_0)dW(z)dW(s) < K_4\int_{t_0}^{t}\int_{t_0}^{s}dW(z)dW(s).$$

This last error term dominates and determines the strong order of the convergence of the Euler scheme, where $K_4 = \mathcal{L}^1 g(X(t_0), t_0)$ is a known constant. The error in this scheme can be analysed:

$$\begin{aligned}
K_4 \int_{t_0}^{t} (W(s) - W(t_0))dW(s) &= K_4\{\int_{t_0}^{t} [W(s) - W(t_0)]dW(s) \\
&= K_4\{\int_{t_0}^{t} W(s)dW(s) - \int_{t_0}^{t} W(t_0)dW(s)\} \\
&= K_4\{\frac{W^2(t) - W_{t_0}^2}{2} - \frac{(t-t_0)}{2} - W(t)W(t_0) + W_{t_0}^2 \\
&= \frac{K_4}{2}\{W^2(t) - W_{t_0}^2 - (t-t_0) - 2W(t)W(t_0) + 2W_{t_0}^2 \\
&= \frac{K_4}{2}\{W^2(t) - 2W(t)W(t_0) + W_{t_0}^2 - (t-t_0)\} \\
&= \frac{K_4}{2}\{[W(t) - W(t_0)]^2 - (t-t_0)\} \\
&= \frac{K_4}{2}[\Delta W^2(t_n) - \Delta t].
\end{aligned}$$

Note that, the stochastic integral (2.51) has a local truncation error of $\mathcal{O}(\Delta t)$. The increments of Wiener process are independent of each other for a time step size Δt regardless of their size. Due to this fact, we need to add the variances of these local errors in order to obtain the variance of the global error instead of adding local errors for every time step. Assuming N steps in the integration, i.e $t = N\Delta t$, we find the variance of the global truncation error is equal to $N\mathcal{O}(\Delta t)^2 = t\mathcal{O}(\Delta t)$. From this we conclude that the stochastic integral (2.51) has a global truncation error of $\mathcal{O}(\Delta t^{\frac{1}{2}})$. Whereas the Milstein scheme has $\mathcal{O}(\Delta t)$ in the strong sense for scalar equations. For vector systems for Milstein scheme has $\mathcal{O}(\Delta t^{\frac{1}{2}})$. Nevertheless, in the weak sense the Milstein scheme has the same order of convergence as that of the Euler scheme.

The 1 dimensional Milstein scheme can be written as follows;

$$\begin{aligned}
X(t) = X(t_0) &+ f(X(t_0), t_0)\Delta t_0 + g(X(t_0), t_0)\Delta W(t_0) \\
&+ \frac{1}{2} g(X(t_0), t_0)\frac{\partial g}{\partial x}\left\{[W(t) - W(t_0)]^2 - (t-t_0)\right\} + \mathrm{Errt2}. \quad (2.56)
\end{aligned}$$

Thus, the Milstein scheme for scalar stochastic differential equation which has higher strong order than that of Euler scheme has been obtained:

$$\begin{aligned}
X_{n+1} = X_n &+ f(X_n, t_n)\Delta t + g(X_n, t_n)\Delta W_n \\
&+ \frac{1}{2} g(X_n, t_n)\frac{\partial g}{\partial x}[\Delta W^2(t_n) - \Delta t]. \quad (2.57)
\end{aligned}$$

More analysis of the error terms in Eqn. (2.55) can heuristically lead to higher order of convergence of the numerical schemes. For example the following error term leads to;

$$\int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^0 f(X(z), z)dzds < K_1 \int_{t_0}^{t}\int_{t_0}^{s} dzds = \mathcal{O}(\Delta t^2),$$

where K_1 is a constant. This deterministic error term introduces a local error of $\mathcal{O}(\Delta t^2)$ and as a consequence, a global error of $\mathcal{O}(\Delta t)$. Next, consider the following;

$$\int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^1 f(X(z),z)dW(z)ds < K_2 \int_{t_0}^{t}\int_{t_0}^{s} dW(z)ds = \mathcal{O}(\Delta(W(t_n)\Delta t) = (\Delta t^{1.5}),$$

$$\begin{aligned}\int_{t_0}^{t}\int_{t_0}^{s} \mathcal{L}^0 g(X(z),z)dzdW(s) &< K_3 \int_{t_0}^{t}\int_{t_0} dW(z)ds\\ &= \mathcal{O}(\Delta t\Delta W(t_n)) = \mathcal{O}(\Delta t^{1.5}).\end{aligned}$$

Those two stochastic terms introduce a strong local error of $\mathcal{O}(\Delta t^{1.5})$ and as a consequence, a strong global error of $\mathcal{O}(\Delta t)$.

One can keep on expanding the next multiple Itô integrals to an arbitrary higher order. Each time, the remainder will involve the next set of multiple Itô integrals with non-constant integrand. Some of these integrals can be solved analytically. Take the following examples whose solutions are well known and are obtained in for example [42]:

$$\begin{aligned}\int_{t_0}^{t} ds &= t - t_0\\ \int_{t_0}^{t}\int_{t_0}^{s} dzds &= \frac{1}{2}(t-t_0)\\ \int_{t_0}^{t}\int_{t_0}^{s} dW(z)dW(s) &= \frac{1}{2}\{[W(t)-W(t_0)]^2 - (t-t_0)\}.\end{aligned}$$

However, it is not possible to derive an analytical expression for most of the stochastic integrals. Consequently these terms also have to be evaluated numerically. Generally, higher order numerical methods that are based on stochastic Taylor expansion are therefore not very useful in practice. The exception here is that in some applications based on some specific nature of the functions f and g, terms in stochastic the Taylor expansion may drop out, in this way, a higher-order scheme can be obtained. Having discussed the Taylor stochastic expansion. It is now the right time to discuss various numerical schemes in the following section.

2.7 Numerical schemes

The numerical schemes which can be used respectively for Itô and Stratonovich SDEs are given by the equation (2.21) and (2.23). Therefore, in order to approximate the right solution, care must be taken with the selection of the numerical scheme [35, 41, 42]. The schemes discussed in this section can only be applied to either Itô or Stratonovich SDEs. Because transformation rules between these two interpretations exist, the selection of numerical scheme is unrestricted, as long as the model is given in the right interpretation, or is transformed likewise.

2.7.1 Euler scheme

The Euler scheme is a result of the stochastic Taylor expansion. The basic Euler scheme for scalar SDEs is derived from the Itô stochastic differential equation (2.42) (see Eqn. (2.49 in section 2.6) for 1 -dimension Euler scheme):

$$X_{n+1} = X_n + f(X_n, t_n)\Delta t_n + g(X_n, t_n)\Delta W(t_n) \tag{2.58}$$

If $Y(t) = X_2(t)$, the 2-dimensional Itô SDE with 2-dimensional Brownian process is written:

$$\begin{aligned} X_{n+1} &= X_n + f(X_n, t_n)\Delta t_n + g(X_n, t_n)\Delta W_1(t_n) \\ Y_{n+1} &= Y_n + f(Y_n, t_n)\Delta t_n + g(Y_n, t_n)\Delta W_2(t_n). \end{aligned}$$

The scheme computes discrete approximations $X_n \approx X(t_n)$, at times $t_n = \sum_{l=0}^{n-1} \Delta t_l$. In practice it is common to use a single pre-chosen value for the step size Δt_l. The stochastic Euler scheme is consistent with the Itô calculus because the noise term in (2.58) approximates the relevant stochastic integral over $[t_n, t_{n+1}]$ by evaluating the integrand at lower end point, thus

$$\int_{t_n}^{t_{n+1}} g(X(s), s)dW(s) \approx g(X_n, t_n)\Delta W(t_n).$$

As in the scalar case, the vector stochastic Euler scheme have strong convergence order $\beta_1 = \frac{1}{2}$ and have weak order $\beta_2 = 1$ (see e.g., [16]). The Euler scheme is applicable only in a sense of the Itô interpretation, therefore, if we are given the Stratonovich SDEs, first we have to transform them into their Itô equivalents.

2.7.2 Milstein Scheme

In Section 2.6, equation (2.57) is what we call the Milstein scheme. This is a more accurate scheme in case of the scalar stochastic differential equation compared to the Euler scheme, for example.

$$\begin{aligned} X_{n+1} = X_n + f(X_n, t_n)\Delta t + g(X_n, t_n)\Delta W_n& \\ + \frac{1}{2} g(X_n, t_n)\frac{\partial g}{\partial x}(\Delta W^2(t_n) - \Delta t).& \end{aligned} \tag{2.59}$$

Milstein scheme is $\mathcal{O}(\Delta t)$ i.e., $\beta_1 = 1$ in the strong sense for scalar equations. For vector systems it is generally only $\mathcal{O}(\Delta t)^{\frac{1}{2}}$, i.e., $\beta_1 = \frac{1}{2}$. In the weak sense the Milstein scheme has the same order of convergence as that of the Euler scheme. The 2-dimensional Milstein scheme can be written as follows;

$$\begin{aligned} X_{n+1} &= X_n + f(X_n, t_n)\Delta t + g(X_n, t_n)\Delta W_n + \frac{1}{2} g(X_n, t_n)\frac{\partial g}{\partial x}(\Delta W^2(t_n) - \Delta t) \\ Y_{n+1} &= Y_n + f(Y_n, t_n)\Delta t + g(Y_n, t_n)\Delta W_n + \frac{1}{2} g(Y_n, t_n)\frac{\partial g}{\partial y}(\Delta W^2(t_n) - \Delta t). \end{aligned}$$

Note that the partial derivative of the stochastic g must be available. Furthermore, similar lines with minor changes can be followed in order to derive the Stratonovich schemes.

Note that, the development of higher-order schemes based on Taylor expansions require more and more derivatives of the coefficient functions f or g. It is also necessary to deal with multiple stochastic integrals (see section 2.6), which cannot be calculated exactly anymore, and thus require numerical approximations for those integrals. However, the alternatives is the use of derivative free explicit schemes to avoid the derivative of the drift f and the diffusion terms. Examples of such schemes are Heun and Runge-kutta, more of their information can be found in [35, 33, 34, 41].

2.7.3 Heun Scheme

The Heun scheme evaluates both the f and g functions at the current point as well as at the estimated succeeding point, and the results of both functions are averaged to get the definite rate of change. Using these rates, an improved approximation of the next point is made.

$$\begin{aligned}
X_{n+1}^* &= X_n + f(X_n, t_n)\Delta t + g(X_n, t_n)\Delta W(t_n) \\
X_{n+1} &= X_n + \frac{1}{2}\{f(X_n, t_n) + f(X_{n+1}^*, t_{n+1})\}\Delta t + \frac{1}{2}\{g(X_n, t_n) + g(X_{n+1}^*, t_{n+1})\}\Delta W(t_n) \\
Y_{n+1}^* &= Y_n + f(Y_n, t_n)\Delta t + g(Y_n, t_n)\Delta W(t_n) \\
Y_{n+1} &= Y_n + \frac{1}{2}\{f(Y_n, t_n) + f(Y_{n+1}^*, t_{n+1})\}\Delta t + \frac{1}{2}\{g(Y_n, t_n) + g(Y_{n+1}^*, t_{n+1})\}\Delta W(t_n).
\end{aligned}$$

This scheme can only be used for SDEs formulated in the Stratonovich sense. Stratonovich SDE use Heun scheme, it has order $\mathcal{O}((\Delta t)^1)$ i.e., $\beta = 1$ in the strong sense and $\mathcal{O}((\Delta t)^1)$ i.e., $\beta = 1$ [35, 16, 50].

In the numerical methods, there are two ways of measuring accuracy, namely strong convergence and weak convergence. Therefore, in the following section we define the two concepts as is discussed in(e.g., [35, 16, 50, 41].

2.7.4 Strong convergence

For strong convergence, we require an instance of the stochastic process to match the exact solution of the process which is driven by the same random function as closely as possible.

Definition 2 *Strong order of Convergence*
Under suitable conditions of the SDE, for a fixed time T, the strong order of convergence is β_1 if there exist a positive constant K and a positive constant Δ such that $T = N\Delta t$:

$$\begin{aligned}
\mathbb{E}\{|X(T) - X(N)|\} &\leq K(\Delta t)^{\beta_1} \qquad (2.60) \\
&= \mathcal{O}(\Delta t^{\beta_1}),
\end{aligned}$$

for all $0 < \Delta t < \Delta$,

where $X(T)$ is the exact solution and $X(N)$ the approximated solution. The strong concept measures the rate at which the mean of the error decays as $\Delta t \to 0$. But when one is interested in the distribution of the random process such as those of $X(t)$, leads to a less demand. This leads to the concept of the weak convergence.

2.7.5 Weak convergence

Strong convergence is often too strict, and in many cases we can attenuate our demands and require only weak convergence. For this type of convergence it is not necessary for the tracks to closely match the exact ones, as long as the characteristics of the stochastic state vector remain the same as those found for the exact solution.

Definition 3 *Weak order of Convergence*
The weak order of convergence is β_2 if there exist a positive constant K and a positive constant Δ such that for a fixed time $T = N\Delta t$:

$$\begin{aligned} |\{\mathbb{E}h(X(T),T)\} - \mathbb{E}\{h(X(N),N)\}| &\leq K(\Delta t)^{\beta_2} \\ &= \mathcal{O}(\Delta t^{\beta_2}), \end{aligned} \tag{2.61}$$

for all $0 < \Delta t < \Delta$ and for each functions h with polynomial growth.

The errors in (2.60) and (2.61) are global discretisation errors, and the largest possible values of β_1 and β_2 give the corresponding strong and weak orders, respectively, of the scheme. The result in Figure 2.2(a)–(b) is obtained by solving the Itô SDE (2.25) for

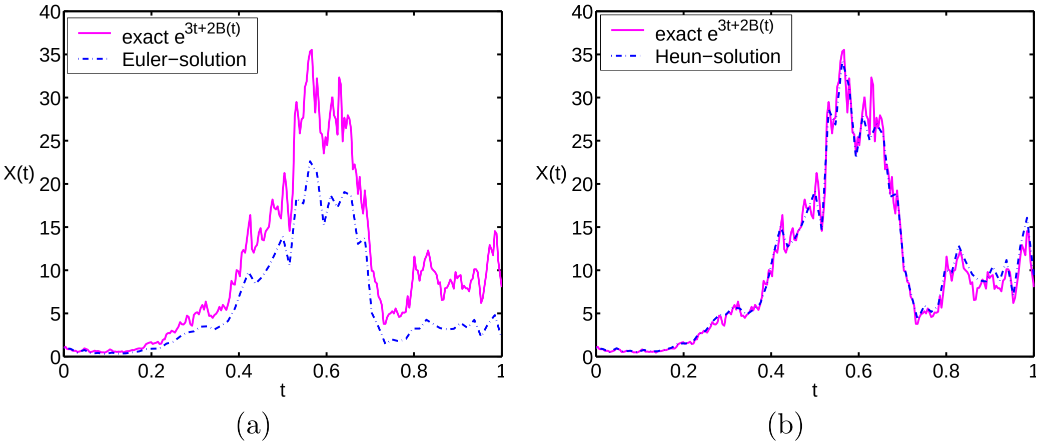
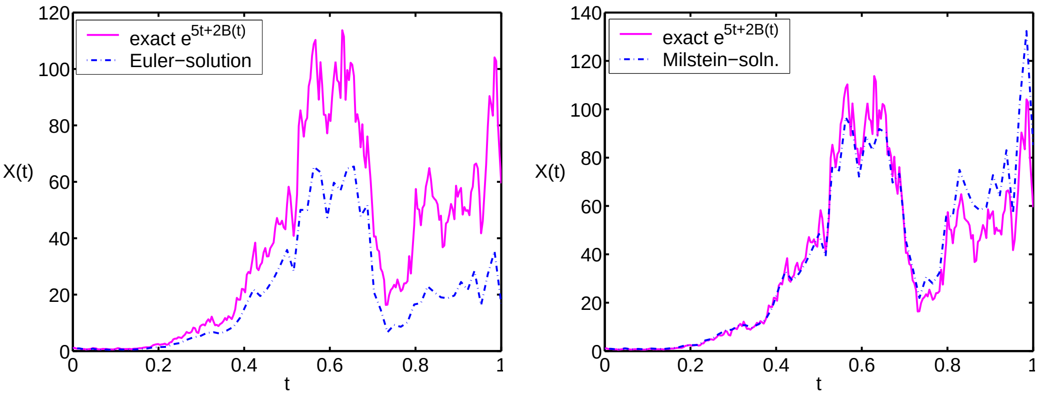
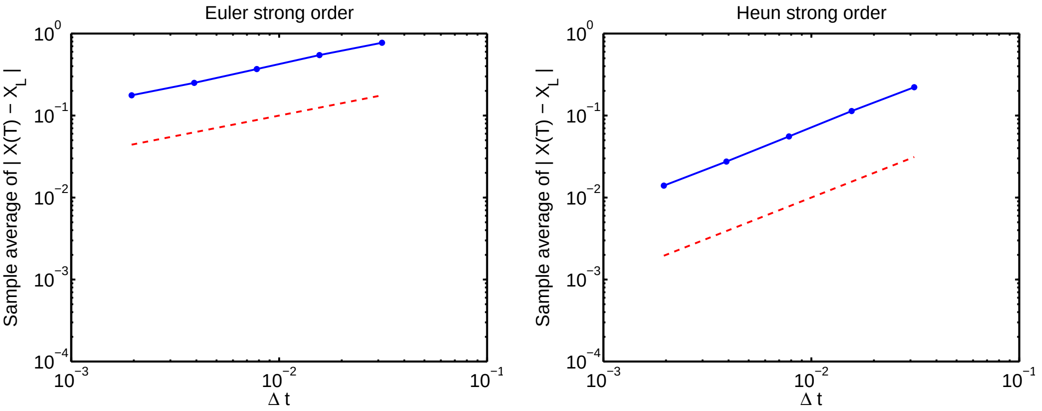


Figure 2.2: Strong solution of an Itô SDE (2.25) for $a = 5, b = 2$, $\Delta t = 2^{-8}$ by Euler scheme (b) of its corresponding Stratonovich SDEs (2.26) for $a = 3, b = 2$, $\Delta t = 2^{-8}$ by Heun scheme.



(a) (b)

Figure 2.3: Strong solution of (a) an Itô SDEs (2.25) for $a = 7, b = 2$, $\Delta t = 2^{-8}$ using Euler scheme (b) of Itô SDE(2.25) for $a = 7, b = 2$, $\Delta t = 2^{-8}$ due to the Milstein scheme.



(a) (b)

Figure 2.4: (a) Results of Itô SDE (2.25) for $a = 2, b = 1$ due to Euler's 0.5 strong order (b) Strat.SDE (2.26) $a = 1.5, b = 1$ due to Heun 1 strong order.

$a = 5, b = 2$ using Euler and the Heun scheme for $a = 3, b = 2$ to solve the Stratonovich SDE (2.26) respectively. Again the Itô SDE (2.25) for $a = 7, b = 2$, was solved by Euler and Milstein scheme respectively see Figure 2.3(a) -(b). While in Figure 2.4 (a)-(b), we have computed respectively the discretised Brownian paths over$[0, 1]$ with $dt = 2^{-9}$. For each path, Euler and Heun schemes are applied in respectively to the Itô SDE (2.25) and Stratonovich SDE (2.26) with 5 different step size: $\Delta t = 2^{p-1}dt$ for $1 \leq p \leq 5$. The end point error in sth sample paths for the pth step size is stored in $Xerr(s, p)$ and XHerr to form a vector of 1000×5. The mean function is used to find the average over all sample paths. This produces a 1×5 array. Hence the pth element of mean(Xerr) and that of mean (XHerr) is an approximation to $e^{\text{strong}}_{\Delta t} = 2^{p-1}dt$. So that the power relation

$$\log e^{\text{strong}}_{\Delta t} = \log C + q \log \Delta t.$$

By using the least square fit method for $\log C$ and q is computed and for both schemes and the values for Euler scheme are $q = 0.5384$ and residue =0.0266. While the values for the Heun scheme are $q = 1.0017$, residue =0.0245. Confirming that Euler scheme has a strong order of $\frac{1}{2}$ and Heun scheme has a strong order of 1. Thus, the choice of the numerical scheme is emphasized here because it is an important consideration, even apart from the order of convergence of the scheme. Sometimes it may be required to transform the original stochastic differential into one of the other type in order to be able to use a specific scheme so as to avoid solving numerically a different physical concept. Any interested reader on the rigorous analysis of the accuracy of the numerical schemes is referred to (e.g., [50, 35, 41]).

In chapter 3 we shall introduce the use of correlated noise called coloured noise to drive the general linear stochastic differential equations and apply it in the simulation of dispersion of pollutants in coastal waters.

Chapter 3

Random flight model for transport of pollutants

Overview[1]: For simulating the dispersion of contaminants in shallow waters, advection-diffusion and tradition particle model(TPM) can be used. The traditional particle model is stochastic in nature and often uses Wiener process as a driving noise. It can be shown to be consistent with the well-known advection-diffusion equation (ADE). A short coming of the ADE and the TPM is the fact that both models describes the dispersion of contaminants correctly if a certain time is passed after the moment of release of a cloud of the contaminants. In the TPM this is caused by the Wiener process which has independent increments. Therefore, this chapter presents a random flight model in which the increments are assumed to be exponentially correlated in time. In this way, we model correctly the behaviour of the model shortly after the release of pollutants. The general random flight model developed in chapter is forced by a coloured noise process representing the short-term correlated turbulent fluid flow velocity of the particles. We also show that for long-term simulations both the traditional particle and the random flight models are consistent with the advection-diffusion equation. In this chapter both TPM and RFM are applied to a real life pollution problem in the Dutch coastal waters.

3.1 Introduction

The rise of environmental awareness in recent years and the need of accurate predictions and improvement of water quality in estuaries, rivers and coastal waters has led to significant development in mathematical modelling of water pollution ([28, 51, 4, 2]). Of the available methods, particle models are gaining popularity [53]. This is because they are easy to implement, can be used for a wide range of applications [2] and because of

---

[1] This chapter was presented in a conference of Coastal waters 2005 in Algarve, Portugal and the extended version of [10], has been submitted for publication to the Journal of Applied Mathematical Modelling (AMM).

the intricacies associated with the finite difference schemes used for approximating the advection-diffusion equations are well known. While the use of particle tracking models can provide valuable information to assist in problem analysis and environmental modelling, the comprehension of the physical processes of the underlying models remain limited. Many particle models for dispersion presented in literatures employ Brownian motion to simulate turbulent diffusion ([28, 21, 31]). Such models assume that particles move according to a simple random walk and consequently have independent increment [32, 53]. It is a well-known fact however that the advection-diffusion equation does only describe dispersion of particles in turbulent fluid flow accurately if the diffusing cloud of contaminants has been in the flow longer than a certain Lagrangian time scale and has spread to cover a distance that is larger in size than the largest scale of the turbulent fluid flow [21]. The Lagrangian time scale (T_L) is a measure of how long it takes before a particle loses memory of its initial turbulent velocity. In reality however there exists short term correlation in time in several hydraulic engineering problems including coastal waters and subsurface diffusion [2]. The traditional advection-diffusion equation is unable to correctly describe the short term correlated behaviour that is found in real turbulent flows at sub-Lagrangian time. In this chapter an extended correlated coloured noise process is employed as the driving force in a random flight model in order to account for diffusion (molecular) processes over short time scales when the eddy(turbulent) diffusion is less than the molecular diffusion. The inclusion of several parameters in the coloured noise process allows for a better match between the auto-covariance of the model and the underlying physical processes. In addition, the random flight model is developed in such a way that the coloured noise process approximates Brownian motion over large time periods. The advection-diffusion equation can also be improved to account for the short term correlation behaviours. The resulting Eulerian model is equivalent to the random flight model for small scale time. In this chapter we choose to numerically implement the random flight model for it is much easier [4] than the numerical implementation of the equivalent Eulerian transport model. Note that the random flight model takes into account the correlation behavior of particle's position at time $t \ll T_L$ and its initial location. The correlation between the two particles is not taken into account because we use a one-particle model [21]. One particle models describe transport of a single particle. By performing experiments with many different particles, one can get an ensemble mean concentration by averaging the results of all individual particles in the experiments. The resulting average concentration value is an important quantity for predictive purposes. The remainder of this chapter is organised as follows, in Section 3.2 we briefly introduce the traditional particle model. Section 3.3 introduces coloured noise processes. Explanations of the techniques for modelling dispersing pollutants are outlined in Section 3.4. The general random flight model is developed in Section 3.4.1. The analysis of spreading properties of a cloud of contaminants such as the variance is done in section 3.5. Several numerical experiments are carried out in Section 3.7. Finally, in Section 3.8 both the traditional particle model and the random flight model are applied to predict the dispersion of pollutants in the Dutch coastal waters.

3.2 The traditional particle model for dispersion in shallow waters

The stochastic mathematical description of a transport process is usually done in terms of stochastic differential equations. In these equations a deterministic and a random component together account for description of the movement of particles. The migration and fluctuation of the position of particles in water at time t, denoted by $(X(t), Y(t))$, are described by means of the following set of stochastic differential equations [28]):

$$dX(t) \overset{\text{Itô}}{=} \left[U + \frac{D}{H}\frac{\partial H}{\partial x} + \frac{\partial D}{\partial x}\right] dt + \sqrt{2D}dW_1(t), \quad X(0) = x_0 \tag{3.1}$$

$$dY(t) \overset{\text{Itô}}{=} \left[V + \frac{D}{H}\frac{\partial H}{\partial y} + \frac{\partial D}{\partial y}\right] dt + \sqrt{2D}dW_2(t), \quad Y(0) = y_0. \tag{3.2}$$

Here D is the dispersion coefficient in m^2/s; $U(x, y, t),\ V(x, y, t)$ are the averaged flow velocities (m/s) in respectively x, y directions; $H(x, y, t)$ is the total depth in m at location (x, y), and $dW(t)$ is a Wiener process with mean $(0, 0)^T$ and $\mathbb{E}[dW_1(t)dW_2(t)^{\mathrm{T}}] = Idt$ where I is a 2×2 identity matrix [34]. Note that the drift part of the particle model (3.1)–(3.2) is not only containing the averaged water horizontal flow velocities but also spatial variations of the diffusion coefficient and the averaged depth. This correction term makes sure that particles are not allowed to be accumulated in regions of low diffusivity as demonstrated by (see e.g., [57, 31, 4]). At closed boundaries particle bending is done by halving the time step sizes until the particle no longer crosses closed boundary. As a result there is no loss of mass through such boundaries. The position $(X(t), Y(t))$ is Markovian and the evolution of its probability density function $f(x, y, t)$, is described by an advection-diffusion type of the partial differential equation known as the Fokker-Planck equation see e.g, [32].

$$\frac{\partial f}{\partial t} = -\frac{\partial}{\partial x}\left(\left[U + \frac{D}{H}\frac{\partial H}{\partial x} + \frac{\partial D}{\partial x}\right] f\right) - \frac{\partial}{\partial y}\left(\left[V + \frac{D}{H}\frac{\partial H}{\partial y} + \frac{\partial D}{\partial y}\right] f\right) + \frac{1}{2}\frac{\partial^2}{\partial x^2}(2Df) + \frac{1}{2}\frac{\partial^2}{\partial y^2}(2Df). \tag{3.3}$$

The initial spreading of a cloud of particles is very small and its distribution can be modelled using a Dirac delta function.

$$f(x, y, t_0) = \delta(x - x_0)\delta(y - y_0). \tag{3.4}$$

This is physically a way of representing a larger concentration or number of particles into an infinitely small space, see [21], for example. The probability density function of particles at a certain location is an exposition of the concentration at that location [28, 21], therefore particle concentration $C(x, y, t)$ can be related to the probability density function $f(x, y, t)$:

$$C(x, y, t) = f(x, y, t)/H(x, y, t). \tag{3.5}$$

By substituting this equation into Fokker-Planck equation (3.3), the following advection-diffusion equation governing the concentration of a conservative pollutants can be expressed as

$$\frac{\partial(HC)}{\partial t} = -\frac{\partial(HUC)}{\partial x} - \frac{\partial(HVC)}{\partial y} + \frac{\partial}{\partial x}\left(DH\frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(DH\frac{\partial C}{\partial y}\right). \tag{3.6}$$

Where U and V are the flow field components in x and y directions, H is the local water depth, D is the dispersion coefficient. An initial condition for the concentration can subsequently be obtained by substituting Eqn. (3.4) into Eqn. (3.5). By interpreting the Fokker-Planck equation (3.3) as an advection-diffusion equation makes the particle model in Eqns. (3.1)–(3.2) to be consistent with the well known advection-diffusion equation (3.6) as shown by [28].

3.3 Coloured noise processes

We introduce coloured noise forces which represent the stochastic velocities of the particles, induced by turbulent fluid flow. We assume that this turbulence is isotropic and that the coloured noise processes are stationary and completely described by their zero mean and Lagrangian auto covariance function.

3.3.1 The scalar exponential coloured noise process

Let the linear stochastic differential equation that models the dynamics of a stochastic velocity of the particle be given by

$$\begin{aligned} du_1(t) &= -\frac{1}{T_L}u_1(t)dt + \alpha_1 dW(t). & (3.7)\\ u_1(t) &= u_0 e^{\frac{-t}{T_L}} + \alpha_1 \int_0^t e^{-\frac{(t-s)}{T_L}} dW(s) & (3.8)\end{aligned}$$

where u_1 is the particle's velocity, $\alpha_1 > 0$ is constant, and T_L is a Lagrangian time scale. For $t > s$ it can be shown [32], that the scalar exponential coloured noise process in Eqn. (3.8) has mean, variance and Lagrangian auto-covariance of respectively,

$$\begin{aligned} \mathbb{E}[u_1(t)] = u_0 e^{\frac{-t}{T_L}},\ \mathrm{Var[u_1(t)]} &= \frac{\alpha_1^2 \mathrm{T_L}}{2}\left(1 - \mathrm{e}^{-\frac{2t}{\mathrm{T_L}}}\right), \\ \mathrm{Cov}[u_1(t), u_1(s)] &= \frac{\alpha_1^2 T_L}{2} e^{-\frac{|t-s|}{T_L}}. & (3.9)\end{aligned}$$

3.3.2 The general vector coloured noise force

The general vector form of a linear stochastic differential equation for coloured noise processes [32, 52] is given by

$$d\mathbf{u}(t) = \mathrm{F}\mathbf{u}(\mathrm{t})\mathrm{dt} + \mathrm{G(t)dW(t)},\ \ \mathrm{d}\mathbf{v}(\mathrm{t}) = \mathrm{F}\mathbf{v}(\mathrm{t})\mathrm{dt} + \mathrm{G(t)dW(t)}. \tag{3.10}$$

Where $\mathbf{u}(t)$ and $\mathbf{v}(t)$ are vectors of length n, F and G are $n \times n$ respectively $n \times m$ matrix functions in time and $\{W(t); t \geq 0\}$ is an m-vector Wiener process with $\mathbb{E}[dW_1(t)dW_2(t)^{\mathrm{T}}] = \mathcal{Q}(t)dt$. In this chapter we extend a special case of the Ornstein-Uhlenbeck process [25, 52] and repeatedly integrate it to obtain the so called coloured noise forcing along the x and y-directions:

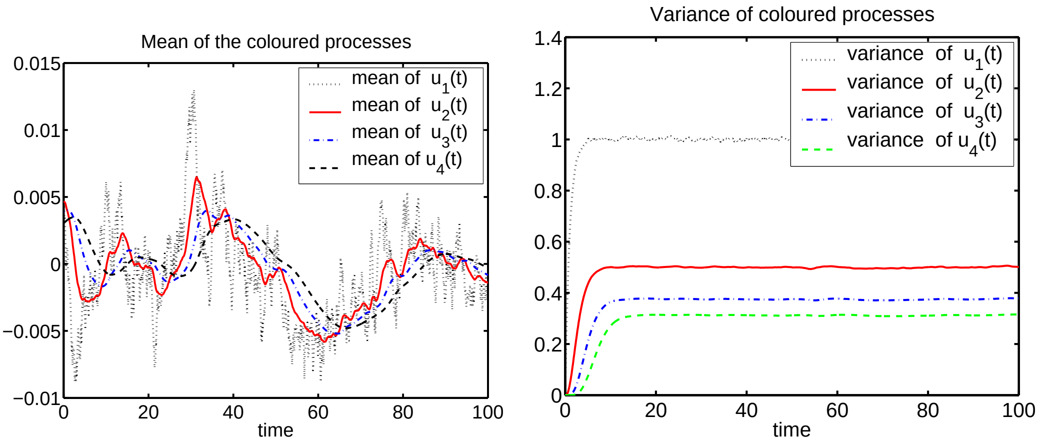
$$\begin{array}{rclrcl} du_1(t) & = & -\frac{1}{T_L}u_1(t)dt + \alpha_1 dW(t), & dv_1(t) & = & -\frac{1}{T_L}v_1(t)dt + \alpha_1 dW(t) \\ du_2(t) & = & -\frac{1}{T_L}u_2(t)dt + \frac{1}{T_L}\alpha_2 u_1(t)dt, & dv_2(t) & = & -\frac{1}{T_L}v_2(t)dt + \frac{1}{T_L}\alpha_2 v_1(t)dt \\ du_3(t) & = & -\frac{1}{T_L}u_3(t)dt + \frac{1}{T_L}\alpha_3 u_2(t)dt, & dv_3(t) & = & -\frac{1}{T_L}v_3(t)dt + \frac{1}{T_L}\alpha_3 v_2(t)dt \\ du_4(t) & = & -\frac{1}{T_L}u_4(t)dt + \frac{1}{T_L}\alpha_4 u_3(t)dt, & dv_4(t) & = & -\frac{1}{T_L}v_4(t)dt + \frac{1}{T_L}\alpha_4 v_3(t)dt \\ \vdots & = & \vdots & \vdots & = & \vdots \\ du_n(t) & = & -\frac{1}{T_L}u_n(t)dt + \frac{1}{T_L}\alpha_n u_{n-1}(t)dt, & dv_n(t) & = & -\frac{1}{T_L}v_n(t)dt + \frac{1}{T_L}\alpha_n v_{n-1}(t)dt. \end{array} \tag{3.11}$$

This way, a Lagrangian auto-covariance of the velocity processes is modelled more accurately and it is now possible to take into account the characteristics of an isotropic homogeneous turbulent fluid flow. The vector Langevin equation (3.11) generates a stationary, zero-mean, correlated Gaussian process denoted by $(u_n(t), v_n(t))$. The Lagrangian time scale $\mathrm{T_L}$ indicates the time over which the process remains significantly correlated. The linear system in Eqn. (3.11), is the same in the Itô and the Stratonovich sense because the diffusion function is not a function of state but only of time. In order to get more accurate results we integrate the stochastic differential equation that is driven by the coloured noise processes by using Heun scheme (see e.g., [41, 51, 34]).

The main purpose of this chapter is the application of coloured noise forcing in the dispersion of a cloud of contaminants so as to improve the short term behaviour of the model while leaving the long term behaviour unchanged. Being the central part of the model, it is important to study the properties of coloured noise processes in more detail. Coloured noise is a Gaussian process and it is well known that these processes can be completely described by their mean and covariance functions [3]. From Eqn. (3.9) also see Figure 3.1 (a), it is easily seen that the mean approaches zero throughout and therefore requires little attention. The covariance, however, depends not only on time but also on the initial values of $u_n(0)$ and $v_n(0)$. This immediately gives rise to the question of how to actually choose or determine these values. Let's consider the covariance matrix of the stationary process $\mathbf{u}$ in the stochastic differential equations of the form (3.10). It is known [32], for example that covariance function P can now be described by

$$\frac{dP}{dt} = FP + PF^T + GQG^T. \tag{3.12}$$

By equating dP/dt to 0, we can find the steady state covariance matrix $\bar{P}$. This matrix can then be used to generate instances of coloured noise processes. This way of generating instances of u vector ensures that the process is sampled at its stationary phase thus removing any artifacts due to a certain choice of start values that would otherwise be used. The auto-covariance is depicted in Figure 3.2. Note that the behaviour of a physical process in this case depends on the parameters in the Lagrangian auto-covariance. Of course the short term diffusion behaviour is controlled by the auto-covariance function. This provides room for the choice of parameters e.g.,$\alpha_1, \alpha_2 \cdots$. The mean, variance and the auto-covariance are not stationary for a finite time t but as $t \rightarrow \infty$, they approach the limiting stationary distribution values as shown in Figures 3.1(a), 3.1(b) and 3.2.



(a) $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1$ (b) $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1$

Figure 3.1: (a) Shows that the mean of the coloured noise processes tends to zero, (b) Shows that the variance of coloured noise processes started from non-stationary state to stationary state.

3.4 Modelling dispersion of pollution processes by coloured forcing noise

Let us consider the following stochastic differential equation,

$$dX(t) \overset{\text{Itô}}{=} f(X(t), t)dt + g(X(t), t)dW(t). \tag{3.13}$$

As stated earlier, the model in Eqn. (3.13) is not physically accurate. The shortcoming of these traditional particle models is caused by the fact that the driving noise in the stochastic differential equations (3.1)–(3.2) and (3.13) are modelled as Brownian processes and as a result, have independent increments for time periods $t \ll T_L$ (see e.g., [32, 3, 2]). Rather, for $t \ll T_L$ it is more realistic to assume that the increments are correlated. Therefore, for short time scales, a coloured force is applied so as to take into consideration the short time correlation behaviour.

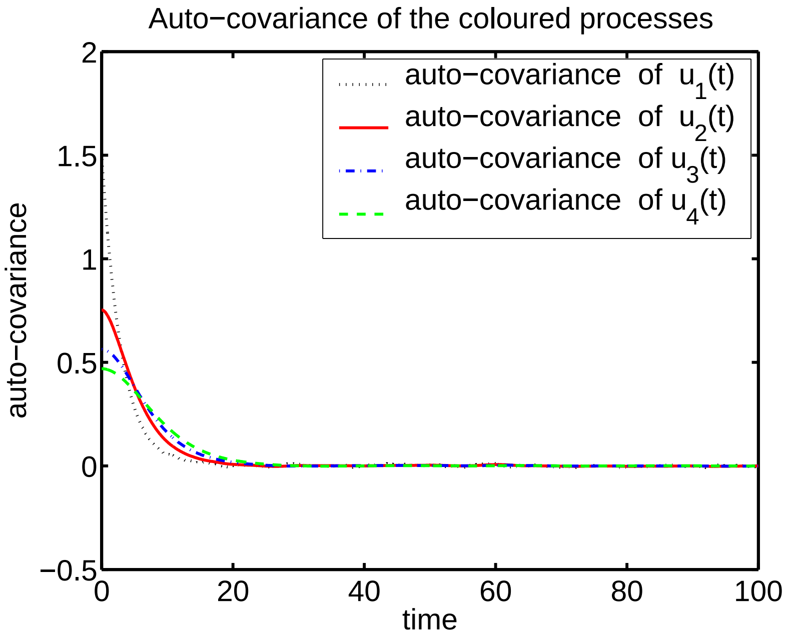


Figure 3.2: The auto-covariance of coloured noise processes started from non-stationary state for $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1$ and tends to zero at stationary state.

3.4.1 Random flight model

To model dispersion in shallow waters we have developed an extension of the random flight model due to [28]. In this extension the coloured noise process is generalised to $(u_n(t),\ v_n(t))$ which represents the velocity of the particle at time t in respectively the x and y directions. This way the Lagrangian auto-covariance processes can be modelled more accurately by taking into account the characteristics of the turbulent fluid flow for $t \ll T_L$. By using the following set of equations the random flight model remains consistent with the advection-diffusion equation (3.6) for long period simulations while modelling accurately the short term correlation of the turbulent fluid flows. In this application we choose $n = 4$:

$$\begin{array}{rclrcl}
du_1(t) &=& -\frac{1}{T_L}u_1(t)dt + \alpha_1 dW(t), & dv_1(t) &=& -\frac{1}{T_L}v_1(t)dt + \alpha_1 dW(t)\\
du_2(t) &=& -\frac{1}{T_L}u_2(t)dt + \frac{1}{T_L}\alpha_2 u_1(t)dt, & dv_2(t) &=& -\frac{1}{T_L}v_2(t)dt + \frac{1}{T_L}\alpha_2 v_1(t)dt\\
du_3(t) &=& -\frac{1}{T_L}u_3(t)dt + \frac{1}{T_L}\alpha_3 u_2(t)dt, & dv_3(t) &=& -\frac{1}{T_L}v_3(t)dt + \frac{1}{T_L}\alpha_3 v_2(t)dt\\
du_4(t) &=& -\frac{1}{T_L}u_4(t)dt + \frac{1}{T_L}\alpha_4 u_3(t)dt, & dv_4(t) &=& -\frac{1}{T_L}v_4(t)dt + \frac{1}{T_L}\alpha_4 v_3(t)dt,
\end{array} \tag{3.14}$$

$$\begin{aligned} dX(t) &= \left[U + \sigma u_4(t) + (\tfrac{\partial H}{\partial x} D)/H + \tfrac{\partial D}{\partial x}\right] dt \\ dY(t) &= \left[V + \sigma v_4(t) + (\tfrac{\partial H}{\partial y} D)/H + \tfrac{\partial D}{\partial y}\right] dt. \end{aligned} \tag{3.15}$$

These systems of vector equations are Markovian we shall refer to this set of equations as the random flight model. The random flight model (3.14)–(3.15) is integrated for many different particles. Note that at the start of the simulation all particles have initial Gaussian velocities $(u_4(0), v_4(0))$ with zero mean and variance that agrees with covariance matrix $\bar{P}$ at a steady state.

3.5 The spreading behaviour of a cloud of contaminants

The characteristics of a spreading cloud of contaminants due to Brownian motion and coloured noise processes are discussed in the following sections.

3.5.1 Long term behaviour of Brownian motion force

Let us, for now assume there is no drift term in Eqn. (3.13) that is, $f(X(t), t) = 0$, and let

$$g(X(t), t) = \sqrt{2D}.$$

It follows that,

$$dX(t) \overset{\text{Itô}}{=} \sqrt{2D} dW(t). \tag{3.16}$$

By applying Theorem 1(see Appendix, 3.10) to Eqn. (3.16), it can be shown that the variance of a cloud of contaminants grows linearly with time:

$$\mathrm{Var[X(t)]} \overset{\text{Itô}}{=} \mathrm{2Dt + constant}. \tag{3.17}$$

3.5.2 Long term spreading behaviour of a cloud of contaminants subject to coloured noise forcing

As discussed in Section 3.3.1 with, for example, $u_1(t)$ from Eqn. (3.8) as the coloured noise forcing and still assuming there is no background flow, the position of a particle at time t is given by

$$dX(t) = \sigma u_1(t) dt, \implies X(t) = X(0) + \sigma \int_0^t u_1(m) dm. \tag{3.18}$$

For simplicity, yet without loss of generality, let $X(0) = u_i(0) = 0$, for $i = 1, 2, \cdots, n$. Now, Eqn. (3.8) leads to $u_1(m) = \alpha_1 \int_0^m e^{-\frac{1}{T_L}(m-k)} dW(k)$, and consequently,

$$X(t) \overset{\text{Itô}}{=} \sigma \alpha_1 T_L \int_0^t (1 - e^{-\frac{1}{T_L}(t-k)}) dW(k). \tag{3.19}$$

Using Theorem 1 (see Appendix, 3.10), it can be shown that the position of a particle at time t is normally distributed with zero mean and variance:

$$\frac{\mathrm{Var[X(t)]}}{t} = \sigma^2\alpha_1^2T_L^2\left[1 - \frac{2T_L}{t}(1 - e^{\frac{-t}{T_L}}) + \frac{T_L}{2t}(1 - e^{\frac{-2t}{T_L}})\right].$$

Thus, a position of a particle observed over a long time span as modelled by the coloured noise process $u_1(t)$ behaves much like the one driven by Brownian motion with variance parameter $\sigma^2\alpha_1^2T_L^2$. Hence, the dispersion coefficient is related to variance parameters $\sigma^2\alpha_1^2T_L^2 = 2D$. We can clarify this by considering Eqn.(3.19), where the second part is $u_1(t)$ itself;

$$X(t) = \sigma T_L\left[\alpha_1 W(t) - u_1(t)\right],\ \text{where } u_1(t) = \alpha_1\int_0^t e^{-\frac{1}{T_L}(t-k)}dW(k).$$

Let us now re-scale the position process in order to better observe the changes over large time spans. Doing so, for $N > 0$, we get

$$X_N(t) = \frac{1}{\sqrt{N}}X(Nt) \ = \sigma T_L\left[\alpha_1\tilde{B}(t) + \frac{1}{\sqrt{N}}u_1(t)\right], \tag{3.20}$$

where $\tilde{B}(t) = \frac{B(Nt)}{\sqrt{N}}$ remains a standard Brownian motion process. For sufficiently large N it becomes clear that Eqn. (3.20) behaves like Brownian motion:

$$X_N(t) \ \approx \sigma\alpha_1T_L\tilde{B}(t).$$

3.5.3 The analysis of short term spreading behaviour of a cloud of contaminants

For scalar coloured noise, it can be shown using Eqn. (3.9) that

$$\mathrm{Cov[u_{t+\tau}u_t]} \ = \ \mathbb{E}[u_{t+\tau}u_t] = \mathbb{E}[v_{t+\tau}v_t] = \frac{1}{2}\alpha_1^2T_Le^{\frac{-|\tau|}{T_L}}. \tag{3.21}$$

Subsequent application of equation (3.21) that is,

$$\mathbb{E}[u_\tau u_s] = \frac{1}{2}\alpha_1^2T_Le^{\frac{-|\tau-s|}{T_L}},$$

yields,

$$\mathrm{Var[X_t]} = \sigma^2\int_0^{\mathrm{t}}\int_0^{\mathrm{t}}\frac{1}{2}\alpha_1^2\mathrm{T_L}\mathrm{e}^{\frac{-(\tau-\mathrm{s})}{\mathrm{T_L}}}\,\mathrm{d}\tau\mathrm{ds}. \tag{3.22}$$

The integration of equation (3.22) can easily be solved by separately considering the regions $\tau < s$ and $\tau > s$, and it can be shown that

$$\begin{aligned}\mathrm{Var[X_t]} \ &= \ \sigma^2\alpha_1^2T_L^3\left(\frac{t^2}{2T_L^2} - \frac{t^3}{6T_L^3}\cdots\right)\\ &= \ \frac{\sigma^2\alpha_1^2T_Lt^2}{2} - \frac{\sigma^2\alpha_1^2t^3}{6} + \cdots.\end{aligned} \tag{3.23}$$

Since we are interested only in the short time analysis, Eqn. (3.23) is considered only for very small values of t in a sense that for $t \ll T_L$ the variance of a cloud of particles shortly after deployment is then given by the following equation:

$$\mathrm{Var[X_t]} = \frac{1}{2}\sigma^2\alpha_1^2\mathrm{T_L t^2}. \tag{3.24}$$

With the constant dispersion coefficient $D = \frac{1}{2}\sigma^2\alpha_1^2 T_L^2$, the variance of the cloud of particles, therefore initially grows with the square of time:

$$\mathrm{Var[X(t)]} = \frac{\mathrm{D}}{\mathrm{T_L}}\mathrm{t}^2. \tag{3.25}$$

3.5.4 The general long term behaviour of a cloud of contaminants due to coloured noise

Let us assume that there is no flow in the model and therefore have

$$dX(t) = \sigma u_1(t)dt \longrightarrow X(t) = \int_0^t \sigma u_1(s)ds, \quad u_1(s) = \alpha_1 \int_0^s e^{-\frac{1}{T_L}(s-k)}dW(k),$$

$$X(t) = \left(\frac{1}{T_L}\right)^0 \sigma\alpha_1 \int_0^t \int_0^{t-k} e^{-\frac{1}{T_L}(s-k)}\frac{(s-k)^0}{0!}dsdW(k), \; X(0) = 0.$$

It can then be shown that

$$u_2(s) \;\; = \tfrac{1}{T_L}\alpha_1\alpha_2 \textstyle\int_0^s e^{-\frac{1}{T_L}(s-k)}(s-k)dW(k), \tag{3.26}$$

where $0 < m < s < t$. Since $k < s$, the position of a particle due to coloured noise force Eqn. (3.26) is given by

$$X(t) = \left(\frac{1}{T_L}\right)^1 \sigma\alpha_1\alpha_2 \int_0^t \int_0^{t-k} e^{-\frac{1}{T_L}(s-k)}\frac{(s-k)^1}{1!}dsdW(k).$$

In general, a position due to $u_n(t)$ force is: $X(t) = \int_0^t \sigma u_n(s)ds$, and it follows that

$$X(t) \overset{\text{Itô}}{=} \left(\frac{1}{T_L}\right)^{n-1} \sigma \prod_{i=1}^n \alpha_i \int_0^t \left[\int_0^{t-k} e^{-\frac{1}{T_L}(s-k)}\frac{(s-k)^{n-1}}{(n-1)!}\right] dsdW(k). \tag{3.27}$$

Careful manipulation using integration by parts of the integral within the square brackets of Eqn. (3.27), yields

$$X(t) \overset{\text{Itô}}{=} (T_L)^n \left(\frac{1}{T_L}\right)^{n-1} \sigma \prod_{i=1}^n \alpha_i \int_0^t [1 + ...]\, dW(k), \quad \text{for} \;\; n \geq 1. \tag{3.28}$$

Finally, with the aid of Theorem 1 (see Appendix, 3.10) and also in [52], the variance of a cloud of contaminants can be computed as described in the sections above. The derivation of velocity $v_n(t)$ of the particle along the y direction proceeds completely analogously. Let us now compute the variance of the general equations for position given by Eqn. (3.28)

$$\mathrm{Var[X(t)]} \quad = \sigma^2 \left(T_L\right)^2 \textstyle\prod_{i=1}^{n} \alpha_i^2 \int_0^t \left[1 - .....\right]^2 dk. \tag{3.29}$$

For $\sigma > 0$, $\alpha_i > 0$, and $T_L > 0$, the process again behaves like a Brownian process with variance parameters $T_L^2 \sigma^2 \prod_{i=1}^{n} \alpha_i^2$ as $t \to \infty$. Thus the appropriate diffusion coefficient from Eqn. (3.17) equals $D = \frac{\sigma^2 T_L^2 \prod_{i=1}^{n} \alpha_i^2}{2}$. This relation is important because it gives a criterion for various choices of parameters α_i, $i = 1, \cdots, n$, $T_L > 0$. In a simulation the constant dispersion coefficient D is often specified whereas σ must be solved in terms of the other parameters.

3.6 The traditional discrete particle model driven by Brownian motion

Analytical solutions of stochastic differential equations do not always exist due to their complexity and nonlinearity. Therefore, stochastic numerical integration schemes are often applied [41, 51, 35]. An example of a numerical scheme is the Euler scheme which, although not optimal in terms of order of convergence, it is easy to implement and requires only $O(\Delta t)$ in the weak sense [34]. Here the time interval $[t_0, T]$ is discretised as $t_0 = 0 < t_1 < t_2 < \cdots < t_{n-1} < t_n = T$, with $\Delta(t_k) = t_{k+1} - t_k$, $\Delta W(t_k) = W(t_{k+1}) - W(t_k)$, for $k = 0, 1, \cdots n$.

$$\bar{X}(t_{k+1}) \quad = \bar{X}(t_k) + \left[U + (\tfrac{\partial H}{\partial x} D)/H + \tfrac{\partial D}{\partial x}\right] \Delta(t_k) + \sqrt{2D} \Delta W_1(t_k) \tag{3.30}$$

$$\bar{Y}(t_{k+1}) \quad = \bar{Y}(t_k) + \left[V + (\tfrac{\partial H}{\partial y} D)/H + \tfrac{\partial D}{\partial y}\right] \Delta(t_k) + \sqrt{2D} \Delta W_2(t_k), \tag{3.31}$$

where $\bar{X}(t_{k+1})$ and $\bar{Y}(t_{k+1})$ are the numerical approximations of the $X(t_{k+1})$ and $Y(t_{k+1})$ positions respectively due to the traditional particle model. The noise increments $\Delta W_1(t_k)$ and $\Delta W_2(t_k)$ are independent and normally distributed $\mathrm{N}\left(0, \Delta(\mathrm{t_k})\right)$ random variables which can be generated using e.g.,pseudo-random number generators [34], for example. The domain information consisting of flow velocities and depth is computed using a hydrodynamic model known as WAQUA (see [49]). The flow averaged fields are only available on grid points of a rectangularly discretised grid and therefore, interpolation methods are usually used to approximate the values at other positions.

3.6.1 Boundaries

One problem with numerical integration of particle positions arises in the vicinity of boundaries. Given the current location, $(X(t), Y(t))$, we may find that the new location, $(X(t + \Delta t), Y(t + \Delta t))$, is on the other side of a boundary, i.e. the particle has crossed a boundary. Depending on the type of boundary this may be physically impossible. We consider two types of boundaries, the first type, closed boundaries, represents boundaries intrinsic to the domain such as banks, sea bed, and coastal lines. The second type of boundaries are open boundaries, which arise from the modeller's decision to artificially limit the domain because particles are not expected to reach any further or simply because no domain information is available at those locations. It is undesirable to have particles cross the first type of boundary, whereas for the second type it is quite natural. Based on this classification, we apply the following rules to particles crossing borders during integration:

- In case an open boundary is crossed by a particle, the particle remains in the sea but is now outside the scope of the model and is therefore removed.

- In case a closed boundary is crossed by a particle during the drift step of integration, the step taken is cancelled and the time step halved until the boundary is no longer crossed. However, because of the halving, say n times, the integration time is reduced to $2^{-n}\Delta t$, leaving a remaining $(1-2^{-n})\Delta t$ integration time, which, at a constant step size, requires at least another $2^n - 1$ steps in order to complete the full time-step Δt. Note that at each of these steps it may be needed to further reduce the step size. This further reduction applies only to the current time step, leaving the step size of following sub-steps unaffected. This method effectively models shear along the coastline.

- If a closed boundary is crossed during the diffusive part of integration, the step size halving procedure described above is maintained with the modification that in addition to the position, the white noise process is also restored to its state prior to the invalidated integration step. The process of halving the time step and continuing integration with the reduced step size is repeated until the full Δt time step has been integrated without crossing a boundary.

3.7 Numerical Experiments

Before applying both the traditional model (3.1)–(3.2) and the random flight model (3.14)–(3.15) to a real life pollution problem. Let us consider a two dimensional channel domain with flow field (see Figure 3.3). In order to compare the spreading behaviour of a cloud of contaminants some experiments using both particle models were carried out. From now onwards in this chapter we shall denote Brownian motion by BM and coloured noise by CN.

We have released $40,000$ particles at the center point $(0,0)$ marked by "X" (see Figure 3.3) also the simulation starts at time $t_0 = 0$. The scattering of a cloud of contaminants due to coloured noise or Brownian motions forces is followed at a specified time steps after release. Generally a large number of particles are used [2] in numerical simulations. The simulations parameters that have been used for simulations in this chapter are summarised in the Table 3.1 (see Appendix 3.10). The channel domain Figure 3.3 is mainly designed to discuss numerical problems that can be encountered

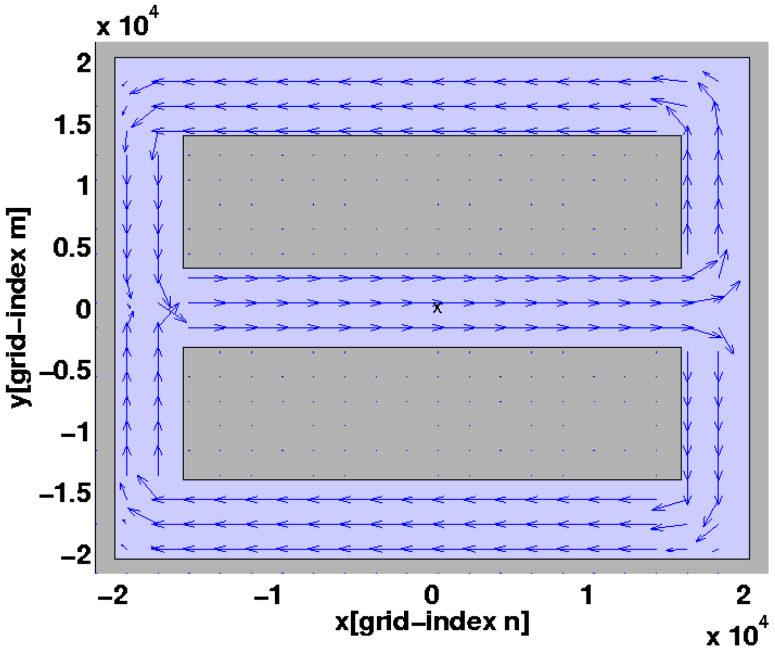
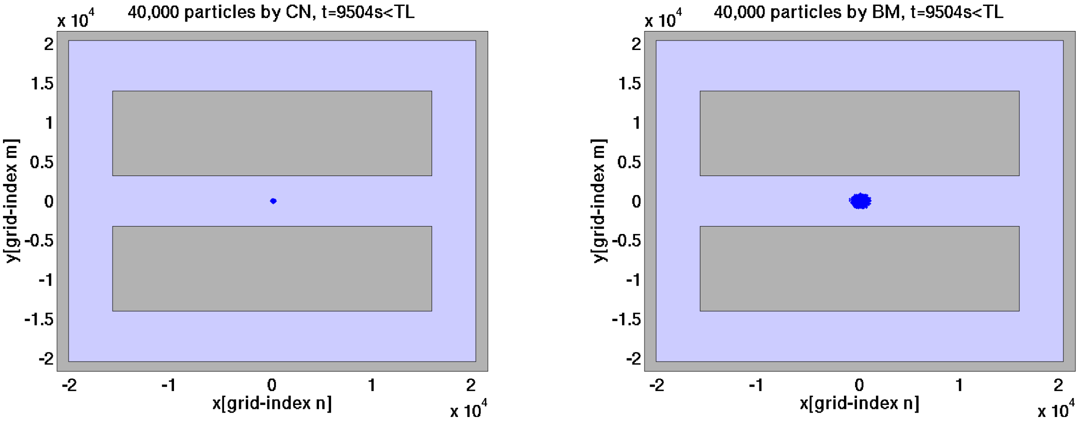


Figure 3.3: Domain showing flow velocity and a point at which particles are released.

in particle models as is discussed in Section 3.6.1. The results in the Figures 3.4(a)-(b), 3.5(a)-(b) and 3.6(a)-(b) and 3.7(a)-(b) respectively show the spreading of a cloud of particles as well as a marked track particle. By using the random flight model the short term spreading is much smaller and moves persistently but slowly when you consider the temporal time scales $t : t \ll T_L$, while it is random and faster when you use Brownian motion. However in the long scale the spreading of a cloud has similar behaviours (see e.g., Figure 3.5(b)). This suggests that if one is interested in the accurate modelling in the short term behavior for $t : t \ll T_L$ then using the traditional particle model is not good, in that case it is better to use the random flight model.

In this work we have considered a series of experiments in a stationary homogeneous turbulent flow with zero mean velocity and the Lagrangian time scale as T_L is introduced in the models. We also carried out an experiment in the empty domain using random flight model as well as the traditional particle model so as to show the differences between the small scale fluctuations and their similarity in the long scale fluctuations. The simulation of the spreading of a cloud of 20,000 particles is tracked in an empty domain and its variance is computed. We have shown that once the particles have been in the flow longer than the time scale T_L, the variance of the spreading cloud grows linearly with time similar to the behaviour of the advection-diffusion equation. Before that time,



(a)

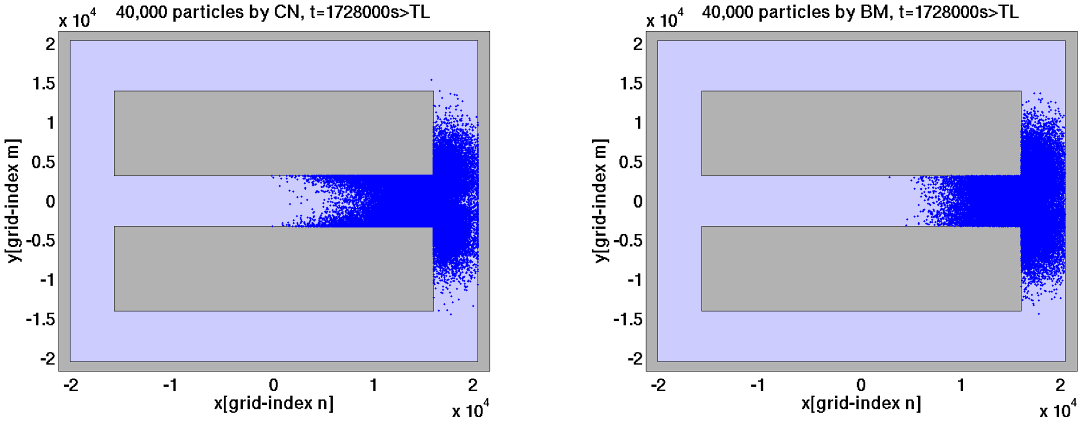
(b)

Figure 3.4: (a) Dispersion of a cloud of $40,000$ *particles released in the idealised channel domain by using Random flight for* $t < T_L$*, (b) Dispersion of a cloud of 40,000 particles released in the idealised channel domain by using Tradition particle model for* $t < T_L$*.*

the variance grows with the square of time (quadratically), creating two different zones of diffusion when using the random flight model, (see Figure 3.8). In Section 3.5.4 we have suggested that for $t \gg TL$ we can define a turbulent mixing coefficient similar to constant dispersion coefficient D such that $D = \frac{\sigma^2 T_L^2 \prod \alpha_i^2}{2}$.

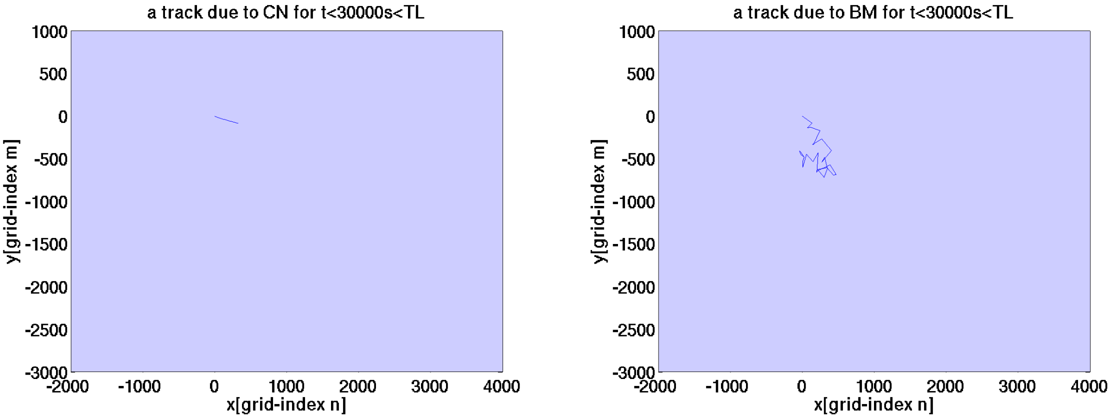
3.8 Application of both the traditional particle and random flight models

For comparison purposes we have applied both models to the simulations of the transport of pollutants in the Dutch coastal waters in and around the Wadden sea. In this chapter we choose a grid cell size of $800m \times 800m$ this grid cell size is sufficiently small to model the dispersive effect of the tidal flow induced by topographic variations [61]. The interpolation methods can be employed to get data at arbitrary positions, we have included few examples of the averaged flow field in the North sea (see Figure 3.9(a)-(b)). The spreading of a cloud of $400,000$ particles is tracked (see Figure 3.10(a)-(b) and Figure 3.11(a)-(b)) where a cloud of contaminants is released from a fixed grid position and tracked by either a Brownian motion or a coloured noise force (see Table 3.1) for the values of the simulation parameters. The tracks of a marked particle in the Wadden sea are plotted in Figures 3.12(a)-(b) and Figures 3.13(a)-(b).



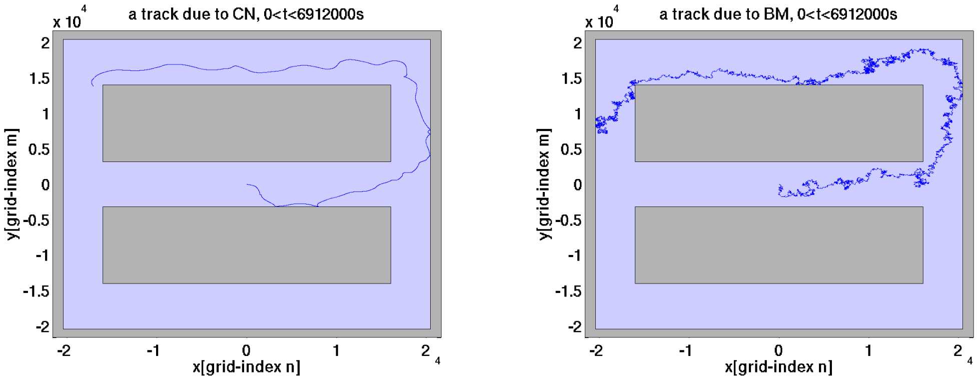
(a) (b)

Figure 3.5: (a) Dispersion of a cloud of 40,000 particles released in the idealised channel domain by using Random flight for $t \gg T_L$, (b) Dispersion of a cloud of 40,000 particles released in the idealised channel domain by using Tradition particle model for $t \gg T_L$.



(a) (b)

Figure 3.6: (a) Tracking of a single marked particle domain by using Random flight for $t < T_L$, (b) Tracking of a single marked particle domain by using Tradition particle model for $t < T_L$.



(a) (b)

Figure 3.7: (a) Tracking of a single marked particle by using Random flight for $t \gg T_L$, (b) Tracking of a single marked particle by using Tradition particle model for $t \gg T_L$.

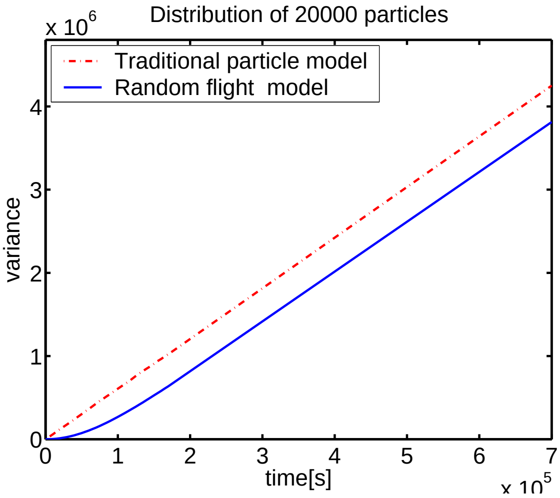
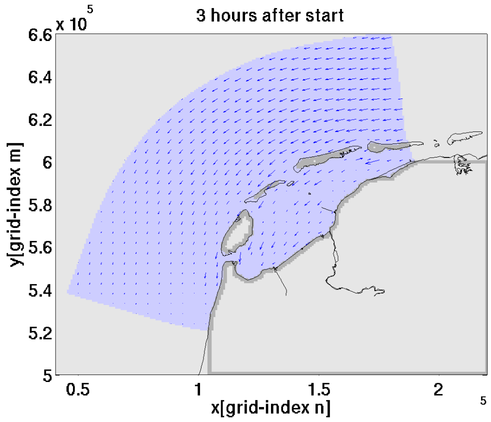
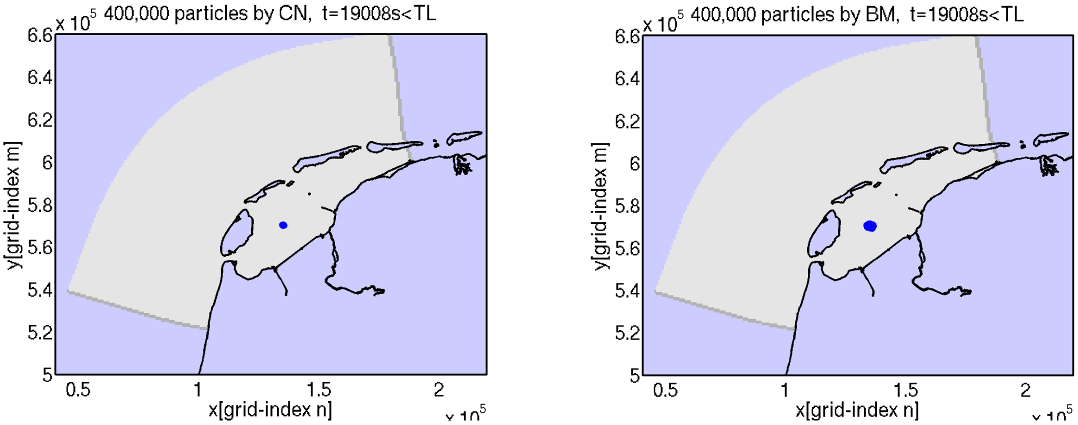


Figure 3.8: Distribution of a cloud of 20,000 particles in the idealized empty domain showing the two zones.



(a) (b)

Figure 3.9: (a) Example of tidally averaged flow fields 3 hours after, (b) Example of tidally averaged flow fields 6 hours after.



(a) (b)

Figure 3.10: (a) Dispersion of a cloud of 400, 000 *particles released at* $(x, y) = (135km, 570km)$ *in the Wadden sea by using Random flight for* $t < T_L$, *(b) Dispersion of a cloud of 400,000 particles released at* $(x, y) = (135km, 570km)$ *in the Wadden sea by using Tradition particle model for* $t < T_L$.

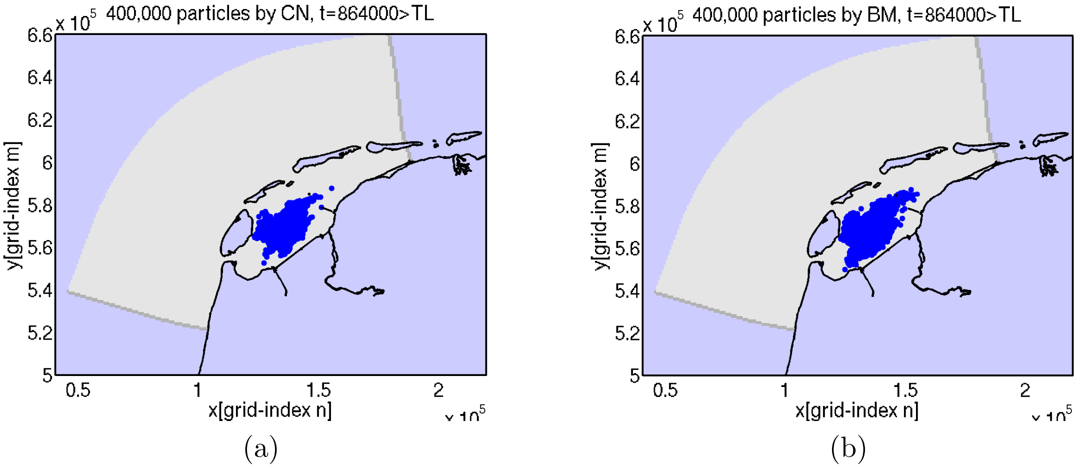


Figure 3.11: (a) Dispersion of a cloud of $400,000$ *particles released at* $(x, y) = (135km, 570km)$ *in the Wadden sea by using Random flight for* $t \gg T_L$*, (b) Dispersion of a cloud of 400,000 particles released at* $(x, y) = (135km, 570km)$ *in the Wadden sea by using Tradition particle model for* $t \gg T_L$*.*

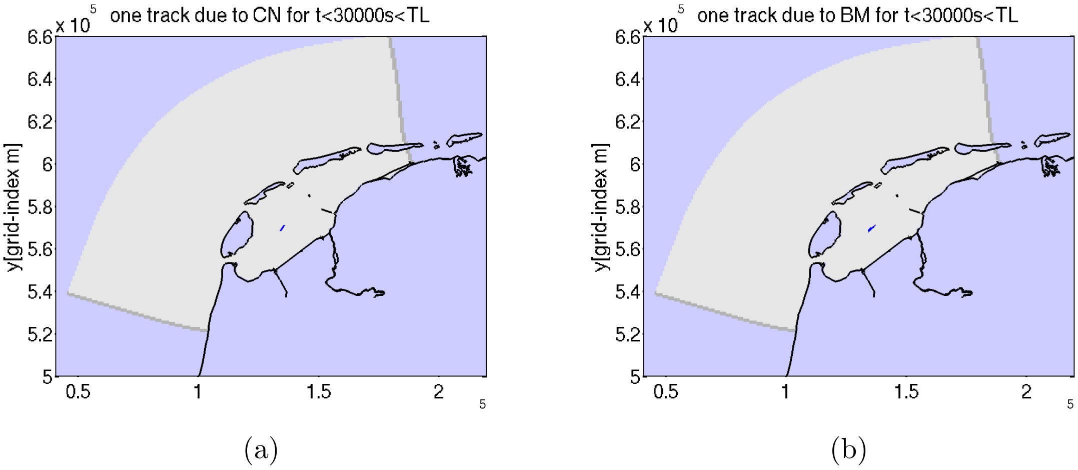
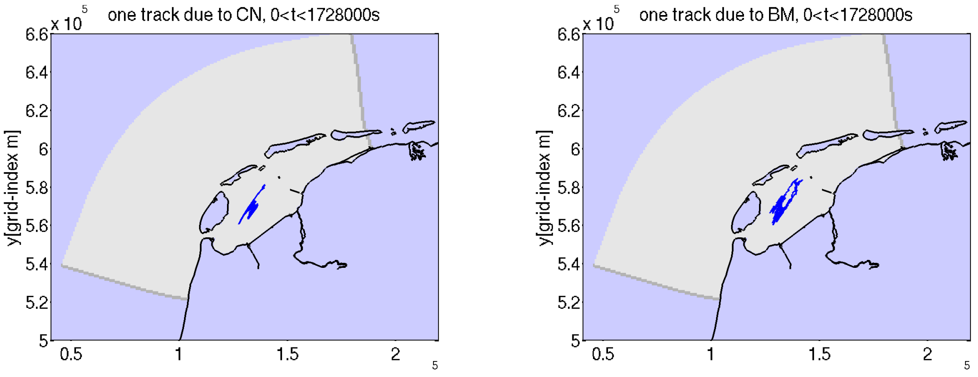


Figure 3.12: (a) Tracking of a single marked particle domain by using Random flight for $t < T_L$*, (b) Tracking of a single marked particle by using Tradition particle model for* $t < T_L$*.*



(a) (b)

Figure 3.13: (a) Tracking of a single marked particle by using Random flight for $t \gg T_L$, (b) Tracking of a single marked particle by using Tradition particle model for $t \gg T_L$.

3.9 Conclusions

The results show that the spreading of a cloud of particle for the short time scale is slow when it is driven by coloured noise force while it is much faster when it is driven by Brownian motion on the same scale. Therefore, an improvement in modelling the dispersion of pollutants on a small time scale is attained when a coloured noise force is employed. Thus, a random flight model will provide the modeller with an enhanced tool for the short term simulation of the pollutants by providing more flexibility to account for correlated physical processes of diffusion in the shallow waters. However, in this chapter a general analysis shows that a process observed over a long time spans as modelled by the coloured noise force behaves much like a Brownian motion model with variance parameter $\sigma^2 T_L^2 \prod_{i=1}^{n} \alpha_i^2$. The use of coloured noise however is more expensive in terms of computation and therefore it is advisable to use the random flight model for short term behaviour while adhering to the traditional particle model for long-term simulations.

3.10 Appendix

It is well known[52], that for any continuous function we have the following theorem.

Theorem 1 *Let $g(x)$ be continuous function and $\{W(t), t \geq 0\}$ be the standard Brown-*

ian motion process. For each $t > 0$, there exits a random variable

$$\mathcal{F}(g) = \int_0^t g(x)dW(x),$$

which is the limiting of approximating sums

$$\mathcal{F}_n(g) = \sum_{k=1}^{2^n} g(\frac{k}{2^n}t)[B(\frac{k}{2^n}t) - B(\frac{k-1}{2^n}t)],$$

as $n \to \infty$. The random variable $\mathcal{F}(g)$ is normally distributed with mean zero and variance

$$\mathrm{Var}[\mathcal{F}(g)] = \int_0^t g^2(u)du,$$

if $f(x)$ is another continuous function of x then $\mathcal{F}(f)$ and $\mathcal{F}(g)$ have a joint normal distribution with covariance

$$\mathbb{E}[\mathcal{F}(f)\mathcal{F}(g)] = \int_0^t f(x)g(x)dx.$$

Summary of the simulation parameters of particle for pollutants dispersion in shallow waters

Channel	**Unit**	**Value**	**North sea**	**Unit**	**Value**
# of steps	-	8999	# of steps	-	2499
Δt	s	864	Δt	s	864
Particles	-	40000	Particles	-	$400,000$
α_1, α_2	-	$\alpha_1 = 1, \alpha_2 = 0.2$	α_1, α_2	-	$\alpha_1 = 1, \alpha_2 = 0.2$
α_3, α_4	-	$\alpha_3 = 1.2, \alpha_4 = 2$	α_3, α_4	-	$\alpha_3 = 1.2, \alpha_4 = 2$
Tracks	-	5	Tracks	-	5
Grid offset	m	$(-21600, -21600)$	grid offset	m	$(40550, 500000)$
Grid size	-	105×105	Grid size	-	201×201
Cell size	m	400×400	Grid size	m	800×800
Init. point	m	$(0, 0)$	Init. point	m	$(1.35 \times 10^5, 5.7 \times 10^5)$
D	m/s^2	3	D	m/s^2	3
T_L	s	30000	T_L	s	30000

Table 3.1: The simulation parameters of the particle model for the dispersion of pollutants in shallow waters.

Chapter 4

The particle model for simulation of sediment transport in coastal waters

Overview[1]: In this chapter we have developed a mathematical model known as a particle model to describe transport of sediment particles in shallow waters. The additional processes in the model such as erosion and deposition are modelled by using probabilistic concepts. The resulting Fokker-Planck equation due to this particle model, is interpreted into an Eulerian transport model. By so doing, the underlying particle model is shown to be consistent with the Eulerian transport model with the deposition and erosion terms. Eventually, we have implemented the particle model. This particle model uses real data such as the water depths, water levels as well as the velocities of the flow fields of the Dutch coastal waters of the North Sea. The idea was to test our particle model for the approximation of the sea bed-level changes.

4.1 Introduction

The morphology of many shallow water areas is characterised by a complex pattern in both space and time [46]. This behaviour is caused by the feedback among the *water motion*, *sediment transport* and *bottom changes*. It is clear that the morphological behaviour is very sensitive to changes in external conditions [59]. The change can be caused, for example, by the rise of sea level or human activities [23, 27]. Therefore, there is a need to simulate and predict morphological processes and their sensitivity to changing conditions both for economic and ecological reasons.

Morphological changes in estuaries play an important role in coastal protection, ecology, economy, water quality, dredging activities in channels and so forth. In recent years, different morphodynamic models have been developed (see e.g. [40]). It was demonstrated in [61] that the effects of bottom topography on tidal motions is to generate

---

[1] This chapter is a revised version of [9] published in the J. of NAMP. 8 (2004), pp. 131-138.

residual currents and overtides, which turn out to have a large impact on the mixing properties of such systems.

The bedform changes have a significant effect on the flow dynamics contributed by imbalance between sediments in and out from those areas. The imbalance can easily be disturbed by the external factors, such as extreme storm events, mean sea-level rise, changes in tidal regime, human interferences and so on. Therefore, a better prediction of these bedforms is required to be able to understand their sensitivity to external conditions. In this chapter we have developed and described a particle-based approach to simulate entrainment, transport, and settling of non-cohesive sediments in shallow waters. Sediment distributions are modelled as a set of particles that are tracked on an individual basis by solving Lagrangian transport equations that account for the drift part by the mean flow, settling, and random horizontal motions.

The rest of this chapter is organised as follows. The brief description of an Eulerian transport model is done in Section 4.3. While the particle model for sediment transport is discussed in Section 4.4. The interpretation of the partial differential equation called Fokker-Planck equation into the well-known Eulerian transport model for sediment transport is described in Section 4.4.5. The description and discussion of the two dimensional channel for a test case of sediment transport is carried out in Section 4.6. The numerical implementation has been carried out in the North sea of the Dutch coastal waters. The results are available in the application section 4.8 of this chapter.

In our work we do carry out the estimation of the change in bedforms. Nevertheless, we do not yet recompute the flow velocities when a change in the shallow water depth occurs.

4.2 Shallow water flow equations

In order for particle models to describe transport problems in shallow waters, the inputs such as water flow velocities $[U(x,y,t),V(x,y,t)]^T$, water levels ξ, water depths $H(x,y,t)$ and so forth are required. In our application, the inputs are often computed by the hydrodynamic model(WAQUA), which can solve the depth-averaged shallow water equations or 3 dimensional shallow water [49]. The generated results in this case are written into a matlab format that can be loaded and read in the particle model for simulation of sediment transport. The inputs are assumed to satisfy the shallow water equations [18]. The momentum equations are represented by the following equations:

$$\frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+V\frac{\partial U}{\partial y}+g\frac{\partial \xi}{\partial x}-fV+g\frac{U(U^2+V^2)^{\frac{1}{2}}}{(C_z)^2H}=0 \qquad (4.1)$$

$$\frac{\partial V}{\partial t}+U\frac{\partial V}{\partial x}+V\frac{\partial V}{\partial y}+g\frac{\partial \xi}{\partial y}+fU+g\frac{V(U^2+V^2)^{\frac{1}{2}}}{(C_z)^2H}=0. \qquad (4.2)$$

The velocity is uniform over the vertical, therefore, for that reason, the rise and fall of free surface is given by equations of conservation of mass called the continuity equation:

$$\frac{\partial H}{\partial t}+\frac{\partial(UH)}{\partial x}+\frac{\partial(VH)}{\partial y}=0, \tag{4.3}$$

where

$$\begin{aligned} H &= h+\xi \text{ is the total depth;}\\ \xi & \quad \text{is the water-level with respect to a reference;}\\ h & \quad \text{depth of the water with respect to a reference;}\\ C_z & \quad \text{bottom friction coefficient (Chezy coefficient);}\\ g & \quad \text{acceleration of gravity;}\\ f & \quad \text{Coriolis parameter.}\end{aligned}$$

The shallow water equations are entirely described by equations (4.1)-(4.3), provided the closed and open boundary conditions and initial fields are given [29].

4.3 Eulerian sediment transport model

In this section we briefly introduce the Eulerian model for sediment transport. We consider noncohesive type of sediment particle. The dynamics of the suspended particles can be described by the well-known Eulerian transport model with the source and sink terms included. The following Eulerian sediment transport model is similar to that in [59], for example:

$$\begin{aligned}\frac{\partial(HC)}{\partial t}+\frac{\partial(HUC)}{\partial x}+\frac{\partial(HVC)}{\partial y}-\frac{\partial}{\partial x}(D\frac{\partial HC}{\partial x})-\frac{\partial}{\partial y}(D\frac{\partial HC}{\partial y})\\ =-\gamma HC+\mathcal{E}(U,V)\cdot\lambda_s.\end{aligned} \tag{4.4}$$

Where $C(x,y,t)$ is depth averaged concentration, γ is the deposition coefficient, $\mathcal{E}(U,V)=(U^2+V^2)(m^2s^{-2})$ is a function of flow velocities and the term $\lambda_s\cdot\mathcal{E}(U,V)$ models erosion of sediment particles. The particle pick up function is parameterized as $\lambda_s\cdot\mathcal{E}(U,V)$, where, λ_s is the erosion coefficient, it can be related to sediment properties (grain size, grain shape). This parameterisation is motivated by the analysis of field observations reported in [46] and reference therein. Typically, $\lambda_s\approx 3\times10^{-2}(kgsm^{-4})$ for fine sand. In this article $\lambda_s=0.0001(kgm^{-4}s)$ is within the range reported in literature (see e.g., [45, 15]). Note that the term γHC models the deposition of sediment and γ is the deposition coefficient, it is reported that $\gamma\approx 4\times10^{-3}s^{-1}$ [46] for fine sand.

4.3.1 Determination of bedlevel changes by Eulerian transport model

In addition to suspension and deposition processes, the following equation is used to determine the depth changes and therefore the change of bed-level in each grid cell i, j:

$$\frac{\partial h}{\partial t} = \frac{1}{(1-po)\rho_s}(d_e - s_e). \tag{4.5}$$

Where $s_e = \gamma HC$ stands for deposition and $d_e = \lambda_s \cdot \mathcal{E}(U, V)$ stands for erosion (term responsible for suspending particles). Sea bed porosity is represented by po, ρ_s is the density of sediment particles.

In this section we have constructed a simplified transport model which is derived from the Eulerian transport model (4.4). This simple model is then made consistent with the simplified Lagrangian particle model. In this way it becomes easy to compare the bedlevel changes to see if they are similar. We simplified Eqn. (4.4) by assuming that the deposition and erosion processes balance:

$$\gamma CH \;=\; (U^2 + V^2) \cdot \lambda_s. \tag{4.6}$$

Quite often, transport in water is defined as the product of the concentration of sediment particles C and a velocity U or V as well the depth of water in that grid cell. Thus, transport along x and y directions is respectively given by;

$$q_x \;=\; UCH \text{ and } q_y = VCH,$$

where in vector form $\overline{q} = [q_x, q_y]^T$, using Eqn. (4.6), it follows that

$$\overline{q} = [(U^2 + V^2)U \cdot f_d, (U^2 + V^2)V \cdot f_d]^T, \tag{4.7}$$

where $f_d = \frac{\lambda_s}{\gamma}$ stands for the drag force. This depends on the properties of a particle for example its size or its area. Thus, in order to determine how much mass exits or comes into a given location, it is important to consider the divergence. The divergence determines the average rate of how much mass comes into the cell(change of mass per second per area):

$$\frac{\partial m}{\partial t} \;=\; -\text{div}(\overline{\text{q}}), \tag{4.8}$$

where $\overline{q} = \frac{1}{T}\int_0^T q dt$, div stands for divergence. Consequently, Eqn. (4.8) represents the rate of how much mass stays behind or leaves the cell by assuming the absence of destruction or creation of a matter. Since we want to determine the effects of sediment transport on the sea bedforms, the equation for the bed level is represented by;

$$\frac{\partial h}{\partial t} = \frac{1}{(1-po)\rho_s}\frac{\partial m}{\partial t}. \tag{4.9}$$

In the present application the determination of the bed level change using finite difference scheme is estimated by the following equations:

$$\frac{\partial h}{\partial t} \;=\; -\frac{1}{\rho_s(1-po)} \cdot \mathrm{div}(\overline{\mathrm{q}}).$$

For cases where flow is in one direction for example when $v = 0$, transport is given by the following equation:

$$\frac{\partial h}{\partial t} \;=\; -\frac{f_d}{\rho_s(1-po)} \cdot \mathrm{div}(\overline{\mathrm{U}^3}).$$

With the aid of Eqn. (4.8)–(4.9), accordingly the determination of bed level changes is now done by using the following equation:

$$\Delta h \;\approx\; \frac{-f_d T}{\rho_s(1-po)} \cdot \left(\frac{\partial \mathbf{u_m}}{\partial x} + \frac{\partial \mathbf{v_m}}{\partial y}\right). \tag{4.10}$$

where $\mathbf{u_m} = \frac{1}{T}\int_0^T (U^2+V^2)U dt$ and $\mathbf{v_m} = \frac{1}{T}\int_0^T (U^2+V^2)V dt$. Next, let us now discuss the Lagrangian particle model in the following section.

4.4 A particle model for sediment transport in shallow waters

A particle model is a description of a transport process by means of random walk models. Random walk model is defined as the stochastic differential equation that describes the movement of a particle that subsequently undergoes a displacement, which consists of the drift part and a stochastic(diffusive) part [40].

4.4.1 Integration of particle movement

In this section, the following 2-dimensional stochastic differential equations is developed:

$$\begin{aligned} dX(t) &\overset{\text{Itô}}{=} \left[U + \tfrac{D}{H}\left(\tfrac{\partial H}{\partial x}\right) + \tfrac{\partial D}{\partial x}\right] dt + \sqrt{2D} dW_1(t), \\ dY(t) &\overset{\text{Itô}}{=} \left[V + \tfrac{D}{H}\left(\tfrac{\partial H}{\partial y}\right) + \tfrac{\partial D)}{\partial y}\right] dt + \sqrt{2D} dW_2(t), \end{aligned} \tag{4.11}$$

where the Brownian process $W_1(t)$ and $W_2(t)$ are Gaussian [35], and $D(x, y, t)$ is the horizontal dispersion coefficient for sediment transport. Typically, $D = \mathcal{O}(10-100)m^2/s$ [46]. Note that $U(x, y)$ and $V(x, y)$ are the flow velocities along the x and y direction respectively given in m/s, $H(x, y)$ is the averaged total depth plus relative water levels due to waves, $dW_1(t)$ and $dW_2(t)$ are independent increments of Brownian motions with mean $(0, 0)^T$ and covariance $\mathbb{E}[dW_1(t)dW_2(t)^{\mathrm{T}}] = Idt$ where I is an identity matrix ([3, 32]). The simulation of sediment transport is initiated with zero number of particles.

4.4.2 Deposition of sediment particles

We associate with each sediment particle a binary state which at any time t is given by

$$\mathrm{S}_t = \begin{cases} 1 & \text{particle is suspension} \\ 0 & \text{otherwise (particle is on sea bed).} \end{cases}$$

Given a particle in suspension, we are interested in the transition of state 1 to state 0. In a continuous form, this transition can be modelled by the following equation

$$\frac{d\mathrm{P}(S_t = 1)}{dt} = -\gamma \cdot \mathrm{P}(S_t = 1), \qquad \text{where initially} \qquad \mathrm{P}(\mathrm{S}_0 = 1) = 1 \tag{4.12}$$

where $\gamma(x, y, t)$ is the deposition coefficient, in this chapter $\gamma = \gamma(x, y, t)$ is constant, $\mathrm{P}(S_t = 1)$ is the probability that the state of the particle at time t is 1. When the particle is in the flow, its evolution is described by the following transition probability equation in discrete form:

$$\begin{aligned} \mathrm{P}(S_{t+\Delta t} = 1 \mid S_t = 1) &= \mathrm{P}(S_0 = 1) \cdot [1 - \gamma \cdot \Delta t] \\ &= [1 - \gamma\Delta t] \end{aligned} \tag{4.13}$$

During the period of the time step, (Δt), the system state (e.g. flow field and turbulence patterns) is assumed constant. It follows that

$$\mathrm{P}(S_{t+\Delta t} = 0 \mid S_t = 1) = 1 - [\mathrm{P}(S_{t+\Delta t} = 1 \mid S_t = 1)], \tag{4.14}$$

is the probability that a particle will be deposited.

4.4.3 Suspension of sediment particles

Mass represents concentration of a group of particles at a certain location. A source term is included in our particle model such that the expected number of suspended particles (enp) in grid cell i, j at time t is given by

$$enp_{(i,j,t)} = \frac{\Delta x \cdot \Delta y \cdot \Delta t \cdot \left(U^2 + V^2\right) \cdot \lambda_s}{\mathcal{M}_p}, \tag{4.15}$$

where $\mathcal{M}_p$ is the mass of each particle, Δx and Δy are the width of the grid cells along x and y directions respectively, Δt is the time step size, and λ_s is the erosion coefficient. For each grid cell i, j we use the expected number of particles from Eqn. (4.15) to determine the actual number of particles to be suspended. This is done by drawing a number from a Poisson distribution function. Furthermore, the treatment of boundaries condition is the same as that in Section 3.6.1 in chapter 3. In addition, this chapter have also considered the pick up of sediment particles at the inflows this will be discussed in the next section.

4.4.4 Particle flux at open boundaries

We now consider particle flux at open boundaries. This flux is the difference between particles flowing out of and particles flowing into the domain. The number of particles flowing out should not be controlled as it is a natural consequence of the movement of a particle. As soon as it crosses the open boundary it is considered gone and further integration is no longer possible as no data outside the domain is given. For this very reason, however, we do need to explicitly model the particles flowing in. We determine the expected number of particles entering the domain as follows:

$$enp_{(i,j,t)} = \begin{cases} \frac{\Delta y \cdot \Delta t \cdot V \cdot (U^2+V^2) \cdot \lambda_s}{\gamma \mathcal{M}_p} & \text{in-flow parallel to y-axis} \\ \\ \frac{\Delta x \cdot \Delta t \cdot U \cdot (U^2+V^2) \cdot \lambda_s}{\gamma \mathcal{M}_p} & \text{in-flow parallel to x-axis,} \end{cases} \tag{4.16}$$

where γ is the deposition coefficient. The actual number of particles added at the domain boundary at each iteration is obtained by drawing a value from the Poisson distribution parameterised by the above calculated expectation value.

For each grid cell i, j we use the expected number of particles from Eqn. (4.15) to determine the actual number of particles to be suspended by drawing a number from a Poisson distribution function. We assume that particles are infinitely many on the sea bed. However, the particle that is suspended is not the same as the one that is deposited.

Before implementing the particle model (4.11)-(4.16), we first required to show the consistence between the Fokker-Planck equation and its the Eulerian transport model. This will be described in the next Section.

4.4.5 The connection between the Fokker-Planck equation and Eulerian transport model

In this section the relationship between the particle model and Eulerian transport model for sediment transport is discussed in detail. Since we are interested in the particle being in suspension, we assume that the particle at position (x, y) at time t has expectation of their mass $\langle \cdot \rangle$ defined by;

$$\langle m(x,y,t) \rangle = f(x,y,t) \cdot \mathrm{P}(\mathrm{S_t} = 1). \tag{4.17}$$

This is known as mass density of particles per unit area. We aim at deriving the Fokker-Planck equation that incorporates suspension and sedimentation state of the particle. Let D be the diffusion coefficient, $\mathrm{P}(\mathrm{S_t} = 1)$, is the probability that particle is in suspension, $(S_t = 1)$ denotes a state that a particle is in suspension and $(S_t = 0)$ denotes the state that the particle is deposited on the sea bed. The stochastic process (X_t, Y_t) is a Markov process. The probability density function of the particle position $f(x, y, t)$ based on the two dimensional SDEs (4.11) evolves according to the following Fokker-Planck equation [28].

$$\begin{aligned}\frac{\partial f(x,y,t)}{\partial t} &= -\frac{\partial}{\partial x}\left(\left[U + (\frac{\partial H}{\partial x}D)/H + \frac{\partial D}{\partial x}\right] \cdot f(x,y,t)\right)\\ &\qquad -\frac{\partial}{\partial y}\left(\left[V + (\frac{\partial H}{\partial y}D)/H + \frac{\partial D}{\partial y}\right] \cdot f(x,y,t)\right)\\ &\qquad\qquad + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(f(x,y,t)\cdot 2D\right) + \frac{1}{2}\frac{\partial^2}{\partial y^2}\left(f(x,y,t)\cdot 2D\right). \end{aligned} \tag{4.18}$$

In this chapter we have extended the two dimensional pollution transport model (4.11) which was discussed in [28]. The particle model now includes the erosion and deposition terms. The next step is to derive a Fokker-Planck equation as follows, we first differentiate equation (4.17) with respect to time t, we obtain

$$\frac{\partial}{\partial t}\langle m(x,y,t)\rangle = \mathrm{P}(S_t = 1)\frac{\partial}{\partial t}f(x,y,t) + f(x,y,t)\frac{\partial}{\partial t}\mathrm{P}(S_t = 1),$$

next with the aid of Eqn. (4.12), it follows that,

$$\frac{\partial}{\partial t}\langle m(x,y,t)\rangle = \mathrm{P}(S_t = 1)\frac{\partial}{\partial t}f(x,y,t) - \gamma f(x,y,t)\cdot \mathrm{P}(S_t = 1). \tag{4.19}$$

Therefore, we also add the erosion term to Eqn. (4.19) and come up with

$$\frac{\partial}{\partial t}\langle m(x,y,t)\rangle = \mathrm{P}(S_t = 1)\frac{\partial}{\partial t}f(x,y,t) - \gamma f(x,y,t)\cdot \mathrm{P}(S_t = 1) + \lambda_s \cdot \mathcal{E}(U,V) \tag{4.20}$$

next we multiply on both sides of Eqn. (4.18) by $\mathrm{P}(\mathrm{S_t} = 1)$ to obtain

$$\begin{aligned}\mathrm{P}(S_t = 1)\frac{\partial}{\partial t}f(x,y,t) &= -\frac{\partial}{\partial x}\left(\left[U + \frac{D}{H}\frac{\partial H}{\partial x} + \frac{\partial D}{\partial x}\right] f(x,y,t)\cdot \mathrm{P}(S_t = 1)\right)\\ &\qquad -\frac{\partial}{\partial y}\left(\left[V + \frac{D}{H}\frac{\partial H}{\partial y} + \frac{\partial D}{\partial y}\right] f(x,y,t)\cdot \mathrm{P}(\mathrm{S_t} = 1)\right)\\ &\quad + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(2D\cdot f(x,y,t)\mathrm{P}(\mathrm{S_t} = 1)\right) + \frac{1}{2}\frac{\partial^2}{\partial y^2}\left(2D\cdot f(x,y,t)\cdot \mathrm{P}(\mathrm{S_t} = 1)\right). \end{aligned} \tag{4.21}$$

The substitution of Eqn. (4.21) into equation (4.20) gives the following Fokker-Planck equation with the deposition and erosion terms.

$$\begin{aligned}\frac{\partial\langle m(x,y,t)\rangle}{\partial t} &= -\frac{\partial}{\partial x}\left(\left[U + \frac{D}{H}\frac{\partial H}{\partial x} + \frac{\partial D}{\partial x}\right]\cdot \langle m(x,y,t)\rangle\right)\\ &\qquad -\frac{\partial}{\partial y}\left(\left[V + \frac{D}{H}\frac{\partial H}{\partial y} + \frac{\partial D}{\partial y}\right]\cdot \langle m(x,y,t)\rangle\right)\\ &\quad + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(2D\cdot \langle m(x,y,t)\rangle\right) + \frac{1}{2}\frac{\partial^2}{\partial y^2}\left(2D\cdot \langle m(x,y,t)\rangle\right)\\ &\qquad\qquad - \gamma\cdot \langle m(x,y,t)\rangle + \lambda_s \cdot \mathcal{E}(U,V). \end{aligned} \tag{4.22}$$

Therefore, the last two terms in the transport equation (4.22) stand for the deposition and erosion terms respectively. An average mass $\langle m(x, y, t)\rangle$ per unit area (kg/m^2) of a particle is related to a particle depth averaged concentration $C(x, y, t)$ in mass per unit volume (kg/m^3) ([57, 28]).

Since the concentration is given in kg/m^3, the expected mass of a particle at position (x, y) can be related by the concentration at that position by the following equation:

$$\langle m(x, y, t)\rangle = H(x, y, t) \cdot C(x, y, t). \tag{4.23}$$

By substituting Eqn. (4.23) into the Fokker-Planck equation (4.22) one can obtain the Eulerian sediment transport model (4.4). Consequently, the transport equation (4.4) which was discussed in Section 4.3, is consistent with the particle model for sediment transport (4.11)-(4.16).

Therefore, after having constructed the particle model for sediment transport, it is now necessary to develop the equations that cater for the bed level changes using the particle model.

4.4.6 Determination of bedlevel changes using particle models

Comparing with a simplified form of Eqn. (4.4), where in this case we assume for the local change in mass $\frac{\partial m}{\partial t} \approx d_e - s_e$. We also assume that the deposition and erosion processes balance (see Section 4.3.1). The approximation of the change in mass in each grid cell i, j in the this particle model is determined by the following equations:

$$\frac{\partial m}{\partial t} = \frac{\Delta \mathcal{N}_p}{\Delta t} \frac{1}{\Delta x \Delta y} \mathcal{M}_p. \tag{4.24}$$

Using equation (4.5) and the fact that $\frac{\partial m}{\partial t} \approx d_e - s_e$, the equation for the bed level change can be derived as follows,

$$\frac{\partial h}{\partial t} = \frac{\Delta \mathcal{N}_p}{\Delta t} \frac{1}{\rho_s (1 - po) \Delta x \Delta y} \mathcal{M}_p.$$

Where $\mathcal{M}_p$ is the mass of each particle, $\Delta \mathcal{N}_p$ is the difference between deposited and suspended particles at each iteration in each grid cell i, j, $\rho_s = 2650 kg/m^3$ denotes the density of an individual grain particle and $po \approx 0.5$ denotes the bed porosity [46]. Hence the cumulated change in sea bedlevel Δh in m for all time steps is determined by the following equation:

$$\begin{aligned} \Delta h &= \int_0^T \frac{\partial h}{\partial t} dt \\ \Delta h &\approx \sum \frac{\Delta \mathcal{N}_p}{\rho_s (1 - po) \Delta x \Delta y} \mathcal{M}_p. \end{aligned} \tag{4.25}$$

More information about the effect of parameters on the sea bed level changes can be found in [15], for example.

In the next section we shall briefly discuss the numerical approximation of our particle model.

4.5 Numerical approximation of the particle model

Euler scheme is used in the numerical implementation of the particle model. The scheme is convergent in the weak sense with accuracy of order $\mathcal{O}(\Delta t)$ and it is half order accurate in the strong sense. Higher order schemes for stochastic differential equations are described in [33]. The discretisations of the hydrodynamic flow models is widely discussed by [49], for example. The particle model (4.11)-(4.16) is discretised and uses the following Euler scheme to approximate the numerical solutions. We discretise the two dimensional stochastic differential equations for integrating the movement of the particle in similar way to that as in [28] with the modifications by the inclusion of the sedimentation and deposition parts:

$$\bar{X}(t_{k+1}) = \bar{X}(t_k) + \left[U + (\frac{\partial H}{\partial x}D)/H + \frac{\partial D}{\partial x}\right]\Delta t_k + \sqrt{2D}\Delta W_1(t_k) \tag{4.26}$$

$$\bar{Y}(t_{k+1}) = \bar{Y}(t_k) + \left[V + (\frac{\partial H}{\partial y}D)/H + \frac{\partial D}{\partial y}\right]\Delta t_k + \sqrt{2D}\Delta W_2(t_k) \tag{4.27}$$

$$\mathrm{P}_{k+1}(S_t = 1) = (1 - \gamma(x, y, t)\Delta_k)\mathrm{P}_k(S_t = 1). \tag{4.28}$$

Where $\bar{X}(t_{k+1})$ and $\bar{Y}(t_{k+1})$ are the numerical approximations of $X(t)$ and $Y(t)$ respectively, while $\bar{X}(t_0) = X(t_0) = x_0$ and $\bar{Y}(t_0) = Y(t_0) = y_0$ are initial locations of a particle. In addition to Eqns. (4.26)-(4.28), we also use Eqns. (4.15)-(4.16) to make the simulation of sediment transport complete. There are several schemes that can be used for simulation process for instance, Euler, Heun, Milstein scheme and Runge kutta methods. Much detailed work on numerical methods for stochastic models can be found in (e,g., [50, 35, 41]).

Numerical schemes such as the Euler scheme often show very poor convergence behaviour [34]). This implies that, in order to get accurate results, small time steps are needed thus requiring much computation. Another problem with the Euler (or any other numerical scheme) is its undesirable behaviour in the vicinity of boundaries; a time step that is too large may result in particles unintentionally crossing boundaries. To tackle this problem we define two types of boundaries. Closed boundaries representing boundaries intrinsic to the domain, and open boundaries which arise from the modeller's decision to artificially limit the domain at that location. Besides these boundary types we also define what happens if, during integration, a particle crosses one of these two borders. More discussion is found in Section 3.6.1 of chapter 3 in this thesis.

4.6 Experiment of sediment transport in the two dimensional channel

Let us design a two dimensional channel $x = [-10000m, 10000m]$, $y = [2m, 4500m]$ (see Figure 4.1) with the inflows, main domain and the outflows and use it as a test case. The channel domain is designed in such a way that the flow is only in along the x direction, $v = 0$ along y. The depths are generated by a Gaussian function, the flow field at each time in each grid cell is generated according to the following function:

$$u = \frac{h_0}{h}\left(u_1 \sin(\omega \cdot t) + u_0\right), \qquad v = 0 \tag{4.29}$$

where $h_0 = 20[m]$ is an undisturbed depth below mean sea level, $u_0 = 0.025 m/s$ is the tidal mean flow, tidal amplitude $u_1 = 0.5[m/s]$. Hence, Eqn. (4.29) generates the flow fields such that it is possible to determine the average sediment transport [15]. It generates u along x direction and $v = 0$ along y direction.

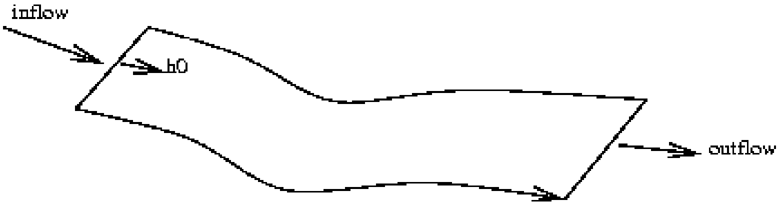


Figure 4.1: Two dimensional channel showing the inflow, main domain and outflow.

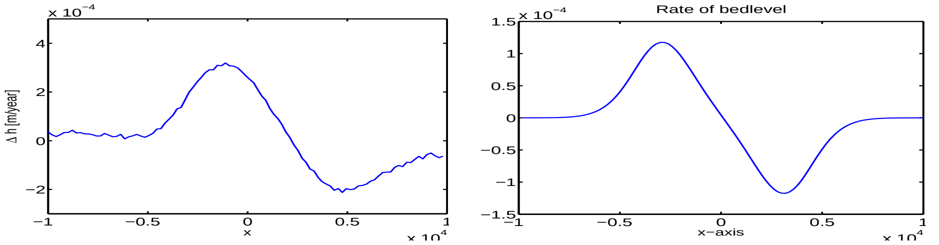
Primary input of the model

The initial field is defined as

$$depth = h_0 + h_1 \cdot \exp(-((x - 0.0).^2)/(2 \cdot (wd)^2)),$$

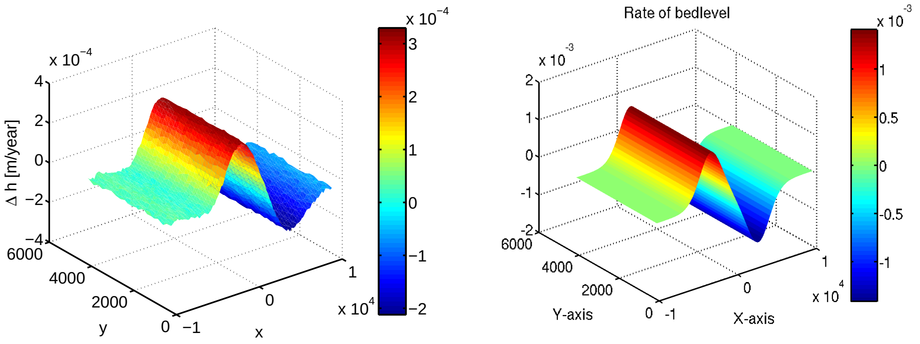
where the initial amplitude of the disturbance $h_1 = 10.0$, width of disturbance $(wd) = 2000m$, the tidal period $T = 720$ minutes. Constant for sediment transport rate $K = 0.16 kgs^3/m^6$, porosity $p0 = 0.5$, $\rho = 2600[kg/m^3]$, density of sediment, horizontal domain length of x goes from $-10000m$ to $10000m$. The horizontal domain length of y goes from $2m$ to $4500m$.

Diffusion constant $D = 10m^2/s$, starting time $Tstart = 0s$, number of seconds in a year $Tyear = 365 \cdot 24 \cdot 60 \cdot 60 s/year$. Final time s $Tstop = 100 \cdot Tyear$, $M = 50$ is the number of grid points across the channel, $dt = 1.0 \cdot year$ is the time-step s. Tidal mean discharge per unit width m^2/s is given by $q_0 = h_0 \cdot u_0$, tidal amplitude discharge per unit width m^2/s is given by $q_1 = h_0 \cdot u_1$. We then determine the bedforms as described in Section 4.3.1. It is known that the divergence of transport describes the rate of the bed level changes in each grid cell. As a matter of comparison, we compute the divergence of transport by using the original data from the hydrodynamic model described in this section. Note that by using a two dimensional channel, we first compute the U, V as well as its depth. Then compute average load transport and finally compute the divergence using a finite difference methods. Where the divergence of the U-vector field along x and the V-vector field along y is evaluated using central differences wherever possible and forward or backward differences on the boundaries. This is done where the average bedforms are estimated by using Eqn. (4.10) in Section 4.3.1. The results obtained are eventually compared with those obtained when the estimations of the bedforms are carried out using Eqn. (4.25) with the aid of the Lagrangian particle model discussed in Section 4.4.6. Note that both results due to Eqn. (4.10) and Eqn. (4.25) are similar, (see e.g Figure 4.3). The results depicted from the figures shows that there is deposition followed by erosion (see Figure 4.2 and Figure).



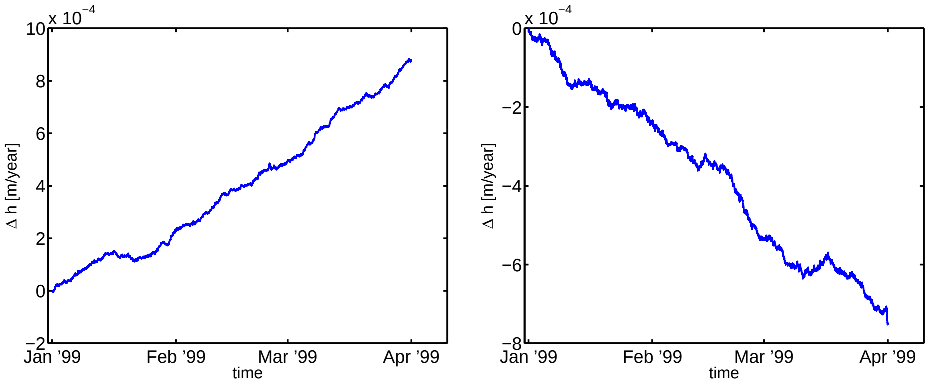
(a) (b)

Figure 4.2: Change of bed level in $m/year$ for a two dimensional channel (a) is due to the particle model, while (b) the result is computed by using the Eulerian approach.



(a) (b)

Figure 4.3: Change of bed level in $m/year$ for a two dimensional channel (a) is due to the particle model, while (b) the result is computed by using the Eulerian approach.



(a) Deposition (b) Erosion

Figure 4.4: Cumulative local changes in depth by using data of 90 *days for two selected grid cells in the ideal two dimensional channel domain.*

The Figures 4.2 and 4.3 represent the results obtained by solving the same problem by using two approaches. Figure 4.2(a) and Figure 4.3(a) are due to a simplified particle model with very small effect of the diffusion. On the other hand, in Figures 4.2(b) and Figure 4.3(b) are due to a simplified Eulerian model, no diffusion is considered. We should expected to get deposition at the retardation of the flow and erosion at the acceleration of the flow. Resulting in the net migration of the channels in the direction of $(\overline{U^3})$ as in [58]. The positive sign on the colorbar in the figures imply that deposition is taking place while the negative sign implying the occurrence of erosion of sediments. Some results, in Figure 4.4(a,b), represent the local depth change in two selected cells. Part (a) shows a steady deposition in the grid cell at the location $(x, y) = (5km, 2km)$ whereas another grid cell in the location $(x, y) = (-0.1km, 0.3km)$, part (b), there is a steady erosion. The diffusion coefficient in the test case for the particle model is $0.00001m^2/s$, $\gamma = 0.00013s^{-1}$. The parameters for the test case experiments using ideal domain with the PM are found in table 4.1.

4.7 Transport of heavy particles in shallow water

In this section we will consider heavy particles. Heavy particles tend to attain the equilibrium much faster than the tidal cycle. The description of the derivation of the equations for the heavy particles have followed similar lines using the same equations as those for the lighter particles.

$$\frac{\partial(CH)}{\partial t} + \frac{\partial(HUC)}{\partial x} + \frac{\partial(HVC)}{\partial y} = S_e - S_d \tag{4.30}$$

Constant	**Unit**	**Value**	**Constant**	**Unit**	**Value**
porosity	-	0.50	γ	s^{-1}	0.00013
grid offset	m	$(-10000, 2)$	λ_s	$kg{\cdot}s{\cdot}m^{-4}$	0.001
grid size	-	22×101	D	$m^2 \cdot s^{-1}$	0.000010
cell size	m	200×200	δ	-	0
sand density	kgm^{-3}	2650	$\mathcal{M}_p$	kg	3000
initial location	m	$(1800, 400)$	$f = \frac{\lambda_s}{\gamma}$	$kg \cdot s^2 \cdot m^{-4}$	$\mathcal{O}(10^{-4})$

Table 4.1: Parameters used by particle model for the sediment transport in the test case

where in this case $S_e = \lambda_s U^k$ is erosion term, $-S_d = \gamma CH,$ is the deposition term where λ_s, γ, k are constants. In addition, an equation for the change of the bedlevel is presented:

$$\frac{\partial h}{\partial t} = \frac{1}{\rho_s(1-po)} \cdot (S_e - S_d). \tag{4.31}$$

However, the difficulty with the heavy (coarser) material is that the parameter γ becomes larger (settling velocity) and this makes the equation (4.30) 'stiff'. The maximum allowed time step thus becomes very small and since we want to make long simulation this is not very convenient. Therefore, the solution is to make a first order approximation and assume that the two source terms S_e and S_d are much bigger than the right side terms of Eqn. (4.30). Therefore, we can write equation (4.30):

$$\delta \left(\frac{\partial(CH)}{\partial t} + \frac{\partial(HUC)}{\partial x} + \frac{\partial(HVC)}{\partial y} \right) = S_e - S_d, \tag{4.32}$$

now we substitute a Taylor series expansion of the depth averaged concentration CH

$$CH = Q = Q_0 + \delta Q_1 + \mathcal{O}(\delta^2),$$

into Eqn. (4.32) to get the following equations such that 0-order is given by :

$$S_e - S_d = 0, \tag{4.33}$$

while the 1^{st} order is:

$$\frac{\partial(Q_0)}{\partial t} + \frac{\partial(UQ_0)}{\partial x} + \frac{\partial(VQ_0)}{\partial y}) = -(\gamma Q_1). \tag{4.34}$$

In other words, in Eqn. (4.34), we are looking for a source term (γQ_1) that balances the advective transport of the 0 -order solution. Although this is easy in finite difference approach, however, in the particle model, Eqn. (4.34) can be approached as follows. The best we can do so far is to solve:

$$\frac{\partial Q}{\partial t} + \frac{\partial (UQ)}{\partial x} + \frac{\partial (VQ)}{\partial y} = 0, \tag{4.35}$$

with (4.33) as an initial condition for every time step separately and then set

$$\gamma Q_1 \approx \frac{\partial Q}{\partial t}. \tag{4.36}$$

In other words, Eqn. (4.34) should be more or less balanced. If we now omit the source term and measures the rate of changes, these should be approximately equal. Therefore for the heavy particles, between the beginning and end of the integration time loop should do the following

(i) First we remove all deposited particles.

(ii) Followed by generating particles according to Eqn. (4.33)

(iii) Store net change in number of particles (concentration).

(iv) Next we do one time step integration of the particle using Eqn. (4.35).

(v) Compute differences over the previous step using (iii).

For now we have assumed that

$$|\frac{\partial Q_0}{\partial t}| \ll |\gamma Q_1|.$$

Note that

$$\begin{aligned} \text{transport vector } q_x &= UHC = \frac{\lambda_s}{\gamma} U(U^2+V^2) \qquad (4.37) \\ \text{transport vector } q_y &= VHC = \frac{\lambda_s}{\gamma} V(U^2+V^2), \qquad (4.38) \end{aligned}$$

since $\lambda_s U^2 - \gamma CH = 0$, the equation for the rate change of the bedlevel due to the transport coarse material is given by

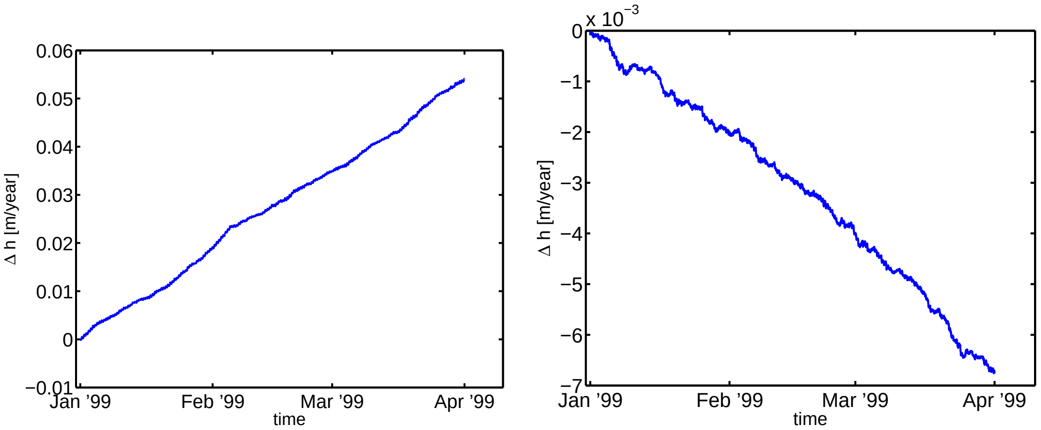
$$\frac{\partial h}{\partial t} = -\left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y}\right) \cdot \frac{1}{\rho_s(1-po)}, \tag{4.39}$$

where $f_d = \frac{\lambda_s}{\gamma}$ is the pick up function which depends into the characteristics of the sediment/sand materials. For example in case of a mixture of larger sand of volume=l^3 will have a mass=$\rho_s l^3$. While a sand of double size whose volume is $8l^3$ will have mass=$\rho_s 8l^3$. Therefore, the two particle require different value of the drag force particle f_d. But in this chapter we assume that all particles have the same mass, l=length.

4.8 Application and results

Having implemented and tested the particle model for sediment transport, in this section we carry out the simulation experiments for sediment transport using real data of the Dutch coastal waters. The Lagrangian particle model is again compared with the Eulerian approach where a finite difference method is used. A realistic application of the particle model (4.11)-(4.16) is implemented and applied in the shallow waters of the Dutch coastal notably in the Wadden sea. The results for the light particles in Figure 4.5 (a) exhibits a steady change of deposition where $\Delta h \approx 0.0056 m/year$ and in Figure 4.5 (b) a steady erosion such that $\Delta h \approx -0.0063 m/year$. We have also noticed that the results in Figure 4.6(b) due to the Eulerian transport model is similar to the results due to the particle model for sediment transport in Figure 4.6(a).

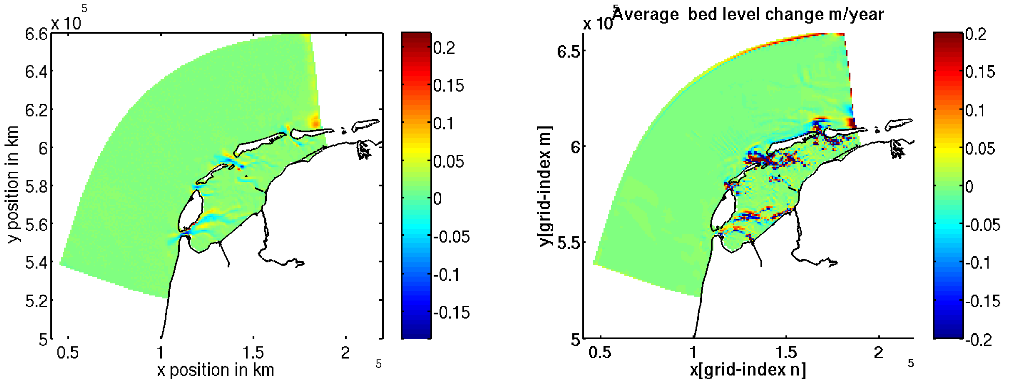
Furthermore we also tested another experiments for the heavy particles using the particle model and used slightly the same simulation parameters as before. Using the same Eqn. (4.10) and Eqn. (4.25) respectively for the Eulerian and Lagrangian approach for the particle model. The result for the heavy particles due to the particle model in Figure 4.7 (a) gave relatively better results and similar to that simplified Eulerian transport model in Figure 4.7 (b). The parameters which were used in the implementation of the heavy particles using the particle model are found in Table 4.2.



(a) Deposition

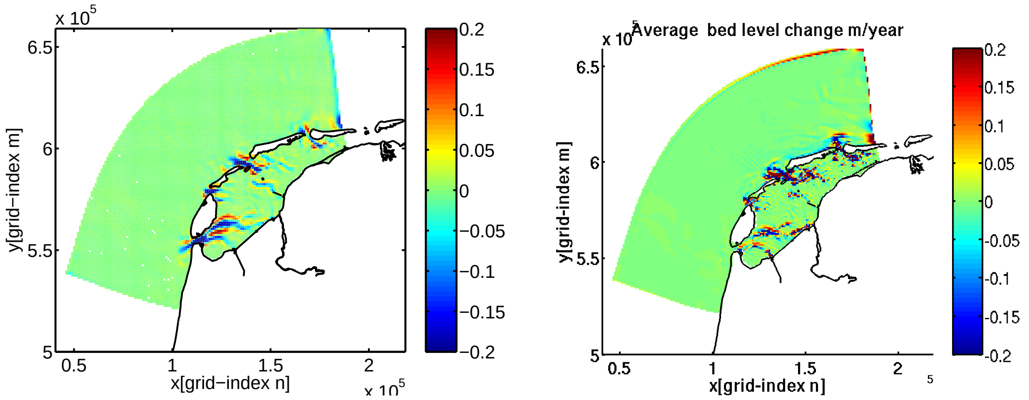
(b) Erosion

Figure 4.5: Cumulative local changes in depth by using data for 90 days for light particles in the two selected grid cells.



(a) (b)

Figure 4.6: Change of bed level in $m/year$ (a) is the change computed by using the particle model for light particle, while (b) the result is computed by using the Eulerian approach.



(a) (b)

Figure 4.7: Change of bed level in $m/year$ for 90 days, (a) is the change computed by using the particle model for Heavy particle, while (b) is computed by using the Eulerian approach

4.9 Discussion and conclusions

In this chapter we have developed a particle model and implemented it for sediment transport in shallow waters of the Dutch coastal. We have also derived the Fokker-

Constant	**Unit**	**Value**	**Constant**	**Unit**	**Value**
porosity	-	0.5	γ	s^{-1}	0.0013
grid offset	km	$(40.55, 500)$	λ_s	$kg{\cdot}s{\cdot}m^{-4}$	0.0001
grid size	-	180×161	D	$m^2 \cdot s^{-1}$	10
cell size	m	1000×1000	δ	-	1.7
sand density	kgm^{-3}	2650	$\mathcal{M}_p$	kg	10000
initial location	km	$(135, 570)$	$f = \frac{\lambda_s}{\gamma}$	$kg \cdot s^2 \cdot m^{-4}$	$\mathcal{O}(10^{-4})$
sedimentation prob.	-	0.0112	Δt	s	3600

Table 4.2: Parameters used by the particle model for real data.

Planck equation and included the deposition and erosion terms based on the developed particle model. Moreover, we have also shown that by interpreting the Eulerian sediment transport as the Fokker Planck equation with the additional terms, it becomes possible to derived the underlying particle model that is consistent with the Eulerian sediment transport model. Furthermore, the results of sediment transport due to particle model has been compared to that obtained by computing the transport using Eulerian model in their simplest form. We have got some results for the changes in the bed level. We have used our model to test the prediction of bed-level changes by using the real data of the Dutch North sea. Therefore, at least for now we can say that we have solved the set of mathematical equations called particle model for sediment transport. These equations have given us reasonable results for the sea bedlevel changes. Nevertheless, the determination of the morphological changes is a complicated process that depends on several factors such as waves, the size of sand, mass and density. The particle model in this case has been simplified, what we can say is that the results are reasonable. But more factors will have to be taken into account. For instance, in the particle model we have considered that all particles have equal mass while in reality each particle has different mass. The flow in WAQUA was computed in curvilinear grid mesh but for our particle model, we first transformed it into rectilinear. The set up of the parameters in the two approaches differs a little, as such the results are slightly different in the two approaches (see Figure 4.6(a)-(b) and Figure 4.7(a)-(b)) for the light and heavy particles respectively.

Chapter 5

Parallel and distributed simulation of sediment dynamics in shallow water using particle decomposition approach

Overview[1]: This chapter describes the parallel simulation of sediment dynamics in shallow water. By using a Lagrangian model, the problem is transformed to one in which a large number of independent particles must be tracked. This results in a technique that can be parallelized with high efficiency. We develop and describe the model by presenting three sediment suspension methods and the corresponding issues of load balancing among the processors. We measure the speed up, efficiency as well as load balance of all three approaches on a Beowulf cluster.

5.1 Introduction

The simulation of sediment transport can be done by the numeric integration of a set of stochastic differential equations augmented with erosion and deposition terms. These simulations give rise to tracks of sediment particles along their path in time, and are known as the Lagrangian particle approach. In this work we develop a parallel Lagrangian particle model for efficient sediment transport simulation in shallow waters. The inputs for our particle models, such as velocity and water depth, are time dependent and are computed by the 2-dimensional hydrodynamic model known as WAQUA. It is one of the oldest and most commonly used computation models at the National Institute for Coastal and Marine Management(RIKZ) in the Netherlands.

In this chapter we only consider a parallel particle model for transport of sediments, the flow computation itself is not considered. The parallel computation of the flow veloc-

---

[1] This chapter [7] is a revised version of the paper submitted to the Journal of Parallel and Distributed Computing.

ity has been implemented by [44], and [60] using a domain decomposition approach. We do not yet consider the two-ways coupling involving the flow and the Lagrangian particle model. Nevertheless, the depth changes are slow and happen after long simulation time [15] unlike the dynamics of the flows. Therefore we do not frequently communicate the depths changes. Even if we would have coupled the two, it is unlikely that it would affect the high speedups that we have attained. But for application in real life (such as stratification due to concentration differences) it is possible and necessary to couple the flow and the particle model. However, there is a prototype for a two-way coupling of the flow and particle transport models which was developed in our group [39].

In order to get accurate results from Monte Carlo simulations of sediment transport, a large number of particles is often needed. However, the computation time in a particle model increases linearly with the number of particles. In addition, as memory requirements grow this too may be a limiting factor. Using parallel processing it is possible to both significantly reduce the simulation time as well as to alleviate memory problems. Because the particle tracks are inherently independent of one another it is natural to compute these in parallel [37]. It is known that parallel particle computations themselves are quite trivial. However, the sediment transport model in this chapter has a number of features that may lead to efficient suspension and sedimentation of particles as we shall see later.

For optimal use of parallel machines, it is important to maintain a good load balance as well as minimise the required communication time. For particle models two basic decomposition methods exist; those in which each processor simulates particles only in a predetermined or potentially adaptive region of the domain and communicates them to the processors managing neighbouring domains. This approach is called domain decomposition. In the second approach, particle decomposition, the particles are evenly distributed to different groups and each group assigned to a processor. This approach is very attractive because it requires only a minimal amount of communication and a good load balance is achieved relatively easy. This in contrast to domain decomposition where subdomains need to be carefully determined to evenly spread the load. In addition, communication between processors on a parallel architecture may inhibit the potential speed up (see e.g., [22, 43]). The disadvantage of the particle decomposition approach is that it requires each processor to store flow data of the entire domain which can be huge in volume. Our parallel implementation in C (see Section 5.4) is based on particle decomposition and uses the message-passing interface (MPI) implementation for deployment on a cluster (see also [22]). Three different approaches to particle suspension techniques were considered (see Section 5.3.2).

This chapter is arranged as follows, we discuss the Lagrangian particle model for sediment transport in Section 5.2 followed by the parallel processing considerations in Section 5.3.

The three different sediment suspension approaches are discussed in Section 5.3.2. The numerical implementation of the Lagrangian particle model is described in Section 5.4. Sections 5.5 and 5.5.1 respectively cover the application and the results. As a realistic case study, the parallel Lagrangian approach is applied to the Dutch coastal waters, notably the Wadden sea.

5.2 Sediment transport using Lagrangian particle models

The Lagrangian particle model in this chapter is slightly different from the particle model discussed in Section 4.4.1 of chapter 4 for it also includes the effect of slope in the model. Let the position of a sediment particle at time t along the x and y directions be denoted by $(X(t), Y(t))$. The numerical integration of movement of such a particle in a flow, taking into account the effect of sea bed slope, proceeds by the following 2-dimensional stochastic differential equations(SDEs):

$$\begin{aligned} dX(t) &\overset{\text{Itô}}{=} \left[U\left(1+\operatorname{sign}(U)\delta\tfrac{\partial H}{\partial x}\right)+\tfrac{D}{H}\left(\tfrac{\partial H}{\partial x}\right)+\tfrac{\partial D}{\partial x}\right]dt+\sqrt{2D}dW_1(t) \\ dY(t) &\overset{\text{Itô}}{=} \left[V\left(1+\operatorname{sign}(V)\delta\tfrac{\partial H}{\partial y}\right)+\tfrac{D}{H}\left(\tfrac{\partial H}{\partial y}\right)+\tfrac{\partial D}{\partial y}\right]dt+\sqrt{2D}dW_2(t) \end{aligned} \qquad (5.1)$$

where U and V are the flow velocities in m/s along respectively the x and y direction. H is the sum of the (interpolated) depth and the relative water levels due to waves at (x, y). Next, $dW_1(t)$ and $dW_2(t)$ represent independent increments of (Gaussian) Brownian processes $W_1(t)$ and $W_2(t)$, and have mean $(0,0)^T$ and covariance $\mathbb{E}[dW_1(t)dW_2(t)^{\mathrm{T}}] = Idt$ with I the identity matrix as in (see e.g., [32, 3]). More information on this concept can be found in [35]. Finally, $D(x, y, t)$ is the horizontal dispersion coefficient for sediment transport, δ is a constant coefficient for the slope of the sea bed. Typically, $D = \mathcal{O}(10-100)m^2/s$ [46]. The effects of slope of the sea bed is incorporated in the drift part such that the velocity of the flow is influenced by the slope. By setting $\delta = 0$, we can choose to omit this dependency.

The drift term consists of the flow fields as well as a correction term. This prevents the particles from accumulating in a location with low diffusivity [57]. A common choice for the random part is the Gaussian probability function [3, 35].

For sediment transport modelling, a number of sub-processes have to be added to the description of pure movement to cover erosion, deposition and behaviour at the sea bed [40]. In the remainder of this section we study these additional processes and consider suspension, sedimentation, and in and outflow at domain boundaries.

The deposition of sediment particles uses the same equations as those in Chapter 4 Eqns. (4.12)–(4.14). The suspension of sediment particles also use similar Eqns. (4.15)–(4.16). This is followed by a discussion about the resulting changes in bed level.

5.2.1 Determination of bed level changes using Lagrangian models

Recall that each particle represents a mass of sediment material. For Lagrangian particle models, the approximate change in mass with respect to time, in each grid cell i, j is determined by the following equation:

$$\frac{\partial m}{\partial t} = \frac{\Delta \mathcal{N}_p}{\Delta t}\frac{1}{\Delta x \Delta y}\mathcal{M}_p. \qquad (5.2)$$

Whereas the change in water depth can be done using:

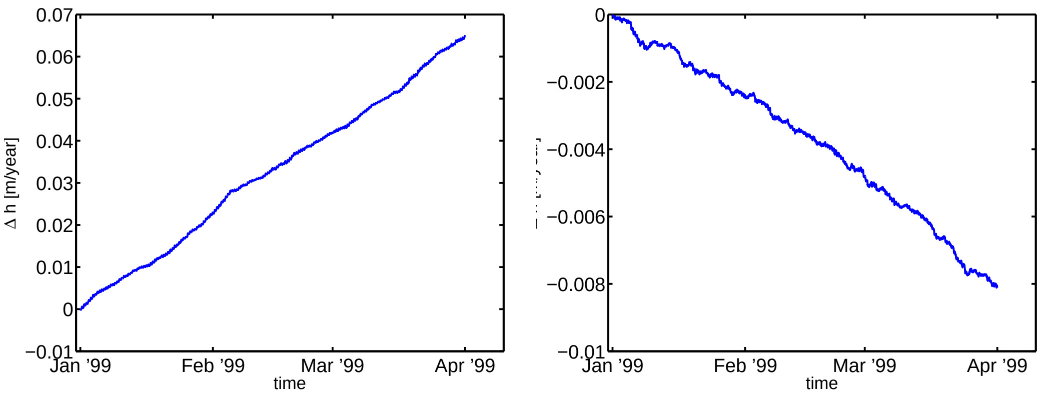
$$\frac{\partial h}{\partial t} = \frac{\Delta \mathcal{N}_p}{\Delta t} \frac{1}{\rho_s(1-po)\Delta x \Delta y} \mathcal{M}_p.$$

Here $\mathcal{M}_p$ is the mass of each particle, $\Delta \mathcal{N}_p$ is the difference between deposited and suspended particles at each iteration. Next, from [46] we have $\rho_s = 2650 kg/m^3$ denoting the density of an individual grain particle and $po = 0.5$ for sea bed porosity.

The cumulative change in water depth Δh (or sea bed level $-\Delta h$) in meter all iterations is determined by the following equation:

$$\Delta h = \int_0^T \frac{\partial h}{\partial t} dt, \qquad \Delta h \approx \sum \frac{\Delta \mathcal{N}_p}{\rho_s(1-po)\Delta x \Delta y} \mathcal{M}_p. \tag{5.3}$$

Some results, plotted in Figure 5.1(a,b), represent the local depth change in two selected cells. Part (a) shows a steady deposition in the first grid cell at the location $(x, y) = (136km, 595km)$ whereas in the second grid cell in the location $(x, y) = (138km, 600km)$, part (b), there is a steady erosion.



(a) Deposition (b) Erosion

Figure 5.1: Cumulative local changes in depth by using data for 90 *days for two selected grid cells.*

For the implementation details of this particle model, see Section 5.4.

5.3 Parallel processing

Having discussed the particle transport model for sediment transport, we now move on to introducing a parallel approach in order to speed up the sediment transport simulation. Parallel processing is the simultaneous processing of different tasks by two or more microprocessors, as by a single computer with more than one central processing unit or by multiple computers connected together in a network so as to form a Beowulf cluster[22].

Terminologies encountered in parallel computations

- **Speedup** — Ratio of the time taken to solve a problem on a single processor to the time required to solve the same problem on a parallel computer with p processors.

$$S(p) \quad = \quad \frac{T_1}{T_p}. \tag{5.4}$$

- **Efficiency** — Ratio of speedup and the number of processors.

$$E(p) \quad = \quad \frac{S(p)}{p} \quad = \quad \frac{T_1}{p \cdot T_p}. \tag{5.5}$$

5.3.1 Work decomposition

In our work, the computation domain of the particle model is characterized by a grid representing the geometry of the coastal waters and particles representing sediment [38]. There are several ways in which the computation can be divided and assigned to processes. The most notable two of these are domain decomposition and particle division. Domain decomposition is often applied in Eulerian hydrodynamic model such as the two dimensional hydrodynamic model WAQUA [60]. We mention this particular model here not only as an example but also because it is the computed velocity of water movement generated by this program that serves as the flow data in our simulation. Note that these data such as velocity, depth, salinity, pressure gradient and so forth, are computed off-line and stored in files which are subsequently read in by our program.

In the Lagrangian particle simulation the amount of work is essentially proportional to the number of particles to be tracked. The loading of flow information from files is considerable but does not dominate the processing time. In general, the particle distribution within the domain is non-uniform.

Below we consider the domain and particle decomposition approaches. The goal is to find a partitioning in which the amount of computation that can be done in parallel should be maximal while the communication should be minimal.

Domain decomposition

In domain decomposition the domain is divided into a number of sub-domains and the work corresponding to each such sub-domain is assigned to a processor (see Figure 5.2(a)). Domain decomposition approach are relatively easy to implement and can be used with the same data distribution over the processors as those in the hydrodynamic model (e.g., [38]). There are also a number of notable disadvantages with domain decomposition (more information on parallel processing concepts can be found in e.g., [43]);

1. Differences in flow velocity and particle density make it difficult to partition the domain in such a way that each processor handles the same amount of computation.
2. The domain may need to be re-partitioned after a number of time steps to maintain a good load balance. It also needs to be done whenever the number of processors used is changed.
3. Particles (as well as their tracks) have to be communicated between the processors when particles cross the sub-domain boundary. Besides, very irregular size of the communication requires sending both size and data. Note that the communication of the particle data is between the neighbouring processors only, but when the subdomain becomes small (*i.e.* as more processors are used), the amount of particles moving in and out a subdomain can be large, and resulting in a relative large communication overhead.
4. Particles along with their path history have to be communicated when they enter the domain region managed by another process. Note that processes handling neighbouring regions need to synchronise after each step to see if particles crossed and if so, transmit the relevant data.

Management of the deposited and suspended particles and their influence on the bed level, or likewise base depth of the water, can be done locally since processes handled the particles that deposited or suspended. Particles near region boundaries can fluctuate in such a way that they need to be repeatedly communicated between the neighbouring processes. To avoid this undesirable situation, one commonly introduces a small amount of overlap known as ghost regions and illustrated in Figure 5.2(b) by the shaded boxes. If the boundary of a ghost region itself is crossed, the particle will end up well within the domain handled by the other process. In order to keep the state of the ghost regions coherent, changes in depth within this region on one processor need to be communicated to processes sharing that boundary region.

Particle decomposition

In this approach the number of particles is evenly divided across the available processes. This way the workload will be balanced under the assumption that the amount of calculations for each particle is more or less equal. Only slight load imbalance due to particles stranding or flowing out of the domain and differences in the number of particles deposited. Note that these processes can be considered stochastic with certain mean, so that over longer periods of time this amount will be the same on each processor leading to an overall stable load distribution without any need for communication. A similar argument applies for suspension.

5.3.2 The process of sediment suspension

In this section we discuss three parallel particle suspension approaches. The relative performance in terms of speed up and efficiency will be considered later, in Sec. 5.5.1.

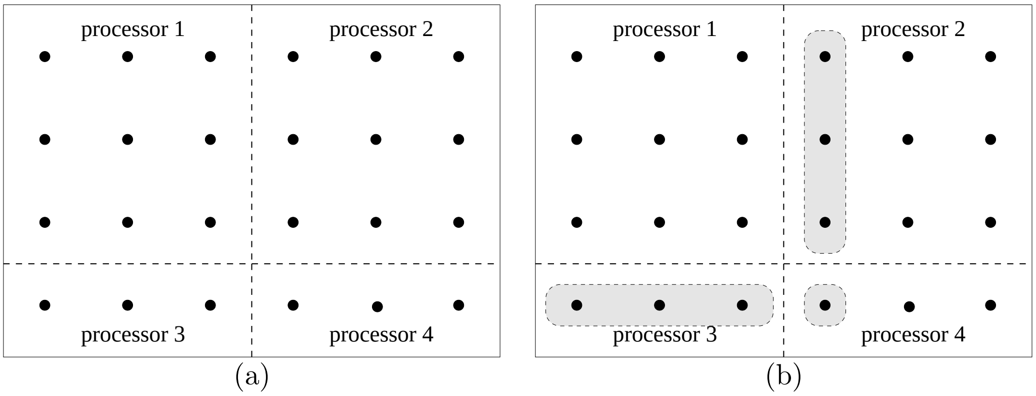


Figure 5.2: (a) Domain decomposition with different parts of the domain assigned to different processor, (b) ghost regions for processor 1 for the same decomposition, indicated by the shaded boxes.

Method I – Modified expectation

For each grid cell i, j, the global expectation is computed and divided by the total number of processors to obtain the local expectation. Using this local expectation, we drawn a number from a Poisson distribution with appropriate parameter and use this as the number of local particles, *i.e.* on this processor, to be suspended in that grid cell.

i. For each grid cell i, j that is part of the defined domain, calculate the expected number of particles to suspended by

$$\mathrm{enp_{i,j}} = \frac{\Delta\mathrm{x} \cdot \Delta\mathrm{y} \cdot \Delta\mathrm{t} \cdot \lambda_\mathrm{s}}{\mathcal{M}_\mathrm{p}}. \tag{5.6}$$

ii. Determine the local number of particles to suspend in each grid cell i, j

$$\mathrm{nLocal_{i,j}} = \left\lfloor \mathrm{Poisspdf}\left(\frac{\mathrm{enp_{i,j}}}{\mathrm{np}} \cdot (\mathrm{U}^2 + \mathrm{V}^2)\right)\right\rfloor$$

where Poisspdf denotes the Poisson distribution function. The Poisson distribution was chosen because it has an unbiased mean and is memoryless.

iii. Add the given number of particles to the data structure,

Method II – Assign domain regions to processors

This method applies a rudimentary form of domain decomposition in which each processor is systematically assigned a number of grid cells. Suspension is done only for those cells assigned to the processor, the other cells are handled by other processors. The method is easy to implement but the domain again needs to be decomposed with the inherent problem of maintaining a non-biased load distribution. The global expectation of the Poisson distribution of the grid cells i, j is determined as in the sequential version, but used directly to determine the number of suspended particles on the current processor. The approach can be summarised as follows.

i. Determine the number of grid cells available and assign them to processors according to the following equation:

$$p_{ij} = (i + j \cdot M) \mod \text{np},$$

with $0 \leq i < M$ and $0 \leq j < N$ the row and column index, M the number or rows and N the number of columns in the cell grid. This way processors are randomly spread over the domain and may lead to relatively good load balance.

ii. Next compute $enp_{i,j}$ in those grid cells allocated to processor ID using Eqn. (5.6).

iii. Determine the local number of particles to suspend in each grid cell i, j

$$\mathrm{nLocal_{i,j}} = \left\lfloor \mathrm{Poisspdf}(\mathrm{enp_{i,j}} \cdot (U^2 + V^2)) \right\rfloor .$$

iv. Finally add the number of particles to the data structure.

Method III – Globally synchronised

For each cell grid i, j a random number is drawn from a special, globally synchronised random number generator according to the distribution used in the sequential program (see Section 5.3.2). This random number, say n, represents the total number of particles to suspend in that cell and is identical on each processor, without communication. Each processor is initially assigned $\lfloor \frac{n}{\text{np}} \rfloor$ particles. The remaining $n \bmod \text{np}$ particles that cannot be evenly distributed are circularly assigned to processors (one to each candidate processor, see Figure 5.3). This way the number of particles is guaranteed to remain perfectly balanced with differences of at most one.

In summary; for each grid cell i, j within the domain, compute $\mathrm{enp_{i,j}}$ using Eqn. (5.6) and perform the following steps.

i. Save the state of the current random number generator and switch to the synchronised global number generator.

ii. Determine the global number of particles to suspend for this cell by drawing from the following distribution:

$$\mathrm{nGlobal_{i,j}} = \lfloor \left(\mathrm{Poisspdf}(\mathrm{enp_{i,j}} \cdot (U^2 + V^2)) \right) \rfloor .$$

iii. Restore the state of the local random number generator

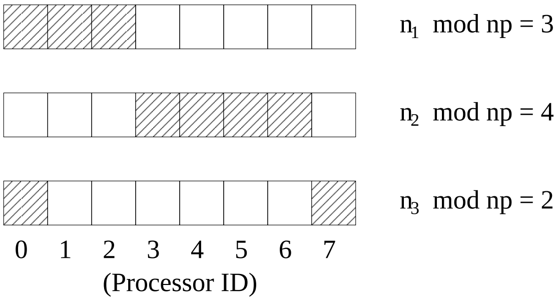


Figure 5.3: Assigning n_k mod np *particles equally among processors for three successive cells* k*. For the first cell we have a remaining 3 particles to assign to 8 processors, which is indicated by the shaded blocks. For the second cell we have 4 which are assigned to processors 3–6. Finally, a remaining 2 particles are assigned to processors 7, and 0.*

iv. Determine the local number of particles to suspend for each grid cell inside the domain

$$\mathrm{nLocal}_{\mathrm{i,j}} = \left\lfloor \frac{\mathrm{nGlobal}_{\mathrm{i,j}}}{\mathrm{np}} \right\rfloor .$$

add one particle for processors p, whenever $(p{+}\mathrm{np}{-}\mathrm{offset}) \bmod \mathrm{np} < (\mathrm{nGlobal}_{\mathrm{i,j}} \bmod \mathrm{np})$.

v. Set offset $:= (\mathrm{offset} + \mathrm{nGlobal}_{\mathrm{i,j}}) \bmod \mathrm{np}$.

vi. Initialise and add the particles to the data structure.

Global random number generator

In order to maintain good load balance throughout, it is important that the number of particles suspended on each processor differs only a little and ideally be equal. By explicitly synchronising the number of particles intended for suspension globally, it is possible to achieve this ideal distribution. This does come at a cost however, because of the communication overhead. By careful manipulation of the pseudo random number generator we can draw random numbers that are the same on each processor. We assume that uniform (pseudo) random numbers are generated according to the following equation:

$$x_{k+1} \equiv a x_k + b \ \mathbf{mod}\ \mathbf{c},$$

where a, b and c are suitable chosen constants, x_k the previously drawn random number, x_{k+1} is the next random number and x_0 represents the initial seed value. Given this, we proceed as follows.

Upon program initialisation we choose (i.e., based on the current time, an input parameter or otherwise) a seed value, say s, on the root processor which is broadcast to all other processors. They then manipulate this number using their processor identifier to obtain an initial local seed value (*e.g.* $s_p = s + p + 1$, where p is the processor ID). The global seed value that was broadcast however, is stored as well for global random number generation. All we then need to do is switch between the generators, based on these two seeds, at appropriate times. If the mathematics library of the programming environment stores the most recently drawn random number, x_{k+1}‘, it suffices to have one global variable storing this number when switching between random number generators (obviously, not until after the present value holding the current seed of the generator to switch to has been copied). No further modifications to the standard random number generator is needed. If, on the other hand, no such functionality is provided a wrapper function posing as the random number generator needs to be written. This implies that the standard generator cannot be used anymore which has the disadvantage that other functions such as drawing from a normal distribution that rely on it have to be rewritten. This will be especially difficult when they are provided in standard libraries that do not offer the possibility of setting the random number generation function.

5.3.3 Sedimentation

Sedimentation itself is done on a particle basis and can therefore be perfectly done in parallel. The only disadvantage is that we have no control over the number of particles deposited and this can therefore differ slightly between processors. This effect is minimal however, because of the law of big numbers in statistics. We expect, after all, to have many particles, otherwise parallel simulation would not be needed in the first place.

5.3.4 Equilibrium determining the expected number of particles in the flow

An equilibrium of the particles can be attained when the expected number of suspended and deposited particles are equal:

$$\bar{n}_t = \frac{\lambda_s \Delta t}{\mathcal{M}_p(1 - e^{-\gamma \Delta t})} \left[\Delta x \Delta y \sum_{(i,j)\in\mathcal{D}} w + \frac{\Delta x}{\gamma} \sum_{(i,j)\in\mathcal{B}} Uw + \frac{\Delta y}{\gamma} \sum_{(i,j)\in\mathcal{B}} Vw \right],$$

where w is defined as

$$\mathrm{w} = \mathrm{U}^2(\mathrm{x}, \mathrm{y}, \mathrm{t}) + \mathrm{V}^2(\mathrm{x}, \mathrm{y}, \mathrm{t}),$$

while $\mathcal{B}$ and $\mathcal{D}$ represent the set of grid cells at the boundary respectively within the domain such that $\mathcal{B} \cap \mathcal{D} = \emptyset$ and $\mathcal{B} \cup \mathcal{D}$ encompasses all valid grid cells. The number of particles at equilibrium determines the amount of work to be done per time step. Most terms in the above equation for $\bar{n}_t$ are independent of Δt and therefore, the expected amount of work per iteration is $\mathcal{O}\left(\frac{\Delta t}{1-e^{-\gamma\Delta t}}\right)$. Given the fact that there are $\mathcal{O}(1/\Delta t)$ iterations in a simulation of fixed time interval, the total amount of work is $\mathcal{O}\left(\frac{1}{1-e^{-\gamma\Delta t}}\right)$. As expected, the total amount of work goes down as the step size increases.

5.3.5 Depth update

The global effect of suspension and sedimentation on the sea bed level cannot be determined without communication, since each processor handles only a subset of all particles. As a result, they only know about the changes in bed level resulting from particles under their control. There are several ways in which the depth can be updated globally.

a. Communicate all deposited and suspended particles between all processor pairs (an optimisation can be done here because we can already know the number of particles suspended in a certain region as this is done the same on each processor). This approach is preferable when the number of particles is smaller than the number of grid cells. However, all-to-all communication of the deposited and suspended particles becomes impractical when the total number of particles is large. In other words, this way of handling particle updates does neither scale very well with the number of particles nor with the number of processors. The advantage is that communication does not need to be done after each iteration, unless the morphology changes during each iteration are significantly large.

b. Locally determine the depth changes in the grid cells and collect this information. This is another way to reduce communication by locally computing of the grid cell depth changes and use a reduction scheme to aggregate all changes. The accumulation of the depth changes can be done over a multiple time steps per processor before you communicate them as there is a large difference in time scales. This approach is preferable when the number of particles is large than the number of grid cells.

c. Determine the depth changes in the grid cells and communicate only those grid cells that have a change of depth. Here, the cell index along with the number of particles deposited or suspended needs to be communicated to the other processors.

Advantages of particle decomposition

The advantages of the particle decomposition approach are:

- The particle division parallel approach is domain independent.
- No lengthy computations prior to the particle simulation are needed.
- It scales well with the number of processors used, without having to re-initialise the domain.
- Load balance can be maintained very well when using the third approach discussed in (Section 5.3.2); suspension can be done while maintaining a perfect load balance without any communication by careful manipulation of the seed value used in the (pseudo) random number generator [22].

However, the disadvantage of the particle decomposition approach is that it requires each processor to store flow data of the entire domain which can be huge in volume.

5.4 Implementation

The following briefly describes the implementation of the Lagrangian particle model for sediment transport in shallow water as described in Section 5.2. The implementation consists of a set of modules, shown in Figure 5.7, working together through predefined interfaces, simplifying replacement of one model implementation by another. The integration module, for example, is given a set of particles, the current time, and the step size and performs numerical integration of the SDEs governing particle movement. The particular choice of scheme used does not affect the other modules. One can either replace the implementation, or pass parameters to the application which are passed to, recognised, and removed by the integration module, prescribing the numerical scheme to use, if implemented by the module. The latter approach of module specific parameters can be implemented in each module and allows for a high level of flexibility, without having to recompile the simulation. This facilitates automatic processing of simulations with different parameters, and can be used to analyse what-if scenarios.

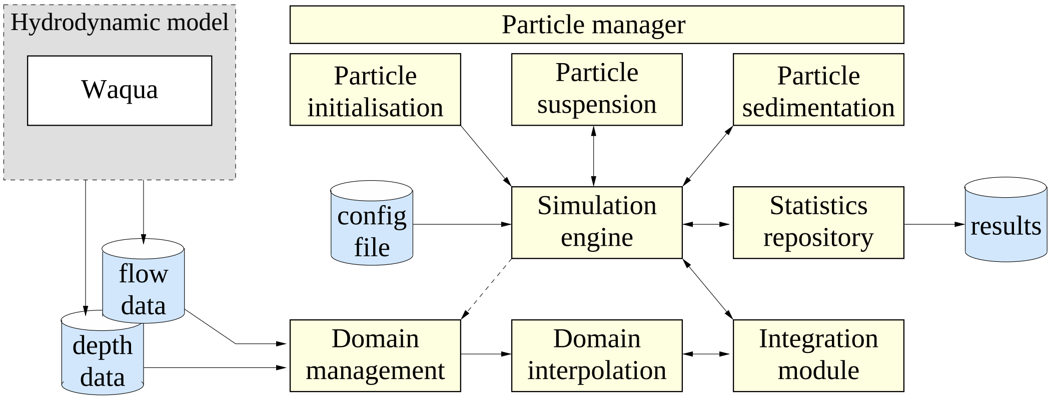


Figure 5.4: An overview of the software for the parallel particle model for sediment transport in shallow water.

5.4.1 Modules

Hydrodynamic model: The hydrodynamic model carries out the flow computation. It is a numerical model, the time steps used by the WAQUA model were 2 minutes and stores the results to disk at 60 minutes intervals along with the depths information in a collection of files. We load the flow at every one hour it is not directly coupled with the particle model. These provide all the domain information required by the simulation. In our case, the software used is water quality model(WAQUA) [60]), a software package of the directorate General for public works and water management in the Netherlands.

Simulation engine: The simulation engine forms the core of the whole simulation. It starts by processing the command line argument and, if needed, loading one or more configuration files. It then triggers the initialisation of all modules and provides them with the parameters given. After all modules are successfully initialised, it enters the main time stepping loop. At each iteration it requests the domain module to ensure the current data are available, does the suspension, integration and sedimentation and instructs the statistics repository to extract all relevant data from this iteration. Finally it requests all modules to finalise and stops the application. Throughout, the simulation engine does all the required (parallel) error checking, and if needed, the graceful abortion of all parallel instances.

Domain management: This module performs the loading, caching and overall management of domain data and is triggered by the simulation engine to prepare data for a specific time, say, t. In addition it also provides interpolation of data at two consecutive times, although this is completely hidden from all other modules and can be requested by specifying the appropriate parameters. Upon request is provides the flow and depth data at grid points.

Domain interpolation: The domain interpolation takes care of interpolating the data given at grid points to flow, depth and domain information at any location. The domain information describes whether the given location is within or outside of the domain, whether the grid cell it is contained in is an open or closed boundary or some region within the domain.

Particle initialisation: The purpose of this module is twofold; it generates the particles that exist at the start of the simulation, and it initialised the data associated with each particle, such as its weight, type, or for example history of a noise process driving the particle.

Particle manager: Memory for particles is allocated dynamically and therefore needs to be freed when it is no longer needed, to avoid running out of available resources. In order to avoid repeated allocation and freeing of particles, the particle memory recycles them and allocates new ones only if no recycled particles are available.

Integration: Provided with a list of particles, the current time and the duration of the time interval the integration module numerically applies the SDEs governing particle movement on the particles.

Particle suspension: At each iteration, this module determines how many particles to suspend in each grid cell. This is done according to the description given in chapter 4 Sec. 4.4.3. The particles are then created and initialised (with random location within the cell) and returned to the statistics engine.

Particle sedimentation: After integration, this module is given a list of all local particles and it determines which of those particles are subject to sedimentation (see chapter 4 Sec. 4.4.2). As such, the list is split into one of particles to keep and one of the particles to be deposited.

Statistics repository: The statistics repository collects information about, amongst others; the number of particles in the flow at each iteration, the particle tracks, and the amount of sedimentation and suspension in cells, The information is globally synchronised at the ending of the program and output to a data file.

5.5 Application

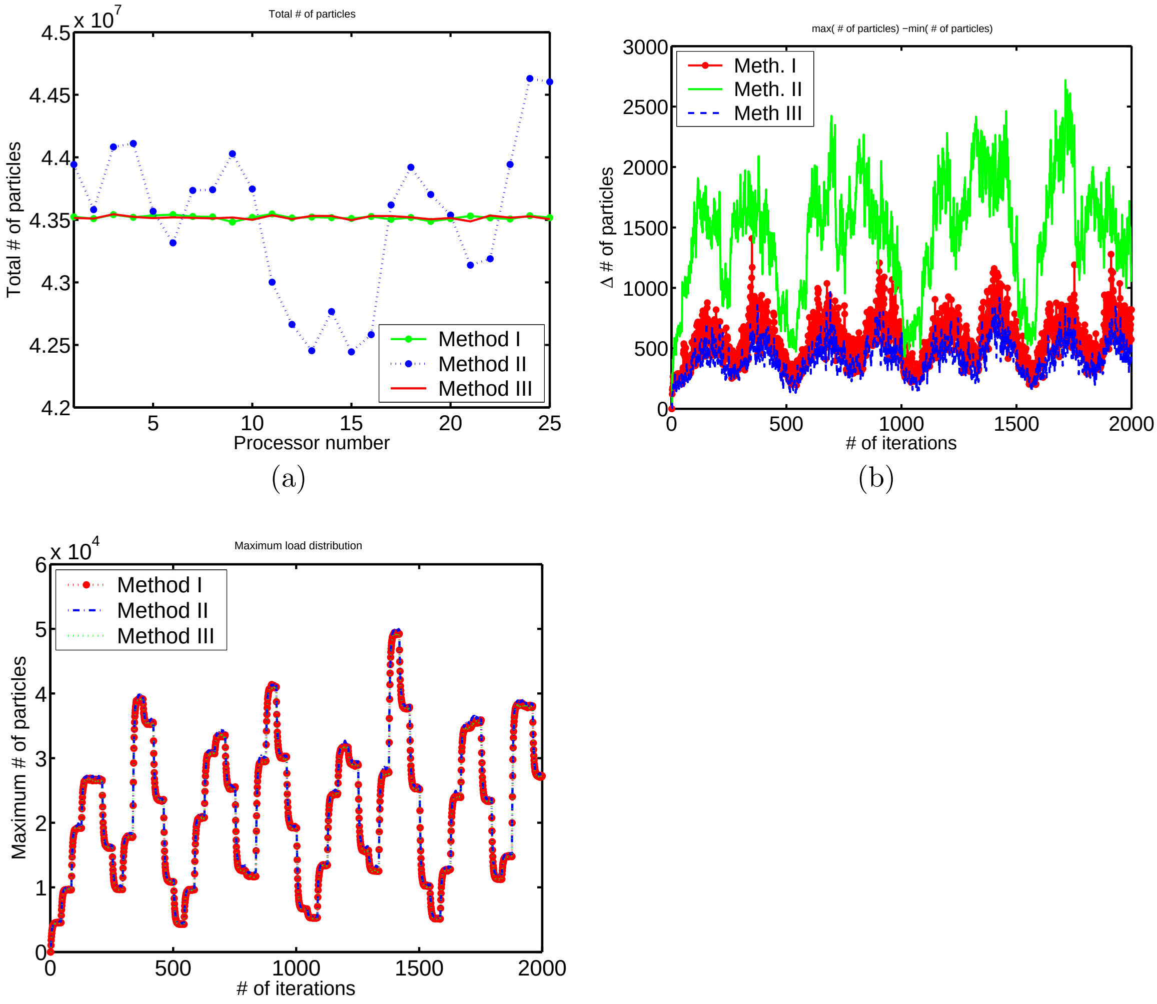
In this section, the Lagrangian particle model designed for the simulation of sediment transport is applied to the Dutch coastal waters in the Wadden sea (see Figure 5.7). The parallel processing experiments are carried out on a distributed memory parallel architecture called DAS-2 [1]. It is a 200-node wide-area distributed system spread out over five clusters throughout the Netherlands with a total of 400 processors. The system was built by IBM and runs on RedHat Linux. It is funded by NWO (the Netherlands organization for scientific research) and the participating five universities. Each node contains:

- Two 1-Ghz Pentium-IIIs of at least 1 GB RAM on each node.
- A Myrinet interface card connected with a high-level 3 switch.
- A Fast Ethernet interface (on-board).

The nodes within a local cluster are connected by a Myrinet-2000 network. It is used as a high-speed interconnection, mapped into user-space. In addition, Fast Ethernet is used as OS network (file transport). A Myrinet-2000 is a switched network, capable of full-duplex data rates up to 2+2 Gbits/s and has low latencies in range of a few micro-seconds.

5.5.1 Experimental results

In this section we carry out a number of experiments using realistic input data of the Wadden sea for the simulation of the sediment transport. The domain of grid size was 201×225 where each grid cell has $\Delta x = 800m$ and $\Delta y = 800m$. Many other simulation parameters are summarised in Table 5.1. In this article we have only used DAS-2 cluster based at the Delft University of Technology which has 32 nodes. The experiments have been carried out using one CPU for each node. For each processor we measure both the simulation and the total execution time. The speedup, efficiency and load balance for different suspension methods are measured using the minimum total execution times and their results are presented in Figure 5.5. The total number of particles in each processor summed over the iterations when 25 processors were used is shown in Figure 5.5(a). The difference between maximum and the minimum number of particles among the processors is determined at every iteration for each suspension methods (see Figure 5.5(b)). From Figure 5.5(b) it can be noticed that the suspension due to method III has relatively good load balance at every iteration. The load distribution in all three approaches at every iteration is almost the same (see Figure 5.5(c)). In addition, Figure 5.6(a) shows the parallel speedups achieved by the three methods. It can be seen that they are all close

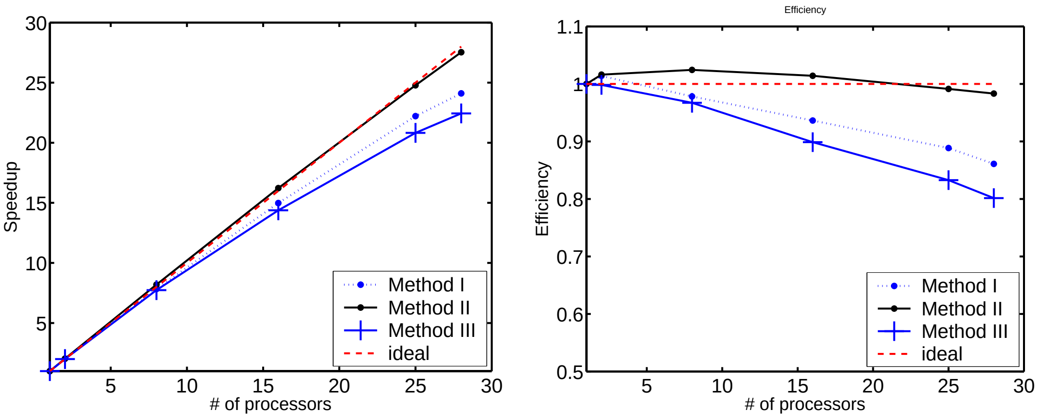


(a) (b)

(c)

Figure 5.5: (a) The total number of particles in each processor summed over all the iterations for each approach (b)The difference between the maximum and the minimum number of particles among the processors at every iteration for each approach;(c) The maximum number of particles in a processor at every iteration for each approach.

to the ideal linear speedup. Especially method II attained a very good speed up even for a large number of processors. Figure 5.6(b) shows the efficiency of the 3 methods. It is known that for morphodynamics processes, water flows play an important role which causes erosion and deposition of sediments resulting into bathymetrical changes.



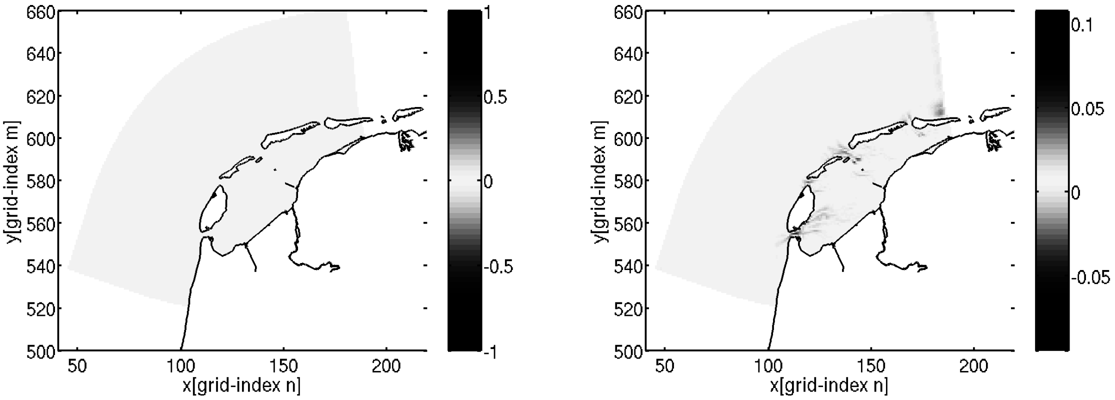
(a) (b)

Figure 5.6: (a) Parallel speedups obtained by the three sediment suspension methods (b) Efficiency of the three methods.

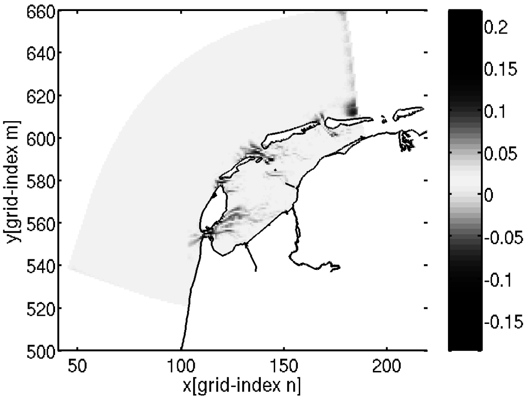
The simulation time scales can be categorised as long simulation time scale known as morphodynamic time scale, varies in (weeks, months or years. The second is known as a tidal time which varies in minutes, hours (see [15]), for example. In this chapter for practical purposes we use only data for 90 days which are stored on the disk. The experiment for bed level change in Figure 5.7 (a)–(c) was carried out using the same simulation parameters as those in the Table 5.1. Except that the mass of each particle $\mathcal{M}_p$ is now set to $10,000kg$, the probability of sedimentation $= 0.1062$, $\Delta t = 864s$ and the number of iterations $= 8999$ for 90 days.

5.6 Discussion and conclusions

In this chapter we developed a parallel Lagrangian particle model and applied it to the real life. The results look promising with good speed up for all three approaches. We have implemented three different suspension approaches in which the technique of globally synchronising random number generators has been introduced in the third approach. Computational experiments have been carried out and the parallel performances of the three methods have been measured. For instance method III has a very good load balance while methods I and II, seem to have relative a higher speed up than Method III. This is contributed by the fact that in method III random numbers are generated with larger expectation in all grid cells while method II selects only some grid cells and assign them to the available processors. Moreover, when you look at the balance of load



(a) Day 0 (b) Day 40



(c) Day 80

Figure 5.7: Sedimentation in the Dutch North sea in m over 80 days. The effect is most pronounced in the regions between the islands.

distribution, the method III clearly excels the two other approaches see Figure 5.5(a)–(b). Suspension process using method III leads to perfect load balance. However, the gain in method III by better load balance is offset by the more expensive suspension costs made in all grid cells. Nevertheless, the good speed up obtained in all three methods is due to the fact that unlike in the domain decomposition, communication is greatly reduced in the particle decomposition approach. There is no need to keep track of particles across the boundaries and to send to other processors. From the results of the studies we recommend the use of method II because it is the most efficient approach in the sediment transport model.

5.7 Summary of the simulation parameters

Constant	**Unit**	**Value**	**Constant**	**Unit**	**Value**
iteration	-	1999	Δt	s	86.4
porosity	-	0.40	γ	s^{-1}	0.00013
grid offset	km	$(40.55, 500)$	λ_s	$kg{\cdot}s{\cdot}m^{-4}$	0.0001
grid size	-	201×225	D	$m^2 \cdot s^{-1}$	3
cell size	m	800×800	δ	-	1.7
sand density	kgm^{-3}	2650	$\mathcal{M}_p$	kg	500
initial location	km	$(135, 570)$	$f = \frac{\lambda_s}{\gamma}$	$kg \cdot s^2 \cdot m^{-4}$	$\mathcal{O}(10^{-4})$
sedimentation prob.	-	0.0112			

Table 5.1: Simulation parameters of the parallel particle model for the sediment transport model.

Chapter 6

Variable time stepping in the parallel particle models for transport problems in shallow water

Overview[1]: The SDEs under consideration are often called particle models(PMs). PMs in this chapter models the simulation of transport of pollutants in shallow waters. The main focus of this chapter is the derivation and efficient implementation of an adaptive scheme for numerical integration of the SDEs in this chapter. The error determination at each integration time step near the boundary when the error of the scheme is dominated by the diffusion term is done by a pair of numerical schemes with strong order of convergence 1 and with strong order 1.5. When the error of the scheme is dominated by the deterministic drift term, we use the aforementioned order 1 scheme and another scheme of strong order 2. Based on the performance of the pair of schemes an optimal step size for a given error tolerance is estimated. Moreover, the algorithm is developed in such a way that it allows for a completely flexible change of the time step size while guaranteeing correct Brownian paths. The software implementation uses the MPI library and allows for parallel processing. By making use of internal synchronisation points it allows for snapshots and particle counts to be made at given times, despite the inherent asynchronicity of the particles with regard to time.

6.1 Introduction

Fixed step size implementations of numerical methods in traditional particle models have limitations. Moreover, the use of fixed small step sizes in the numerical approximation of SDEs may become unnecessary in case the error is very small and large time steps are

---

[1] Part of this chapter [12] has appeared in the proceedings of the Eighth International Conference on Modelling, Monitoring and Management of Water Pollution 4-6 September 2006 Bologna, Italy. The extension is submitted to the Journal of Parallel and Distributed Computing(JPDC).

needed. In this chapter, we have developed two pairs of adaptive schemes for pollutant transport in shallow waters, in each pair there is a scheme with lower and another with higher order. By evaluating the performances of the schemes it has been possible to decide whether higher orders or small time steps are necessary. We have developed three schemes for transport of pollutants namely an explicit order 1 strong scheme, an explicit order 1.5 strong scheme and an explicit order 2 strong scheme. In the shallow waters, the displacement of a pollutant is contributed by the drift and diffusion terms.

Therefore, when the error of the scheme is dominated by diffusion term in areas such as near the boundary, we use a pair of schemes consisting of an explicit order 1 strong scheme and an explicit order 1.5 strong scheme. While when it is dominated by the drift term both schemes become of order 1 strong. Consequently it is necessary to concentrate on the drift term because the effects of the diffusion term on the error of the scheme is hardly there. Therefore, we have been forced to look for a scheme with higher order strong. Hence, in regions where the error of the scheme is dominated by the drift term, we use an explicit order 1 strong scheme and an explicit order 2 strong scheme to evaluate the error. But we have developed the explicit order 2 strong scheme for transport of pollutants in the shallow waters in such a way that it concentrates on the error due to the deterministic drift. But it approximates the diffusion term in a very simple manner by using additive noise in the diffusion term (that is, the diffusion coefficient does not depend on space).

Thus, in the simulation of pollutant transport in shallow waters using SDEs, for example, smaller step sizes are needed to stably integrate in highly irregular areas. In such situations, it is advantageous to employ an adaptive scheme in the particle model. In their works [24, 6] introduced a variable time stepping procedure for the pathwise (strong) numerical integration of a system of SDEs.

The concept of adaptive schemes by mesh refining in Eulerian methods have been used in [55]. Particle models do not suffer from numerical diffusion in the source points [28, 4]. However, the use of an adaptive scheme may lead to high computational cost due to small step sizes or the large number of particles [35]. Fortunately particles are independent from one another, thus allow efficient use of parallel processing. Therefore, in this chapter we also implement parallel processing so as to speed up the computation.

This chapter is organised as follows, the general governing set of SDEs for transport of pollutants in shallow water is discussed in Section 6.2. The concept of the higher order strong adaptive schemes for transport problems in shallow water as well their implementations is discussed in Section 6.3. The procedure for determining the variable step sizes is described in section 6.3.4. The adaptive parallel implementation is described in section 6.4. The results appear in the Section 6.6.1. The concluding remarks are given in the Section 6.7.

6.2 The particle model for pollution transport

The displacement of pollutants in shallow waters is described by:

$$
\begin{aligned}
dX_t &\overset{\text{Itô}}{=} \left[U + \tfrac{D_{XX}(x,y)}{H}\left(\tfrac{\partial H}{\partial x}\right) + \tfrac{\partial D_{XX}(x,y)}{\partial x}\right] dt + \sqrt{2D_{XX}(x,y)}dW_n^x \\
dY_t &\overset{\text{Itô}}{=} \left[V + \tfrac{D_{YY}(x,y)}{H}\left(\tfrac{\partial H}{\partial y}\right) + \tfrac{\partial D_{YY}(x,y)}{\partial y}\right] dt + \sqrt{2D_{YY}(x,y)}dW_n^y
\end{aligned}
\tag{6.1}
$$

(X_t, Y_t) is the position of a particle, $(U, V)^{\mathrm{T}}$ is flow velocities and H is the total water depth. Wiener processes $W_n^x(t)$ and $W_n^y(t)$ are Gaussian [35].

In this chapter have modelled the dispersion process in shallow waters by designing the following dispersion functions where $D_{XX}(x, y)$, $D_{YY}(x, y)$ are the horizontal dispersion coefficient functions in the x and y direction respectively. We assume that, $D_{XY}(x, y) = D_{YX}(x, y) = 0$ in the dispersion tensor and that

$$
\begin{aligned}
D_{XX}(x,y) &= \frac{D11}{1 + e^{-(((x-xb)^2+(y-yb)^2)-K^2)}} \times \\
&\quad \left\{1 + \left([1 + e^{K^2}]\cos(\alpha) - 1\right) e^{-((x-x_b)^2+(y-y_b)^2)}\right\} \qquad (6.2) \\
D_{YY}(x,y) &= \frac{D22}{1 + e^{-(((x-xb)^2+(y-yb)^2)-K^2)}} \times \\
&\quad \left\{1 + \left([1 + e^{K^2}]\sin(\alpha) - 1\right) e^{-((x-x_b)^2+(y-y_b)^2)}\right\}. \qquad (6.3)
\end{aligned}
$$

The diffusion functions are designed in such a way $D_{XX}(x, y)$ and $D_{YY}(x, y)$ model the dispersion as realistic as possible. For instance the dispersion decreases near the boundary and no particle will cross the physical boundary. Where $D11$ and $D22$ are the horizontal dispersion parameters, $\underline{e}_1^T \underline{c} = \|\underline{e}_1\| \|\underline{c}\| \cos(\alpha), \quad \sin(\alpha) = \frac{|c_2|}{\sqrt{c_1^2+c_2^2}}$, with $\underline{e}_k$ the k^{th} column from the identity matrix, $\underline{c} = (c_1, c_2)^{\mathrm{T}}$. Finally, α is the assumed to be the angle between the boundary and x or y direction. Where $\underline{c}$ is a direction vector a long the side of a given boundary cell, (xb, yb) is intersection point on the boundary between the line from (x, y) perpendicular to the boundary. $K \geq 0$ is a parameter modelling the decrease of diffusion coefficient near the boundary.

6.3 Higher order strong adaptive scheme for pollution transport using SDEs

In this chapter we make use of the strong convergence in determining the error at each step. The concept of adaptive scheme in the SDEs has been extensively dealt with and published before by (e.g., [24, 6]). Therefore our contribution in this thesis to the adaptive schemes is that we have applied the adaptive concept for the SDEs to the shallow water problems. The transport of pollutants is contributed by the drift and diffusion term in the transport model. We use the Lagrangian particle-tracking approach in shallow waters. Moreover, in this chapter we have designed the diffusion coefficient which varies with space as it has been described in Section 6.2.

Therefore, in order to determine the error at each step, we have developed pairs of explicit higher strong numerical schemes for SDEs in shallow waters which avoid the use of derivatives of the drift and diffusion terms. And we use the pair of schemes to estimate the error of the schemes at each step. To achieve this procedure, we have developed three numerical schemes for transport of pollutant in shallow waters namely an explicit order 1 strong scheme, an explicit order 1.5 strong scheme and an explicit order 2 strong scheme [34]. When the particle is near boundary regions, in this case the error of the schemes is dominated by the diffusion term. In this case the error of the numerical scheme is mostly related to the gradient(slope) of the dispersion coefficient in that term. Therefore, to determine the error in this case we use two schemes one with an explicit order 1 strong and the explicit order 1.5 strong scheme. But in the region away from the boundary when the diffusion is not space dependent, the effects of diffusion term on the numerical error is hardly there. Consequently, the explicit order 1.5 strong becomes of order 1 strong, and hence both numerical schemes become of 1 order strong. In such regions the error of the schemes is dominated by the drift term. Therefore, at this point we need a numerical scheme with higher strong order such as explicit order 2 strong scheme. Therefore, in this case we have used a pair of schemes with an explicit strong order 1 and an explicit order 2 strong. Based on the performance of the pair of schemes we have been able to determine the error at each step. The error information at each step has been used to determine an optimal time step size for adaptive schemes in the Lagrangian particle tracking.

6.3.1 An explicit Order 1 Strong Scheme

In this section we have developed the following an explicit order 1 strong Scheme for transport of pollutants in shallow waters. This scheme avoids the use of the derivatives of various order of the drift and diffusion coefficient. Here we only give a brief overview of the schemes, an interested reader is referred to (e.g., [35, 34]). Now let us consider the following scheme:

$$\begin{aligned} X_{n+1} &\overset{\text{Itô}}{=} X_n + \left[U + \frac{D_{XX}(X_n, Y_n)}{H}\frac{\partial H}{\partial x} + \frac{\partial D_{XX}(X_n, Y_n)}{\partial x} \right] \Delta t_n \\ &+ \frac{\Delta (W_n^x)^2 - \Delta t_n}{2\sqrt{\Delta t_n}} \left[\sqrt{2D_{XX}(X_{n+1}^{*+1}, Y_{n+1}^{*+1})} - \sqrt{2D_{XX}(X_n, Y_n)} \right] \\ &+ \sqrt{2D_{XX}(X_n, Y_n)}\Delta W_n^x. \end{aligned} \tag{6.4}$$

The expression for Y_{n+1} is similar to the above equation, with the first r.h.s. term X_n replaced by Y_n, all $D_{XX}(\cdot,\cdot)$ terms by $D_{YY}(\cdot,\cdot)$, the superscripts x modified to y. Where $\Delta W_n^x = W^x(t_{n+1}) - W^x(t_n)$ is an independent increment of Wiener processes in the time interval $[t_n, t_{n+1}]$, $n = 0, 1, \cdots$.

$$X_{n+1}^{*+1} = X_n + a^1(X_n, Y_n)\Delta t_n + \sqrt{2D_{XX}(X_n, Y_n)\Delta t_n}.$$

Similarly for Y_{n+1}^{*+1} along y direction. A drift function a^1 is given below:

$$a^1(X_n, Y_n) \quad = \quad \left[U + \frac{D_{XX}(X_n, Y_n)}{H}\left(\frac{\partial H}{\partial x}\right) + \frac{\partial D_{XX}(X_n, Y_n)}{\partial x}\right], \tag{6.5}$$

likewise for a^2 along y direction:

$$a^2(X_n, Y_n) \quad = \quad \left[V + \frac{D_{YY}(X_n, Y_n)}{H}\left(\frac{\partial H}{\partial y}\right) + \frac{\partial D_{YY}(X_n, Y_n)}{\partial y}\right]. \tag{6.6}$$

6.3.2 An explicit Order 1.5 Strong Scheme

As for an explicit order 1 strong scheme, here we also derive derivative free scheme of order 1.5 according to [34]. Therefore, we have developed the following scheme for the transport of pollutant in shallow waters:

$$\begin{aligned}
X_{n+1} \overset{\text{Itô}}{=} \;& X_n + \left\{a_n^{1+}(X_{n+1}^{*+z}, Y_{n+1}^{*+z}) - a_n^{1-}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})\right\} \times \frac{\Delta t_n}{4}\left(R_{n,1}^x + \frac{1}{\sqrt{3}}R_{n,2}^x\right) \\
+ \;& \frac{\Delta t_n}{4}\left\{a_n^{1+}(X_{n+1}^{*+z}, Y_{n+1}^{*+z}) + a_n^{1-}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})\right\} + \sqrt{2D_{XX}(X_n, Y_n)}\Delta W_n^x \\
+ \;& \frac{1}{4\sqrt{\Delta t}}\left\{\sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})} - \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right\}\left[(\Delta W_n^x)^2 - \Delta t_n\right] \\
+ \;& \left\{\sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})} - 2\sqrt{2D_{XX}(X_n, Y_n)} + \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right\} \times \\
& \left\{\Delta W_n^x - \frac{1}{2}\left(R_{n,1}^x + \frac{1}{\sqrt{3}}R_{n,2}^x\right)\sqrt{\Delta t_n}\right\} \\
+ \;& \left[\sqrt{2D_{XX}(X_{n+1}^{*+\phi}, Y_{n+1}^{*+\phi})} - \sqrt{2D_{XX}(X_{n+1}^{*-\phi}, Y_{n+1}^{*-\phi})} - \sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})}\right. \\
& \left. + \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right] \times \frac{1}{4\Delta t}\left\{\frac{1}{3}(\Delta W_n^x)^2 - \Delta t_n\right\}\Delta W_n^x.
\end{aligned} \tag{6.7}$$

The expression for Y_{n+1} is similar to the above equation, with the first r.h.s. term X_n replaced by Y_n, all $D_{XX}(\cdot,\cdot)$ terms by $D_{YY}(\cdot,\cdot)$, the superscripts x modified to y for W and R, and similarly the $1+$ and $1-$ superscripts for a to $2+$ and $2-$. Using the shorthand notation of $\oplus$ for either $+$ or $-$, the following supporting vectors (used in equation 6.7) are defined

$$\begin{aligned}
X_{n+1}^{*\oplus z} &= X_n + \frac{1}{2}a_n^1(X_n, Y_n)\Delta t_n \oplus \sqrt{2D_{XX}(X_n, Y_n)\Delta t_n} \\
X_{n+1}^{*\oplus\phi} &= X_{n+1}^{*\oplus z} \oplus \sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})\Delta t_n}.
\end{aligned}$$

The expressions for Y_{n+1}^{*+z}, Y_{n+1}^{*-z}, $Y_{n+1}^{*+\phi}$, and $Y_{n+1}^{*-\phi}$ are again similar, with the X in the first r.h.s. term replaced to Y, a_n^1 replaced by a_n^2 and D_{XX} by D_{YY}. Consequently, using Eqn. 6.5 we get,

$$a^{1+}\left(X_{n+1}^{*+z}, Y_{n+1}^{*+z}\right) = \left[U + \frac{D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})}{H}\frac{\partial H}{\partial x} + \frac{\partial D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})}{\partial x}\right]$$

$$a^{1-}\left(X^{*-z}_{n+1}, Y^{*-z}_{n+1}\right) = \left[U + \frac{D_{XX}(X^{*-z}_{n+1}, Y^{*-z}_{n+1})}{H}\frac{\partial H}{\partial x} + \frac{\partial D_{XX}(X^{*-z}_{n+1}, Y^{*-z}_{n+1})}{\partial x}\right],$$

likewise for $a^{2+}\left(X^{*+z}_{n+1}, Y^{*+z}_{n+1}\right)$ and $a^{2-}\left(X^{*-z}_{n+1}, Y^{*-z}_{n+1}\right)$.
Where the $D_{XX}(\cdot,\cdot)$, $D_{YY}(\cdot,\cdot)$ approach zero toward the boundary and remain constant away from the boundary. Hence, the effects of the dispersion coefficient becomes small. Thus, we are confronted with the situation where the deterministic drift dominates the error of the scheme. As it has been discussed earlier, in this case, we need a pair of schemes of order 1 and higher strong order 2 of convergence, for example.

6.3.3 An explicit Order 2 Strong Scheme

The derivation of an explicit order 1 strong scheme and the scheme of order 1.5 which are derivative free is done by replacing the derivative with the finite difference [34]. This technique works well for low order explicit schemes, otherwise it results into complicated formulae as the order keeps increasing. Nevertheless, according to [34], in this chapter we have taken some advantage of the structure of the SDEs to avoid difficulties and complicated equations. Therefore, we assume that the SDEs has an additive noise, that is, the diffusion coefficient is not space dependent. In this way, it has been possible to develop a relatively simple explicit higher order scheme for transport of pollutant is shallow waters. Moreover the Itô and Stratonovich concepts coincides for additive noise. However, we have chosen to use Stratonovich concept in the explicit order 2 strong scheme for the additive noise to enable us discuss the Stratonovich concepts as well. Therefore, the following an explicit order 2 strong scheme has been developed:

$$\begin{aligned}
X_{n+1} &\overset{\text{Strat}}{=} X_n + \frac{1}{2}\left\{\underline{a}^1\left(X^{+}_{n+1}, Y^{+}_{n+1}\right) + \underline{a}^1\left(X^{-}_{n+1}, Y^{-}_{n+1}\right)\right\}\Delta t_n \\
&+ \frac{1}{\Delta t_n}\left\{\sqrt{2D_{XX}(t_{n+1})} - \sqrt{2D_{XX}(t_n)}\right\}\left\{\Delta W^x_n \Delta t_n - \Delta M^x_n\right\} \\
&+ \sqrt{2D_{XX}(X_n, Y_n)}\Delta W^x_n
\end{aligned} \qquad (6.8)$$

$$\begin{aligned}
Y_{n+1} &\overset{\text{Strat}}{=} Y_n + \frac{1}{2}\left\{\underline{a}^2\left(X^{+}_{n+1}, Y^{+}_{n+1}\right) + \underline{a}^2\left(X^{-}_{n+1}, Y^{-}_{n+1}\right)\right\}\Delta t_n \\
&+ \frac{1}{\Delta t_n}\left\{\sqrt{2D_{YY}(t_{n+1})} - \sqrt{2D_{YY}(t_n)}\right\} \times \left\{\Delta W^y_n \Delta t_n - \Delta M^y_n\right\} \\
&+ \sqrt{2D_{YY}(X_n, Y_n)}\Delta W^y_n.
\end{aligned} \qquad (6.9)$$

Here $D_{XX}(t) = D11$ and $D_{YY}(t) = D22$ are constants, so that the second line of the above two equations reduces to zero.

The supporting vectors are defined by

$$\begin{aligned}
X^{\oplus}_{n+1} &= X_n + \frac{1}{2}\underline{a}^1(X_n, Y_n)\Delta t_n \\
&+ \frac{1}{\Delta t_n}\sqrt{2D_{XX}(X_n, Y_n)}\left\{\Delta M^x_n \oplus \sqrt{2J^{x,p}_{(1,1,0)}\Delta t_n - (\Delta M^x_n)^2}\right\} \qquad (6.10) \\
Y^{\oplus}_{n+1} &= Y_n + \frac{1}{2}\underline{a}^2(X_n, Y_n)\Delta t_n \\
&+ \frac{1}{\Delta t_n}\sqrt{2D_{YY}(X_n, Y_n)}\left\{\Delta M^y_n \oplus \sqrt{2J^{y,p}_{(1,1,0)}\Delta t_n - (\Delta M^y_n)^2}\right\}, \qquad (6.11)
\end{aligned}$$

where $\oplus$ the plus or minus operator. The definition of $\underline{a}^1(X, Y)$ is obtained by using Itô-Stratonovich transformation [35] of Eqn (6.5), yielding

$$\underline{a}^1(X, Y) = \left[U + \frac{D_{XX}(X, Y)}{H}\left(\frac{\partial H}{\partial x}\right) + \frac{1}{2}\frac{\partial D_{XX}(X, Y)}{\partial x}\right].$$

$$\underline{a}^2(X, Y) = \left[V + \frac{D_{YY}(X, Y)}{H}\left(\frac{\partial H}{\partial y}\right) + \frac{1}{2}\frac{\partial D_{YY}(X, Y)}{\partial y}\right].$$

Higher order schemes such as that of order 2, require the approximation of multiple higher Stratonovich stochastic integral ($J^p_{(1,1,0)}$) (see Eqn. 6.12).

However, these cannot always be expressed in terms of simpler stochastic integrals, especially when the Wiener process is multi-dimensional. Using a method for multiple Stratonovich based on Karhunen-Loève or random Fourier series expansion of the Wiener process (for details, see [35]) we can nevertheless approximate the integrals. This introduces a Brownian bridge into our model, a process fully described in [35]. The Brownian bridge is a restricted Wiener process (hence also referred to as the “tied down” Wiener process) that passes through known points at $t = 0$ and $t = T$ and is given by $\left\{W_t - \frac{t}{T}W_T, 0 \leq t \leq T\right\}$. This can be done by generating an unconstrained (standard) Wiener process which is then linearly scaled in order to meet the required end points. A sample path of a Brownian bridge which is tied down to zero at beginning and at the end points is shown in Figure 6.1.

Following Karhunen-Loève see [35] we define the random variables a^x_r and b^x_r by

$$\begin{aligned}
a^x_r &= \frac{2}{\Delta t}\int_0^{\Delta t}\left(W^x_s - \frac{s}{\Delta t}W^x_{\Delta t}\right)\cos\left(\frac{2r\pi s}{\Delta t}\right)ds \\
\text{and}\quad b^x_r &= \frac{2}{\Delta t}\int_0^{\Delta t}\left(W^x_s - \frac{s}{\Delta t}W^x_{\Delta t}\right)\sin\left(\frac{2r\pi s}{\Delta t}\right)ds, \qquad r = 1, 2, \ldots
\end{aligned}$$

and likewise a^y_r and b^y_r, obtained by replacing the x superscripts by y. (In the remainder of this section we will silently assume this convention, unless otherwise specified). It is known that, for $r \geq 1$ these variables have an $\mathcal{N}\left[0, \frac{\Delta t}{2\pi^2 r^2}\right]$ distribution(see appendix B). They are differentiable samples paths on the interval $[0, T]$.

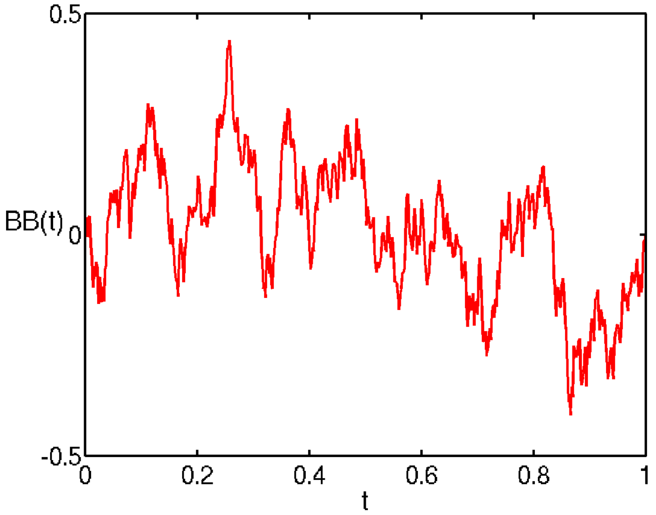


Figure 6.1: A sample path of Brownian bridge.

Let $\zeta_r^x, \xi^x, \zeta_r^y, \xi^y, \eta_r^x, \eta_r^y, \phi_p^x$, and ϕ_p^x denote independent random variables [35], for $r = 1, 2, \ldots$ and $p = 1, 2, \ldots$:

$$\begin{aligned} \xi^x &= \tfrac{1}{\sqrt{\Delta t}} W_{\Delta t}^x & \zeta_r^x &= \sqrt{\tfrac{2}{\Delta t}}\pi r a_r^x & \eta_r^x &= \sqrt{\tfrac{2}{\Delta t}}\pi r b_r^x \\ \mu_p^x &= \tfrac{1}{\sqrt{\Delta t \rho_p}} \textstyle\sum_{r=p+1}^{\infty} a_r^x & \phi_p^x &= \tfrac{1}{\sqrt{\Delta t \beta_p}} \textstyle\sum_{r=p+1}^{\infty} \tfrac{1}{r} b_r^x \\ \mu_p^y &= \tfrac{1}{\sqrt{\Delta t \rho_p}} \textstyle\sum_{r=p+1}^{\infty} a_r^y & \phi_p^y &= \tfrac{1}{\sqrt{\Delta t \beta_p}} \textstyle\sum_{r=p+1}^{\infty} \tfrac{1}{r} b_r^y. \end{aligned}$$

Variance of $\hat{\mu}_p^x = \sqrt{\Delta t \rho_p}\mu_p^x$ can be computed by noting that the variance of a_r^x is given by $\text{var}[a_r^x] = \Delta t/(2\pi^2 r^2)$ (see appendix B), and that the variance of two independent Gaussian variables equals the sum of variances and with the fact that $\sum_{r=1}^{\infty} 1/r^2 = \pi^2/6$ and $\sum_{r=1}^{\infty} 1/r^4 = \pi^4/90$.

$$a_0^x = -\frac{1}{\pi}\sqrt{2\Delta t}\sum_{r=1}^{p}\frac{1}{r}\zeta_r^x - 2\sqrt{\Delta t \cdot \rho_p}\mu_p^x, \qquad \rho_p = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^2}.$$

Using the definition of a_r^x, a_r^y, and for each component and $r = 1, \ldots, p$ with $p = 1, 2, \ldots$, where p is the truncation index in the approximation of multiple integrals. We then define

$$B^x = \sqrt{\frac{\Delta t}{2}}\sum_{r=1}^{p}\frac{1}{r^2}\eta_r^x + \sqrt{\Delta t \beta_p}\phi_p^x, \qquad \beta_p = \frac{\pi^2}{180} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^4}.$$

Furthermore, we have

$$\Delta M_n^x = \tfrac{1}{2}\Delta t\left[\sqrt{\Delta t}\xi^x + a_0^x\right] \qquad C_{x,x}^p = -\tfrac{1}{2\pi^2}\textstyle\sum_{r,l=1_{r\neq l}}^{p} \tfrac{r}{r^2-l^2}\left\{\tfrac{1}{l}\zeta_r^x\zeta_l^x - \tfrac{l}{r}\eta_r^x\eta_l^x\right\}$$

and similar for ΔM_n^y and $C_{y,y}^p$ and with superscripts changed from x to y. Using these random variables it turns out after lengthy computations that we can approximate a multiple integral as follows

$$\begin{aligned} J_{(1,1,0)}^{x,p} &= \tfrac{1}{6}(\Delta t)^2(\xi^x)^2 + \tfrac{1}{4}\Delta t(a_0^x)^2 - \tfrac{1}{2\pi}(\Delta t)^{\frac{3}{2}}\xi^x B^x \\ &+ \tfrac{1}{4}(\Delta t)^{\frac{3}{2}}a_0^x\xi^x - (\Delta t)^2 C_{x,x}^p, \end{aligned} \tag{6.12}$$

$$\begin{aligned} J_{(1,1,0)}^{y,p} &= \tfrac{1}{6}(\Delta t)^2(\xi^y)^2 + \tfrac{1}{4}\Delta t(a_0^y)^2 - \tfrac{1}{2\pi}(\Delta t)^{\frac{3}{2}}\xi^y B^y \\ &+ \tfrac{1}{4}(\Delta t)^{\frac{3}{2}}a_0^y\xi^y - (\Delta t)^2 C_{y,y}^p, \end{aligned} \tag{6.13}$$

$J_{(1,1,0)}^{x,p}$ is an approximation of $J_{(1,1,0)}^{x}$ and it is known [35] that $J_{(1,1,0)}^{x} \geq \frac{(\Delta M^x)^2}{2\Delta t_n}$ always. If it turns out $J_{(1,1,0)}^{x,p} < \frac{(\Delta M^x)^2}{2\Delta t_n}$, we take ΔM^x as the better approximation for $J_{(1,1,0)}$ (see Appendix B.1). Similarly for $J_{(1,1,0)}^{y,p}$ and finally the evaluations of Eqns (6.8)-(6.9) can take place.

6.3.4 Determination of variable time step sizes

By approximating the error of the scheme at each step we use a pair of scheme as explained in Section 6.3. Therefore, let us assume that $(\hat{X}_{n+1}, \hat{Y}_{n+1})$ is the approximated solution obtained from the SDEs (6.1) by using the explicit order 1 strong scheme (6.4). We also use the explicit order 1.5 strong scheme (6.7) when the same particle is tracked in the near boundary regions. The approximated solution due to the explicit order 1.5 strong scheme is denoted by $(X_{\mathrm{ref}_{n+1}}, Y_{\mathrm{ref}_{n+1}})$. But when the particle is away from the boundary, the error of the scheme is dominated by the drift term. Therefore, in this case we use explicit order 1 strong scheme (6.4) and the explicit order 2 strong scheme (6.8)- (6.9) to determine the error at each step. Again $(X_{\mathrm{ref}_{n+1}}, Y_{\mathrm{ref}_{n+1}})$ becomes the approximated solution due to the explicit order 2 strong scheme. Therefore, we have chosen to represent an approximated solution of the SDEs due to higher order strong scheme by $(X_{\mathrm{ref}_{n+1}}, Y_{\mathrm{ref}_{n+1}})$. This solution is due to a reference scheme that is the scheme with a higher order strong in the pair. In both cases it is the reference scheme which is used to advance the numerical computation in the next time step.

Therefore, $(\hat{X}_{n+1}, \hat{Y}_{n+1})$ and $(X_{\mathrm{ref}_{n+1}}, Y_{\mathrm{ref}_{n+1}})$ is used to estimate absolute error [6]. Let tol_i be the tolerance accepted for the ith component then an error estimate of order $q + \frac{1}{2}$ in two-dimensional adaptive particle model is:

$$error = \sqrt{\frac{1}{2}\left(|\frac{X_{\mathrm{ref}_{n+1,1}} - \hat{X}_{1n+1,1}}{tol_1}| + |\frac{Y_{\mathrm{ref}_{n+1,2}} - \hat{Y}_{1n+1,2}}{tol_2}|\right)}, \tag{6.14}$$

where q is considered to be either order $\hat{o}$ or order o. Therefore, it is desirable that $X_{\text{ref}_{n+1,1}} - \hat{X}_{n+1,1} \approx tol_1$ and $Y_{\text{ref}_{n+1,2}} - \hat{Y}_{n+1,2} \approx tol_2$, the step just completed is rejected if $error > 1$ otherwise we compute an optimal step size $(\Delta t)_{opt} = \Delta t_{old} \left(\frac{1}{error}\right)^{\frac{1}{2}}$ until the desired accuracy is attained. For efficient implementation of an adaptive scheme using a variable step size strategy, an optimal step size can be decreased by any safety factor for example 0.8 to avoid oscillatory behaviour in the step size so that it does not increase or decrease too quickly [6]:

$$(\Delta t)_{new} = \Delta t_{old} * \min\left(facmx, \max\left[facmn, fac * \left(\frac{1}{error}\right)^{\frac{1}{2}}\right]\right) \tag{6.15}$$

where $facmx$ and $facmn$ are the maximal and minimal step size scaling factors allowed, respectively for the problems being solved [6]. Variable step size implementation has a possibility of step size acceleration using Eqn. (6.15). This arises when a step fails, possibly due to extreme random sample, in this chapter, we avoid uncontrolled jumps in the step size such that the final step length is given by

$$\Delta t_n = \max\left((\Delta t)_{new}, 0.9 * \Delta t_{n-1}\right).$$

6.4 Variable time stepping implementation in SDEs

The implementation of adaptive scheme differs substantially from the one with a fixed step size in that it is no longer possible to have a single major loop governing the time by taking a single step of fixed size [37]. Instead, the current time differs between the particles and, in addition to the coordinates, each particle now needs a local time associated with. This concept of local time introduces a wide level of asynchronicity into the model, making it hard to define a major loop in the traditional way. Additionally, this lack of synchronous time complicates taking a snapshot of the particle locations at a given time. One option would be to record the last position before and the first after the time at which to generate this data and then use some form of interpolation to estimate the location at that time. However, besides having to perform checks at each integration step and keeping track of the most recent locations, this also lessens the accuracy of the results.

To overcome these difficulties introduced an event mechanism which defines certain synchronisation points. The implementation consists of a number of different modules (see Figure 6.2), each taking care of a certain function within the program. There are for example a statistics module, which takes care of gathering all the information we need during the simulation, a domain module whose function it is to load and manage the flow data at different times and an integration module which does the integration. Now, each module provides the central engine with a list of desired events consisting of the time(s) at which they should occur, a type and possible some additional data. The engine sorts these events in increasing activation time and then sequentially considers the next event to happen (similar to the time stepping look in the fixed step size model).

It then invokes the integration module with the present time and the time to integrate to. The integration routine will then perform the integration and is completely free to decide how this time interval is integrated. It will ensure however that each particle is exactly integrated up to the desired ending time, coinciding with the event, unless of course the particle flows out of the domain before that. This way, the result of the integration call is a set of particles, with their location at exactly the time of the event. (To be more precise, the integration routine is given a single particle at a time and returns the trajectory the particle followed during the time spanned by the integration). Coming back to the events themselves, the statistics module for example generates snapshot events, the domain module provides reload flow data events and the integration module could generate events at regular times to limit the interval time span. When the total integration time is large, one could imagine that creating all these events at the start of the simulation would be too costly in terms of memory (and perhaps sorting). Therefore each module provides event generators, rather than individual events. All the generators have to do is provide the time of the first event due allowing them to be sorted. When the given time is reached the generator is asked to issue that event. If there is any event left it is sorted again according to the time of the next event it provides. This way the events are handled in the correct order as if they were all generated a priori. The main program itself also generates an event telling the main loop to stop at the desired time.

The particle model lends itself extremely well for parallel processing, since the particles do not interact with one another and can therefore be considered on an individual basis. By dividing the particles, instead of the domain, across the processors we take full advantage of this.

6.5 Schematic summary of the adaptive particle model

The following briefly describes the summary of the adaptive particle model in diagram for transport problems in shallow water as described in Section 6.4. The implementation consists of a set of modules, shown in Figure 6.2, working together through predefined interfaces, simplifying replacement of one model implementation by another. For example, the choice of a particular integrating scheme to use does not affect the other modules.

1. Hydrodynamic model e.g WAQUA: Determines the flow and depth data.
2. Particle manager: Recycles the data structure associated with a particle (this avoid having to repeatedly allocate and free memory for them).
3. Particle source.
    - Generates the state of the initial particles (if any).
    - Generates particles flowing in from the open domain boundaries.
4. Statistics repository is responsible for gathering and maintaining information about

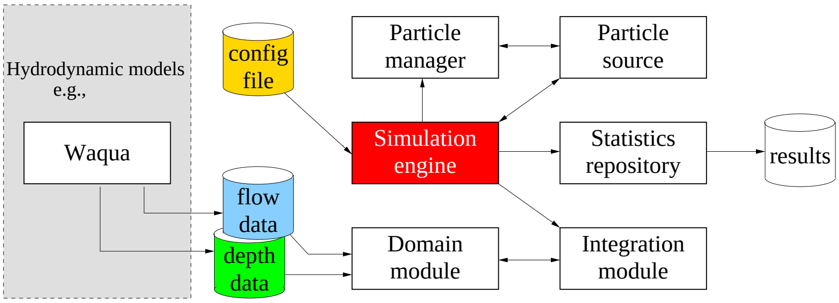


Figure 6.2: An overview of the software for the parallel adaptive particle model for pollution dispersion in shallow water.

- the number of particles in the flow at each iteration,
- the particle tracks,
- Synchronise this information prior to ending the program,
- Output to one (Matlab) file,
- etc.

5. Domain module
    - Triggered by the events to prepare data at a required time t.
    - Loads and caches the data for the file including that time.
    - Provides raw information of flow and depths at grid points.
    - Interpolates the data to any given position.
    - Provides information about whether a particle is inside the domain or outside.
6. Integration module
    - Integrates the equations governing particle movement for a given particle up to a certain point time t.
    - Is free to determine the best time steps.

6.6 Experiments of adaptive particle model parallel processing

In this section, the experiments to predict the dispersion of pollutants are carried out on a distributed memory parallel architecture called DAS-2 [1]. It is a 200-node system with a total of 400 -processors wide-area distributed system. More details about DAS2 can be found in [1].

To carry out the experiments, we started by doing the simulation by using a single processor. Where a given number of 2000 particles was released in a fixed initial location and tracked. The next step for the experiments involved 100000 particles in parallel processing where we used up to 30 processors to measure the speedup. The experiments were first done by using a test case domain followed by the realistic application of the model in the Dutch coastal waters.

6.6.1 Results of the test case

This section discusses the results obtained from the test case. The results in Figure 6.3 (a)-(c) were computed by using one processor in the ideal domain composed of the river, the lake and two islands as well as outflows. In these experiments 2000 particles were initially released at the point $(-20000m, -1800m)$. The result in Figure 6.3 (b) shows that the integration time step sizes for this particular tracked particle vary depending on the nature of the region. The variable dt varies as the particle is tracked along x direction. It appears that the value of dt is small when the particle approaching to the boundaries of the two islands. This is because the error of the scheme at that regions is dominated by the diffusion term mainly by gradient of the diffusion. Therefore, small dt is needed to integrate the particle over there. But we have seen that the value of dt for this track is about twice the minimum dt when the particle is away from the boundary, the error of the scheme is dominated by the drift term.

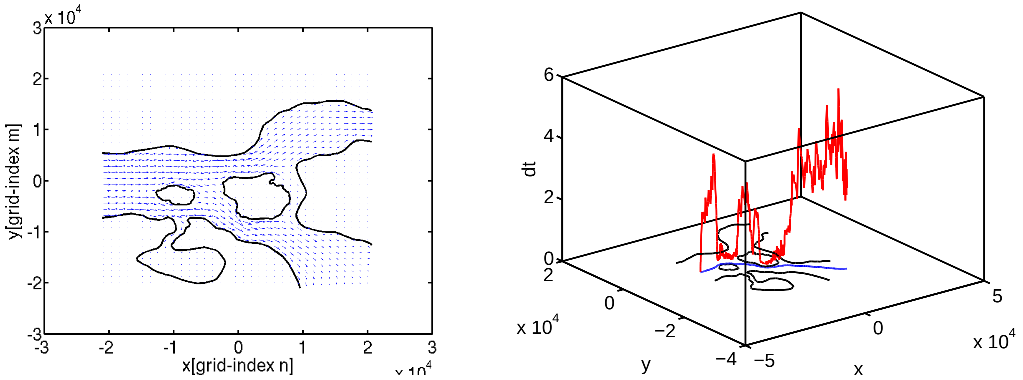
In order to speed up the computation of the adaptive scheme, the parallel processing was implemented. The experiments to measure the total simulation time was measured on the Beowulf cluster called DAS2. In these experiments we initially released $100,000$ particles.

These particles were tracked by using Lagrangian approach through parallel computing where up to 30 processors were used. The measured speedup attained for a fixed problem was 29.3810 which is quite good (see Figure 6.3 (d)).

Where the grid size 105×105, $tol1 = tol2 = 12$, minimum Δt=$0.0001s$, initial $\Delta t = 0.1s$, $p = 10$, $D11 = D22 = 10m^2/s$, initial point $(x_0, y_0) = (-20000m, -1800m)$, $\Delta x = \Delta y = 400m$, $H(x, y) = 10m$ assumed constant for now. $fac = 0.8$, $facmin = 0.6$, $facmax = 1.1$, $K = 1m$. Radius =3, stands for the number of grid rings surrounding the threshold point at each side. Snap shot is taken at every 5 (see minutes Figure 6.3(c)), eventually particles flow out. Boundary threshold distance= $1000m$ is the point where the two schemes of order 1.5 and order 2 exchange, 30 Brownian bridge steps have been used in this experiment.

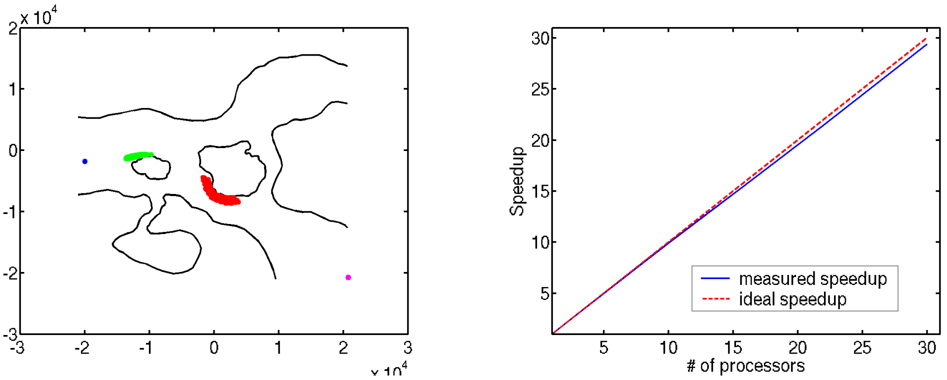
6.6.2 Results from the application of the model to the real data

In this section the adaptive particle model for the pollutants dispersion used the real data of the flow field and water depths. Initially 500 particles were released at the fixed



(a) Flow field

(b) Variation of dt along a track



(c) Snap shot taken after 5 minutes

(d) Speedup

Figure 6.3: Simulation results (a) flow fields (b) Variations of step sizes long the domain (c) Snap shot of particles' position at every 5 minutes (d) Speed up measured on a Beowulf cluster.

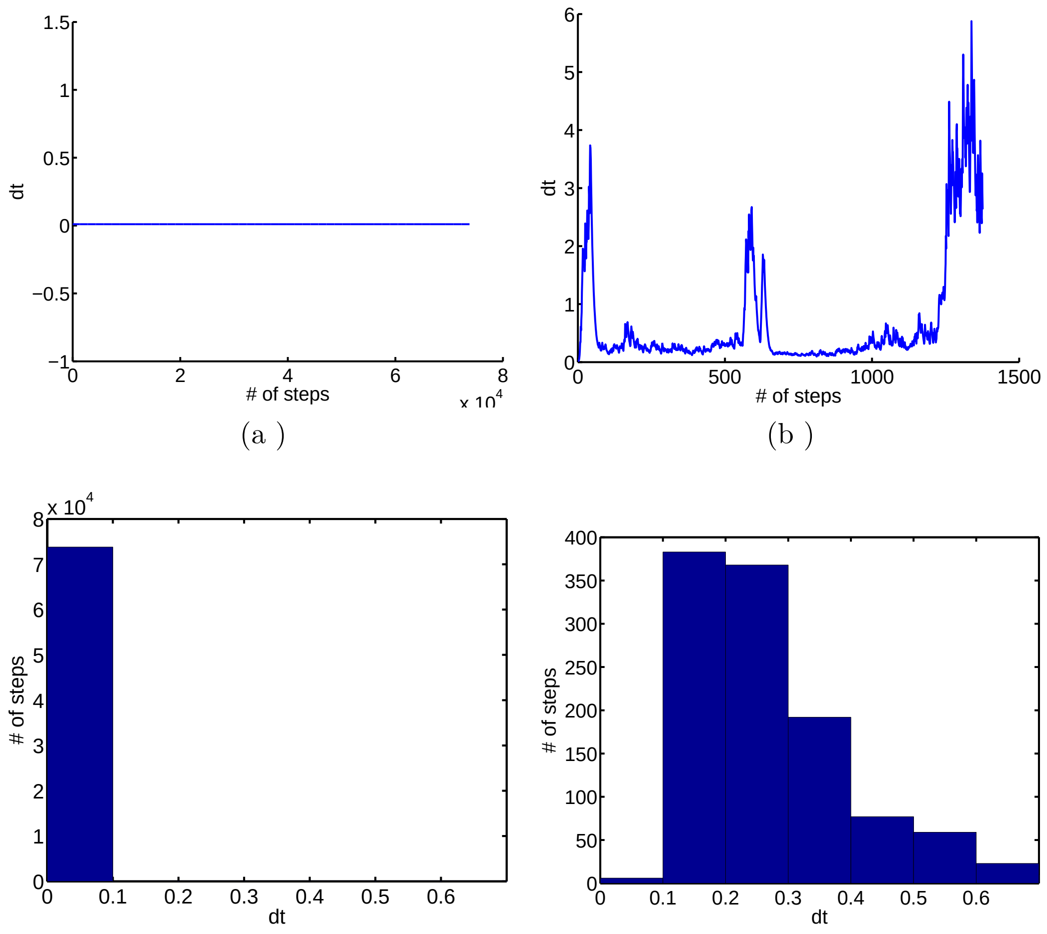
point $(135000, 570000)$. The grid offset $= (40550m, 500000m)$, grid size $= 201 \times 225$, cell size $= 800m \times 800m$, Diffusion Coefficient $D = 3m^2 \cdot s^{-1}$. But one processor was used for the experiments whose results are shown in Figure 6.5 (a)-(c) and Figure 6.6(a)- (d). The results in Figure 6.5 (a,b) show that the integration time step sizes for selected tracked particle vary depending on the nature of the region. Figure 6.5 (a) shows variable step sizes of a tracked particle along x direction while Figure 6.5 (b) shows variable time step sizes of the same tracked particle over all iterations in using realistic data. The results of the snap shot for positions of the particles taken at an interval of $20000s$ are shown in Figure 6.6(a)- (d).

The averaged variable dt $= 0.6971s, 0.6474s, 0.5677s, 0.6193s, 0.6858s$, for 5 tracks. For each of the 5 tracks using fixed time step, the averaged $dt = 0.01s$. The data such as x, y, dt etc were written to the file over the number of samples for successful steps and after each integration step, a new location was added to the track list. The table 6.1 summarizes the results. By looking at the results in table 6.1, for sure we conclude that the use of the variable *dt* leads to low computation costs. More number of steps are needed for fixed but small *dt* (see Figure 6.4(a,b)).

Fixed dt	**values**	**Variable dt**	**values**
# of steps	73797	# of steps	1376
dt minimum	$0.01s$	*dt* minimum	$0.01s$
average *dt*	$0.01s$	average *dt*	0.6971
Total sim. t	$523.933567s$	Total sim. t	$1.099058s$
seed number	100	seed number	100
# of tracks	5	# of tracks	5

Table 6.1: Summary of the simulation data using fixed dt and variable dt in the shallow waters using the Lagrangian approach

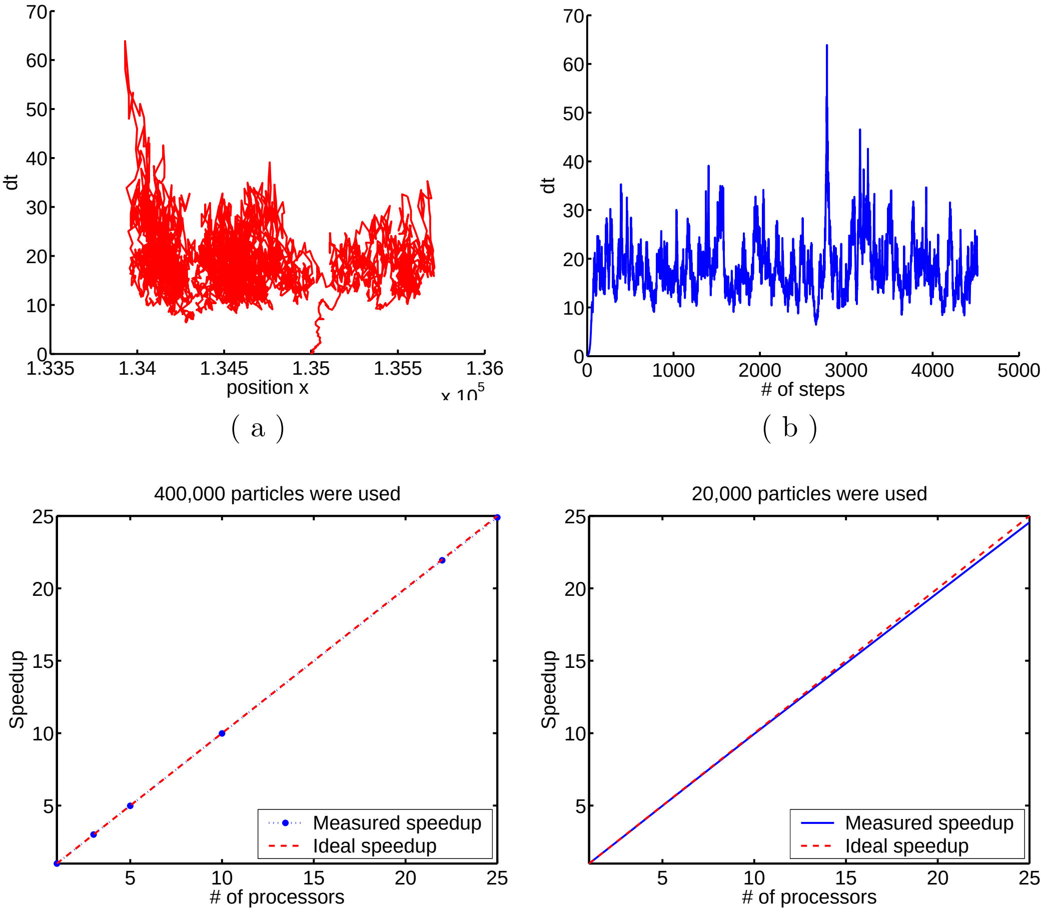
For parallel computing we first used 400,000 where the speedup and efficiency were measured by using up 25 processors. The speedup of 24.8975 was obtained, this was quite super linear (see Figure 6.5(c)). The result in Figure 6.5 (c) suggests that for large enough numbers of particles and long enough simulation times, the stochastic nature of the model furthermore automatically guarantees that a good overall load balance over the processors is maintained, even though at given times some processors may be busier than others. Another experiment was carried out but with 20000 particles, the speedup of 24.5555 was attained (see Figure 6.5(d)). The two cases show that the speedup scales very good, though in the second case it is a bit smaller. However, for a fixed problem the communication overhead grows linearly with the number of processors.



(a) (b)

(c) (d)

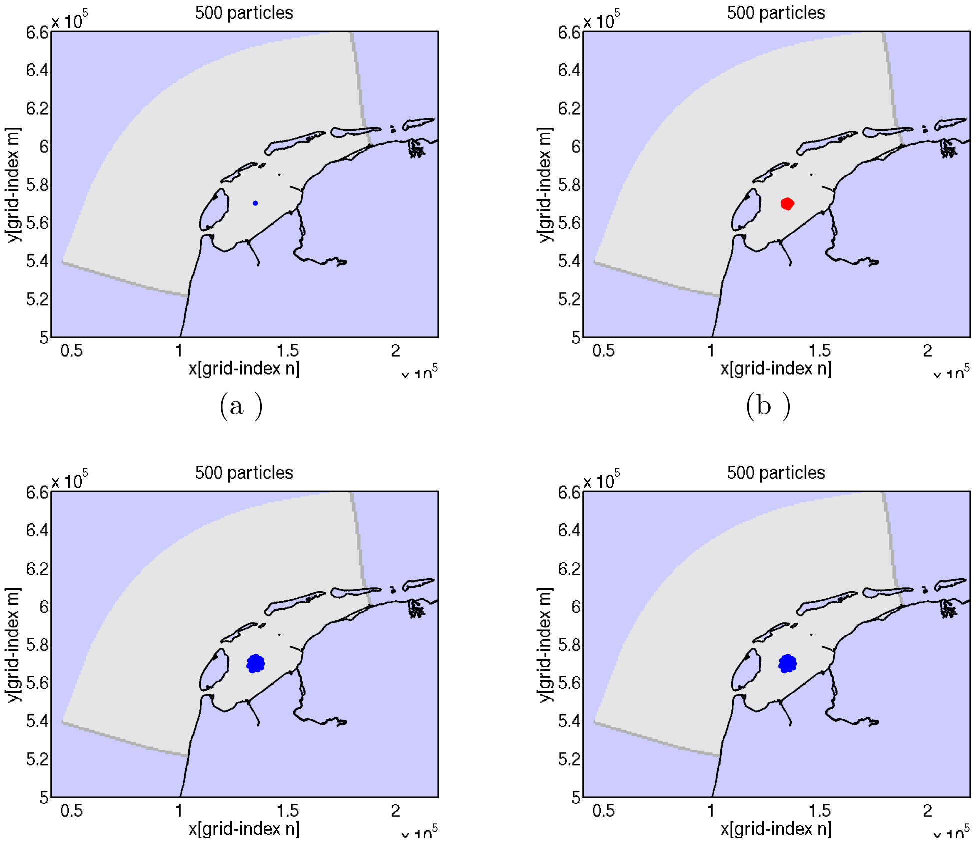
Figure 6.4: (a) Fixed dt over # of steps for a single track 1 *(b) Fixed dt over # of steps for a single track (c) The distribution of the number of # of steps for single track using fixed dt (d) The distribution of the number of # steps for single track using variable dt*



(a) (b)

(c) (d)

Figure 6.5: Simulation results (a) variations of step sizes at different locations along x direction (b) variations of step sizes at different locations along y direction (c) variations of step sizes over all iterations (d) Speed up measured on a Beowulf cluster.



(a) (b)

(c) (d)

Figure 6.6: (a) Snapshot taken after 0 second (b) Snapshot taken after 20000 seconds (c) Snapshot taken after 40000 seconds (d) Snapshot taken after 60000 seconds

6.7 Concluding remarks

In this chapter an adaptive scheme for the parallel simulation of pollutant transport in shallow waters using SDEs has been implemented. We have seen that smaller step sizes are needed to stably integrate in highly irregular areas and vice versa (see Figure 6.3(b), for the test case and Figure 6.5(a)-(b) for a more realistic case. In the adaptive particle model it is also possible to synchronise points in time. In this way it has been possible to take the snap shot of the position of particles at various time in the Dutch coastal waters (see Figure 6.6). Good speed up has been obtained despite the fact that each particle has different *dt*, the use of large number of particles averages the randomness. Furthermore, we have seen that the required computing time is not only reduced by the use of parallel computers, but also the use of variable *dt* in the particle model.

Chapter 7

General conclusion and recommendations

7.1 General conclusions

In this thesis we have developed and presented an efficient particle model for modelling of transport problems in shallow waters. This particle model is based on the two dimensional stochastic differential equations. It calculates the displacement as the sum of a drift deterministic component and a random Markovian component. The random part models the uncertainty by using the Wiener or coloured noise processes. The traditional particle models usually are driven by Wiener process. But the Wiener process has independent increments and it fails to consider accurately the short term spreading behaviour, that is for $t << T_L$. In order to take into account correctly the short term spreading of pollutants, we have used a correlated process known as the coloured noise as the driving force in the random flight model in chapter 3 instead. The results show that when using coloured noise, we have two zones of dispersion, one zone with the variance of the spreading of a cloud of particles growing quadratically with time for $t \ll T_L$ and the second zone with the variance of the spreading of particles growing linearly with the time for $t \gg T_L$. A random flight model will provide the modeller with an enhanced tool for the short term simulation of the pollutants by providing more flexibility to account for correlated physical processes of diffusion in the shallow waters for $t << T_L$.

Furthermore, in chapter 4 we have modified the particle model which was developed by Heemink in [28] for pollution dispersion in coastal waters. Part of the research here was to modify that particle model and develop a particle model for sediment transport in shallow water. This was achieved by adding extra equations for the erosion and deposition terms using probabilistic concepts. The developed model was also applied in the Dutch coastal waters to simulate sediment transport. The results which we have obtained depict that sedimentation processes are active in the channels and shallow water but not in the deep water. This may have been contributed by the fact that there is a strong flow in shallower regions.

It is well known that, in order to get accurate results from Monte Carlo simulations of pollution dispersion or sediment transport using particle models, a large number of particle is needed. Consequently, the computation time in a particle model increases linearly with the number of particles. On the other hand, as memory requirements grow this too may be a limiting factor. A nice characteristic of the particle model is that particle tracks are inherently independent of one another, and is possible to compute these in parallel [37]. In chapter 5 we have used a parallel processing approach where particles were divided into groups almost equally and assigned to each processor in the particle model. By doing so it is possible to significantly reduce the simulation time as well as to alleviate memory problems and consequently good speedup has been attained by running our experiments on a Beowulf clusters known as DAS2.

It is known that particle-based approaches are generally trivially parallelisable because they are independent of each other. Therefore, allows each domain to maintain a relatively good load balance since the particles could be freely distributed among the different processors in the system. Therefore, from the point of view of parallel computing the implementation of a parallel simulation of particle transport alone may not be considered significant enough. Nevertheless, the problems of the simulation of sediment transport studied, have a number of appealing features for parallel computing:

- efficient generation and elimination of particles in the domain due to erosion and deposition is required.

- handling of large amounts of the flow data e.g., n months × 24 samples per day for each grid point require an efficient particle model.

In general, the implementation of the particle model often uses fixed step sizes for Lagrangian particle-tracking down the stream in the domain. But using fixed time step sizes can lead to unnecessary higher computational costs. Therefore, in chapter 6 we have introduced the variable time step sizes in our particle model. To achieve this we have devised a criteria of error computations for handling the error at each step and use this information to determine the optimal step size. This has been achieved efficiently by using a pair of schemes with higher strong order. Furthermore, we also designed the diffusion coefficient functions that are spatially varying. After testing the adaptive scheme in an ideal domain, eventually, we have implemented it for the variable step sizes for the real life modelling of pollutants dispersion in Dutch coastal waters. The results show that for Lagrangian particle-tracking in shallow water it is vital to use the variable time step sizes. Smaller step sizes were observed when tracking a particle in an irregular region of the domain.

One major problem in the adaptive scheme could be how to synchronise the integration time of the particles. But we have designed an adaptive scheme in such a way that it is possible at certain time to synchronise the integration time of all particles. Consequently, the taking of snapshots and particle counts at given times have been possible, despite of the inherent asynchronicity of the particles with regard to time.

7.2 Recommendations

There are a number of interesting topics yet to be done. In the future, there should be a fully implementation of the adaptive scheme in the three dimensional particle model for transport problems in shallow waters. For sediment transport a fully three dimensional particle model should be used. Moreover, the three dimensional particle model for sediment transport should be coupled with the flow model to enable the simulation of the depth changes in a long time simulation. That is of course the best way but very expensive in terms of computing time. Nevertheless, the depth changes take place slowly, therefore we recommend not to determine/communicate the depth changes at every iteration but accumulate the depth changes and communicated them after each certain period interval of relatively long time . Of course the level of coarseness (intervals) in the depth updates determines the overall accuracy.

Bibliography

[1] The distributed ASCI supercomputer home page. *www.cs.vu.nl/das2*, 2005.

[2] P. S. Addison, Bo Qu, A. Nisbet, and G. Pender. A non-Fickian particle-tracking diffusion model based on fractional Brownian motion. *International Journal for numerical methods in fluids*, 25(0):1373–1384, 1997.

[3] L. Arnold. *Stochastic differential equations: Theory and applications.* Wiley, London, 1974.

[4] R. W. Barber and N. P. Volakos. Modelling depth-integrated pollution dispersion in the gulf of thermaikos using a lagrangian particle technique. In M. DE Conceicao Cunha, editor, *Proc. of the 3^{rd} of Inter. Confer. on Water Resources Management, Portugal*, pages 173–184, UK, 2005. WIT Trans. on Ecol. and the Env., Vol 80.

[5] T.Bjørk. *Arbitrage Theory in Continuous Time.* Oxford University Press, Great Britain, UK, 1998.

[6] P. M. Burrage and K. Burrage. Variable stepsize implementation for stochastic differential equations. *SIAM J. Appl. Math.*, 24(3):848–864, 2002.

[7] W. M. Charles, E.van den Berg, H. X. Lin, A. W. Heemink, and M. Verlaan. Parallel and distributed simulation of sediment dynamics in shallow water using particle decomposition approach. *Submitted to the Journal of Parallel and Distributed Computing*, 2006.

[8] W. M. Charles, A. W. Heemink, and E. van den Berg. Stochastic particle models for transport problems in coastal waters. In C.A. Brebbia and M. de Conceicao Cunha, editors, *Proc. of the 7^{th} of Inter. Confer. on Modelling, Measurements, Engineering and Management of Seas and Coastal Regions, Algarve,Portugal*, pages 69–80, UK, 2005. WITPress.

[9] W. M. Charles, A. W. Heemink, and M. Verlaan. A particle model for simulation of sediment transport in coastal waters. *Journal of the Nigerian Association of Mathematical Physics*, 8(1116-4336):131–138, 2004.

[10] W. M. Charles, A.W. Heemink, and E. van den Berg. Random flight models for dispersion of pollutants in shallow waters. *Submitted to the J. Appl. Mathematical modelling*, 2005.

[11] W. M. Charles, D. Spivakovskaya, A. W. Heemink, and H. X. Lin. A parallel particle model for simulating sediment transport in coastal waters. In Gamal Attiya and Yskandar Hamam, editors, *Proc. of the 5th EUROSIM Congress on Modelling and Simulation, (II) 200)*, pages 1–7, Marne la Vallée, 2004.

[12] W. M. Charles, E. van den Berg, H.X Lin, and A. W. Heemink. Variable time stepping in parallel particle models for transport problems in shallow waters. In C. A. Brebbia and J.S Antunes do Carmo, editors, *Proc. of the 8th International Conference on Modelling, Monitoring and Management of Water Pollution*, pages 465–475. WIT Transactions on Ecology and the Environment, 2006.

[13] M. Costa and J. S. Ferreira. Discrete particle distribution model for advection-diffusion transport. *Journal of hydraulics engineering*, pages 525–532, 2000.

[14] M. R. F. Costa. *Dispar method and their implementation on a heterogeneous PC cluster.* Ph.D. thesis, Universidade Nova de Lisboa Faculdade, de Cieências Tecnologia,Lisboa, Potugal, 2003.

[15] K. de Jong. *Tidally Averaged Transport Models.* Ph.D. thesis, Delft University of Technology, the Netherlands, 1998.

[16] J. Desmond. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Rev*, 43(3):525–546, 2001.

[17] K. Dimou and E. Adams. A random-walk particle tracking model for well-mixed estuaries and coastal waters. *Estuarine, Coastal and Shelf Science*, 37:99–110, 1993.

[18] J. J. Dronkers. *Tidal Computations in Rivers and Coastal Waters.* North-Holland, New York, 1964.

[19] D. Dunsbergen. *Particle Models for Transport in Three-dimensional Shallow Water Flow.* Ph.D. thesis, Delft University of Technology, the Netherlands, 1994.

[20] N. Fahmi. *A Finite Volume Solver for the Simulation of Transport Processes.* Ph.D. thesis, Delft University of Technology,the Netherlands, 2006.

[21] H. B. Fischer, E.J. List, R.C.Y. Koh, J. Imberger, and N.H. Brooks. *Mixing in Inland and Coastal Waters.* Academic Press, New York, 1979.

[22] I. T. Foster. *Designing and Building Parallel Programs: Concept and Tools for Parallel Software Engineering.* Addison-Wesley Publishing Company, New York, 1995.

[23] J. Fredsøe and R. Deigaard. *Mechanics of Coastal Sediment transport.* World Scientific, Singapore, 1992.

[24] J. G. Gaines and T. J. Lyons. Variable stepsize control in the numerical solutions of sdes. *SIAM J. Appl. Math*, 57(5):1455–1484, 1997.

[25] C. W. Gardiner. *Handbook of Stochastic Methods for physics, Chemistry and the Natural Sciences*. Springer-Verlag, Berlin, 2004.

[26] I. I. Gihman and A.V. Skorohod. *Stochastic Differential Equations.* Springer, Berlin Heidelberg, New York, 1972.

[27] P. Groen. On the residual transport of suspended load matter by an alternative tidal current. *Neth. J.Sea Res.*, 3:564–674, 1977.

[28] A. W. Heemink. Stochastic modeling of dispersion in shallow water. *Stochastic Hydrology and hydraulics*, 4:161–174, 1990.

[29] A. W. Heemink. Tidally averaged models for dispersion in shallow water. *Stochastic Hydrology and hydraulics*, 3:607–617, 1993.

[30] J. Hoffman. *Numerical Methods for engineers and Scientists.* McGraw-Hill International Editions, San Diego and California USA, 1993.

[31] J. R. Hunter, P.D Crais, and H. E. Phillips. On the use of random walk models with spatially variable diffusivity. *Computation Physics*, 106:366–376, 1993.

[32] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

[33] P. E. Kloeden and E. Platen. A survey of numerical methods for stochastic differential equations. *Stochastic Hydrology and hydraulics*, pages 155–178, 1989.

[34] P. E. Kloeden and E. Platen. *Numerical solutions of Stochastic Differential equations. Application of Mathematics,Stochastic Modelling and applied probability.* Springer-Verlag, Berlin Heidelberg, 1999.

[35] P. E. Kloeden, E. Platen, and H. Schurz. *Numerical solutions of SDE Through Computer Experiments*. Springer, New York, 2003.

[36] C. W. Li and T. S. Yu. Conservative characteristics-based schemes for mass transport. *Journal of Hydraulic Engineering*, ASCE,123(4):303–314, 1994.

[37] H. X. Lin, A. W. Heemink, and J. W. Stijnen. Parallel simulation of the transport phenomena with the particle model Simpar. In B. H. Li, editor, *Proc. of the 4^{th} Inter. Confer. on System Simulation and Scientific Computing*, pages 17–22, Beijing, 1999. Academic Press.

[38] H. X. Lin, H. H. ten Cate, L. Dekker, A. W. Heemink, M. R. T Roest, E. A. H Vollebregt, Th.L van Stijn, and J.B.Berlamont. Parallel simulation of 3-d flow and transport models within the nowesp project. *Inter. J. of the Federation of the European Simulation Societies:Simulation Practice and Theory*, 3:257–271, 1995.

[39] H. X. Lin, H.H. ten Cate, M.R.T. Roest, and E.A.H. Vollebregt. *Technical Documentation Online Coupling of TRIWAQ and SIMPAR.* Contract simona sys/parkop01, code rkz-336, Project Report to National Institute for Coastal and Marine Management (RIKZ), 1997.

[40] A. Matheja. Sediment transport using the random walk method on top of cobra. In *Proc. of the 4^{th} of Inter. Confer. on Hydroinformatics, Iowa City,*, pages 1–7, USA, 2000.

[41] G. N. Milstein. *Numerical Integration of Stochastic Differential Equations.* Kluwer Academic Publishers, Dordrecht, 1995.

[42] B. Oksendal. *Stochastic Differential Equations. An introduction with applications.* Springer, New York, 2003.

[43] M. J. Quinn. *Parallel Computing: Theory and Practice.* McGraw-Hill, 1994.

[44] M. R. T. Roest. *Partitioning for Parallel Finite Difference Computations Coastal Water Simulation.* Ph.D. thesis, Delft University of Technology, the Netherlands, 1997.

[45] H. M. Schuttelaars and H. E. de Swart. An idealized long-term morhpodynamic model of a tidal embayment. *Eur. J.Mech., B/Fluids*, 15(1):55–80, 1996.

[46] H. M. Schuttelaars and H. E. de Swart. Multiple morphodynamic equilibria in tidal embayments. *J. Geophysical Research*, 105(c10):14,105–14,118, 2000.

[47] G. Y. El Serafy. *Data assimilation Particle models for Groundwater Contamination.* Ph.D. thesis, Delft University of Technology, the Netherlands, 2000.

[48] C. Soize. *The Fokker-Planck Equation for Stochastic Dynamical Systems and its Explicit Steady State Solutions. Series on Advanced Mathematics for Applied Sciences, Vol.17.* World Scientific Publishing Co.Pte.Ltd by Utopia Press, Singapore, 1994.

[49] G. S. Stelling. *Communications on construction of computational methods for shallow water flow problems.* Ph.D. thesis, Delft University of Technology,the Netherlands, 1983.

[50] J. W. Stijnen. *Numerical Methods for stochastic Environmental Models.* Ph.D. thesis, Delft University of Technology, the Netherlands, 2002.

[51] J. W. Stijnen and A.W. Heemink. Numerical treatment of stochastic river quality models driven by colored noise. *Water resources research*, 39(3):1–9, 2003.

[52] H. M. Taylor and S. Karlin. *An Introduction to Stochastic Modelling.* Academic Press, San Diego and California USA, 1998.

[53] D. J. Thomson. Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *J. Fluid Mech.*, 180:529–556, 1987.

[54] A. F. B. Tompson and L. W. Grelhar. Numerical simulation of the solute transport in three-dimensional, randomly heterogeneous porous media. *Water Resources Research*, 26(10):2541–2562, 1990.

[55] R. A. Trompert and J. G. Verwer. Analysis of the implicit Euler local uniform grid refinement method. *SIAM J. Sci. Comput.*, 14:259–278, 1993.

[56] G. J. M. Uffink. *Analysis of dispersion by the Random walk method.* Ph.D. thesis, Delft University of Technology, the Netherlands, 1990.

[57] H. F. P. van den Boogaard, M. J. J . Hoogkamer, and A. W. Heemink. Parameter identification in particle models. *Stochastic Hydrology and Hydraulics*, 7(2):109–130, 1993.

[58] J. van den Kreeke, S. E. Hoogewoning, and M.Verlaan. Analytical model for the morphodynamics of a trench in the presence of tidal currents. *Continental Shelf Research*, 22(11-12):1811–1820, 2002.

[59] L. C. van Rijn. *Principles of sediment transport in rivers, estuaries and coastal seas.* Aqua Publications, Amsterdam, 1993.

[60] E. A. H. Vollebregt. *Parallel Software Development Techniques for Shallow Water Models.* Ph.D. thesis, Delft University of Technology, the Netherlands, 1997.

[61] J. F. Zimmerman. Topographic generation of residual circulation by oscillatory tidal currents. *Geophys. Astrophys. Fluid Dyn*, 59:35–47, 1978.

Appendix A

As in [3], for example let us consider 1-dimensional white process. White noise has a constant spectral density $f(\lambda)$ on the entire real axis. More detailed information on this concept can be found for example in [42]. If $\mathbb{E}[\xi(s)\xi(t+s)] = C(t)$ is the covariance function of $\xi(t)$, then, the spectral density is given:

$$f(\lambda) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\lambda t} C(t)dt = \frac{c}{2\pi}, \quad \forall \ \lambda \in \ \Re^1.$$

The positive constant c without loss of generality can take a value equals 1. White noise $\xi(t)$ can be approximated by an ordinary stationary Gaussian process $X(t)$, for example one with covariance:

$$\mathrm{C(t) = ae^{-b|t|}, \ (a > 0, b > 0),}$$

it can be shown that such a process has a spectral density.

$$f(\lambda) = \frac{ab}{\pi(b^2+\lambda^2)}.$$

$$\begin{aligned}
f(\lambda) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\lambda t}\mathrm{C(t)dt} \\
&= \frac{a}{2\pi}\int_{-\infty}^{\infty} e^{-b|t|}e^{-i\lambda t}dt \\
&= \frac{a}{2\pi}\int_{-\infty}^{\infty} e^{-b|t|}\left[\cos(\lambda t) - i\sin(\lambda t)\right]dt \\
&= \frac{a}{2\pi}\underbrace{\int_{-\infty}^{\infty} e^{-b|t|}\cos(\lambda t)dt}_{even} - \frac{a}{2\pi}\underbrace{\int_{-\infty}^{\infty} ie^{-b|t|}\sin(\lambda)dt}_{odd} \\
&= \frac{2a}{2\pi}\int_{0}^{\infty} e^{-bt}\cos(\lambda t)dt \\
&= \frac{a}{\pi}\int_{0}^{\infty} e^{-bt}\cos(\lambda t)dt \\
&= \frac{a}{\pi}\left[\frac{-b}{b^2+\lambda^2}e^{-bt}\cos(\lambda t) + \frac{\lambda}{b^2+\lambda^2}e^{-bt}\sin(\lambda t)\right]_0^{\infty} \\
&= \frac{a}{\pi}\left[\frac{-b}{b^2+\lambda^2}e^{-bt}\cos(\lambda t)\right]_0^{\infty} \\
&= \frac{a}{\pi}\left[0 + \frac{b}{b^2+\lambda^2}\right] \\
f(\lambda) &= \frac{ab}{\pi(b^2+\lambda^2)}. \qquad \text{(A.1)}
\end{aligned}$$

If we now let a and b approach ∞ in such a way that $\frac{a}{b} \to \frac{1}{2}$, we get

$$f(\lambda) \to \frac{1}{2\pi} \;\; \forall \;\; \lambda \in \Re', \qquad \mathrm{C(t)} = \begin{cases} 0 & t \neq 0, \\ \infty & t = 0 \end{cases}, \qquad \int_{-\infty}^{\infty} \mathrm{C(t)dt} \to 1,$$

so that $\mathrm{C(t)} \to \delta(\mathrm{t})$, that is, $X(t)$ converges in a certain sense to $\xi(t)$ [3, 32, 35].

Appendix B

Following Karhunen-Loève see [35], we define the random variables a_r^x and b_r^x by

$$\begin{aligned} a_r^x &= \frac{2}{\Delta t}\int_0^{\Delta t}\left(W_s^x - \frac{s}{\Delta t}W_{\Delta t}^x\right)\cos\left(\frac{2r\pi s}{\Delta t}\right)ds && \text{(B.1)}\\ \text{and}\quad b_r^x &= \frac{2}{\Delta t}\int_0^{\Delta t}\left(W_s^x - \frac{s}{\Delta t}W_{\Delta t}^x\right)\sin\left(\frac{2r\pi s}{\Delta t}\right)ds, \quad r = 1, 2, \ldots && \text{(B.2)}\end{aligned}$$

It is known that, for $r \geq 1$ these variables have an $\mathcal{N}\left[0, \frac{\Delta t}{2\pi^2 r^2}\right]$ distribution. They are differentiable samples paths on the interval $[0, T]$. In this section we have derived the variance of the random variables a_r^x, the variances of the remaining random variables can follow similar lines. Recall that

$$\left(\int_0^1 f(x)dx\right)^2 = \int_0^1\int_0^1 f(x)f(y)dxdy \qquad \text{(B.3)}$$

Let us consider the variance of Eqn. B.1, it follows that

$$\mathbb{E}\left\{\left(\frac{2}{\Delta t}\int_0^{\Delta t}\left(W_s - \frac{s}{\Delta t}W_{\Delta t}\right)\cos\left(\frac{2r\pi s}{\Delta t}\right)ds\right)^2\right\} =$$
$$\frac{4}{(\Delta t)^2}\int_0^{\Delta t}\int_0^{\Delta t}\mathbb{E}\left[\left(W_{s_1} - \frac{s_1}{\Delta t}W_{\Delta t}\right)\left(W_{s_2} - \frac{s_2}{\Delta t}W_{\Delta t}\right)\right]\cos\left(\frac{2r\pi s_1}{\Delta t}\right)\cos\left(\frac{2r\pi s_2}{\Delta t}\right)\times$$
$$ds_1 ds_2$$

Now let us first consider the expression in the square brackets for $s_1 < s_2$ Recall that

$$\begin{aligned}\operatorname{Cov}(X, Y) &= \mathbb{E}[XY] - \mathbb{E}[\mathbb{X}]\mathbb{E}[\mathbb{Y}]\\ &= \mathbb{E}[XY]\end{aligned}$$

but $\mathbb{E}[X]\mathbb{E}[Y] = 0$, therefore, in our case.

$$\operatorname{Cov}\left(W_{s_1} - \frac{s_1}{\Delta t}W_{\Delta t}, W_{s_2} - \frac{s_2}{\Delta t}W_{\Delta t}\right) =$$
$$\operatorname{Cov}(W_{s_1}, W_{s_2}) - \operatorname{Cov}\left(W_{s_1}, \frac{s_2}{\Delta t}W_{\Delta t}\right) - \operatorname{Cov}\left(\frac{s_1}{\Delta t}W_{\Delta t}, W_{s_2}\right) + \operatorname{Cov}\left(\frac{s_1}{\Delta t}W_{\Delta t}, \frac{s_2}{\Delta t}W_{\Delta t}\right)$$

$$\begin{aligned}
&= \operatorname{Cov}\left(W_{s_1}, W_{s_2}\right) - \frac{s_2}{\Delta t}\operatorname{Cov}\left(W_{s_1}, W_{\Delta t}\right) - \frac{s_1}{\Delta t}\operatorname{Cov}\left(W_{\Delta t}, W_{s_2}\right) + \frac{s_1}{\Delta t}\frac{s_2}{\Delta t}\operatorname{Cov}\left(W_{\Delta t}, W_{\Delta t}\right) \\
&= s_1 - \frac{s_2 s_1}{\Delta t} - \frac{s_1 s_2}{\Delta t} + \frac{s_1 s_2}{\Delta t} \\
&= s_1 - \frac{s_2 s_1}{\Delta t} \\
&= \min(s_1, s_2) - \frac{s_2 s_1}{\Delta t}.
\end{aligned}$$

Using Eqn. (B.4) we get the following

$$\frac{4}{(\Delta t)^2}\left\{\int_0^{\Delta t}\int_0^{\Delta t}\left[\min(s_1, s_2) - \frac{s_2 s_1}{\Delta t}\right]\cos\left(\frac{2r\pi s_1}{\Delta t}\right)\cos\left(\frac{2r\pi s_2}{\Delta t}\right)\right\} ds_1 ds_2. \tag{B.4}$$

Change the variable of integration such that $x_1 = \frac{s_1}{\Delta t}$ and $x_2 = \frac{s_2}{\Delta t}$ so Eqn. (B.4) becomes

$$4\Delta t\left\{\int_0^1\left[\int_0^1 \left[\min(x_1, x_2) - x_1 x_2\right]\cos\left(2r\pi x_1\right)\cos\left(2r\pi x_2\right)dx_1\right]dx_2\right\}. \tag{B.5}$$

Now if we let $x_1 < x_2$ we get

$$4\Delta t\int_0^1\left[\int_0^{x_2}\underbrace{\left(x_1 - x_1 x_2\right)\cos\left(2r\pi x_1\right)dx_1}_{x_1<x_2} + \int_{x_2}^1\underbrace{\left(x_2 - x_1 x_2\right)\cos\left(2r\pi x_1\right)dx_1}_{x_1>x_2}\right]\times \cos\left(2r\pi x_2\right)dx_2. \tag{B.6}$$

Let us now consider the inner integral:

$$\int_0^{x_2} x_1\cos\left(2r\pi x_1\right)dx_1 - \int_0^{x_2} x_1 x_2\cos\left(2r\pi x_1\right)dx_1 + \int_{x_2}^1 x_2\cos\left(2r\pi x_1\right)dx_1 - \int_{x_2}^1 x_1 x_2\cos\left(2r\pi x_1\right)dx_1. \tag{B.7}$$

Thus Eqn. (B.6) is integrated by using integration by parts of trigonometric function. We start with :

$$\int_0^{x_2} x_1\cos\left(2r\pi x_1\right)dx_1 \tag{B.8}$$

Let us choose $u = x_1$ and $dv = \cos\left(2r\pi x_1\right)dx_1$ Thus $v = \frac{1}{2r\pi}\sin\left(2r\pi x_1\right)$, hence

$$\begin{gathered}
\left[\frac{x_1}{2r\pi}\sin\left(2r\pi x_1\right)\right]_0^{x_2} + \left[\frac{1}{4r^2\pi^2}\cos\left(2r\pi x_1\right)\right]_0^{x_2} = \\
\left[\frac{x_2}{2r\pi}\sin\left(2r\pi x_2\right) + \frac{1}{4r^2\pi^2}\cos\left(2r\pi x_1\right) - \frac{1}{4r^2\pi^2}\right].
\end{gathered} \tag{B.9}$$

Again we consider the third term:

$$\begin{aligned}\int_{x_2}^{1} x_2 \cos(2r\pi x_1)\, dx_1 &= x_2 \int_{x_2}^{1} \cos(2r\pi x_1)\, dx_1 = \left[\frac{x_2}{2r\pi} \sin(2r\pi x_1)\right]_{x_2}^{1} \\ &= \left[\frac{x_2}{2r\pi} \sin(2r\pi) - \frac{x_2}{2r\pi} \sin(2r\pi x_2)\right] \\ &= \left[-\frac{x_2}{2r\pi} \sin(2r\pi x_2)\right]. \qquad \text{(B.10)}\end{aligned}$$

Note that $\sin(2r\pi) = 0, \quad r = 0, 1, 2, \ldots$.

Thus if we add Eqn (B.9) and (B.10) we get

$$\int_{0}^{x_2} x_1 \cos(2r\pi x_1)\, dx_1 + x_2 \int_{x_2}^{1} \cos(2r\pi x_1)\, dx_1 = \left[\frac{1}{4r^2\pi^2} \cos(2r\pi x_1) - \frac{1}{4r^2\pi^2}\right]. \text{(B.11)}$$

Next we consider the second term:

$$\begin{aligned}x_2 \int_{0}^{x_2} x_1 \cos(2r\pi x_1)\, dx_1 &= x_2 \left\{ \left[\frac{x_1}{2r\pi} \sin(2r\pi x_1)\right]_{0}^{x_2} + \left[\frac{1}{4r^2\pi^2} \cos(2r\pi x_1)\right]_{0}^{x_2} \right\} \\ &= \left[\frac{x_2 x_2}{2r\pi} \sin(2r\pi x_2) + \frac{x_2}{4r^2\pi^2} \cos(2r\pi x_2) - \frac{x_2}{4r^2\pi^2}\right] \text{(B.12)}\end{aligned}$$

Thus

$$\begin{aligned}&\int_{0}^{x_2} x_1 \cos(2r\pi x_1)\, dx_1 + x_2 \int_{x_2}^{1} \cos(2r\pi x_1)\, dx_1 - x_2 \int_{0}^{x_2} x_1 \cos(2r\pi x_1)\, dx_1 = \\ &\left[\frac{1}{4r^2\pi^2} \cos(2r\pi x_1) - \frac{1}{4r^2\pi^2} - \frac{x_2 x_2}{2r\pi} \sin(2r\pi x_2) - \frac{x_2}{4r^2\pi^2} \cos(2r\pi x_2) + \frac{x_2}{4r^2\pi^2}\right] \quad \text{(B.13)}\end{aligned}$$

Finally we consider the fourth integral:

$$x_2 \int_{x_2}^{1} x_1 \cos(2r\pi x_1)\, dx_1$$

$$\begin{aligned}&x_2 \int_{x_2}^{1} x_1 \cos(2r\pi x_1)\, dx_1 = x_2 \left\{ \left[\frac{x_1}{2r\pi} \sin(2r\pi x_1)\right]_{x_2}^{1} + \left[\frac{1}{4r^2\pi^2} \cos(2r\pi x_1)\right]_{x_2}^{1} \right\} \\ &= \frac{x_2}{2r\pi} \sin(2r\pi) - \frac{x_2 x_2}{2r\pi} \sin(2r\pi x_2) + \frac{x_2}{4r^2\pi^2} \cos(2r\pi) - \frac{x_2}{4r^2\pi^2} \cos(2r\pi x_2) \quad \text{(B.14)}\end{aligned}$$

Finally we take the results for all integrals in the square bracket of Eqn. (B.5):

$$\begin{aligned}&\int_{0}^{x_2} x_1 \cos(2r\pi x_1)\, dx_1 - x_2 \int_{0}^{x_2} x_1 \cos(2r\pi x_1)\, dx_1 + x_2 \int_{x_2}^{1} \cos(2r\pi x_1)\, dx_1 \\ &\qquad + x_2 \int_{x_2}^{1} x_1 \cos(2r\pi x_1)\, dx_1 \quad \text{(B.15)}\end{aligned}$$

$$\begin{aligned}&\frac{1}{4r^2\pi^2}\cos(2r\pi x_2)-\frac{1}{4r^2\pi^2}-\frac{x_2x_2}{2r\pi}\sin(2r\pi x_2)-\frac{x_2}{4r^2\pi^2}\cos(2r\pi x_2)+\frac{x_2}{4r^2\pi^2}\\&\quad-\frac{x_2}{2r\pi}\sin(2r\pi)+\frac{x_2x_2}{2r\pi}\sin(2r\pi x_2)-\frac{x_2}{4r^2\pi^2}\cos(2r\pi)+\frac{x_2}{4r^2\pi^2}\cos(2r\pi x_2)\end{aligned}\tag{B.16}$$

$\cos(2r\pi)=1, \quad r=0,1,2\ldots.$ Finally equation (B.16) becomes

$$\frac{1}{4r^2\pi^2}\cos(2r\pi x_2)-\frac{1}{4r^2\pi^2}.\tag{B.17}$$

Next we put Eqn (B.17) into the integral (B.6) and obtain

$$4\Delta t\int_0^1\left[\frac{1}{4r^2\pi^2}\cos(2r\pi x_2)-\frac{1}{4r^2\pi^2}\right]\cos(2r\pi x_2)\,dx_2\tag{B.18}$$

$$\frac{\Delta t}{r^2\pi^2}\int_0^1\cos^2(2r\pi x_2)\,dx_2-\frac{\Delta t}{r^2\pi^2}\int_0^1\cos(2r\pi x_2)\,dx_2\tag{B.19}$$

we now use double/half angle formula.

where

$$\cos^2(2r\pi x_2)=\frac{1}{2}\left[\cos(4r\pi x_2)+1\right]$$

Thus we have

$$\frac{\Delta t}{r^2\pi^2}\int_0^1\frac{1}{2}\left(\cos(4r\pi x_2)+1\right)dx_2-\frac{\Delta t}{r^2\pi^2}\int_0^1\cos(2r\pi x_2)\,dx_2\tag{B.20}$$

$$\frac{\Delta t}{2r^2\pi^2}\int_0^1\cos(4r\pi x_2)\,dx_2+\frac{\Delta t}{2r^2\pi^2}\int_0^1 dx_2+\frac{\Delta t}{r^2\pi^2}\int_0^1\cos(2r\pi x_2)\,dx_2\tag{B.21}$$

$$\frac{\Delta t}{2r^2\pi^2}\int_0^1 dx_2+\frac{\Delta t}{2r^2\pi^2}\int_0^1\cos(4r\pi x_2)\,dx_2+\frac{\Delta t}{r^2\pi^2}\int_0^1\cos(2r\pi x_2)\,dx_2\tag{B.22}$$

$$=\frac{\Delta t}{2r^2\pi^2}+0+0.\tag{B.23}$$

Therefore

$$\operatorname{Var}[a_r]=\left(\frac{2}{\Delta t}\right)^2\mathbb{E}\left\{\left(\int_0^{\Delta t}\left(W_s-\frac{s}{\Delta t}W_{\Delta t}\right)\cos\left(\frac{2r\pi s}{\Delta t}\right)ds\right)^2\right\}=\frac{\Delta t}{2r^2\pi^2}\quad\diamond\tag{B.24}$$

B.1 Approximation of the multiple stratonovich integral $J_{(1,1,0)}$

$J^{x,p}_{(1,1,0)}$ is an approximation of the integral $J^{x}_{(1,1,0)}$ and it is known [35] that $J^{x}_{(1,1,0)} \geq \frac{(\Delta M^x)^2}{2\Delta t_n}$ always, where p is an integration index. If it turns out $J^{x,p}_{(1,1,0)} < \frac{(\Delta M^x)^2}{2\Delta t_n}$, we take ΔM^x as the better approximation for $J_{(1,1,0)}$. Therefore, the derivation can be done as follows: By the Cauchy Schwartz inequality.

$$|\langle x, y\rangle|^2 \leq \left(\sum_i x_i\right)\left(\sum_i y_i\right)$$

$$\int_0^{\Delta t} W_s \cdot 1ds \leq \int_0^{\Delta t} W_s^2 ds \int_0^{\Delta t} 1ds.$$

This can be proved by considering the follows where c is a constant:

$$\begin{aligned}\int_0^{\Delta t} (W_s + c\cdot 1)^2\, ds &= c^2\int_0^{\Delta t} 1^2 ds + 2c\int_0^{\Delta t} W\cdot 1ds + \int_0^{\Delta t} W_s^2 ds \geq 0\\ &= 4\left(\int_0^{\Delta t} W_s \cdot 1ds\right)^2 - 4\int_0^{\Delta t} 1^2 ds \int_0^{\Delta t} W_s^2 ds \leq 0\end{aligned}$$

From the Kloeden book [34] we notice that

$$2J_{(1,1,0)}\Delta t = \int_0^{\Delta t} W_s^2 ds$$

where

$$\Delta Z = \Delta M = \int_0^{\Delta t} W_s ds \sim \mathrm{N}\left[0, \frac{1}{2}\Delta \mathrm{t}\right].$$

In the implementation we have simply taken

$$\max\left(2J^p_{(1,1,0)}\Delta t, (\Delta Z)^2\right).$$

Since $J^p_{(1,1,0)}$ is an approximation for $J_{(1,1,0)}$, theoretically by the Cauchy Schwartz inequality we know that

$$J_{1,1,0)} \geq \frac{(\Delta Z)^2}{2\Delta t} \quad \textit{always}$$

if it turns out that

$$J^p_{1,1,0)} < \frac{(\Delta Z)^2}{2\Delta t},$$

then we take ΔZ as the better approximation for $J_{(1,1,0)}$.

Summary

This thesis presents the results of a study of the application of two dimensional particle models for simulating the pollutants and sediment transport in shallow waters. In order to achieve an improved understanding of complex environmental systems, a high performance computing model is vital for the Lagrangian particle technique. The research in this thesis was first to improve the existing particle model to describe correctly the short term spreading behaviour of a cloud of pollutants in shallow water. Secondly, to develop a high performance particle model for sediment transport. The transported materials in shallow waters are assumed to be dissolved but inactive suspended materials.

The movement of a particle comprises a drift (advective) component and a diffusive(dispersive) component. In numerical modelling the diffusive component of the displacement of a particle at each time step is often implemented by adding random driving noise with an appropriate standard deviation. The driving force in the particle model is often known as Wiener process. It is known that particle methods describe the dispersion of cloud of particles in turbulent fluid flow accurately if that cloud has been in the flow much longer than a certain Lagrangian time (T_L). The Lagrangian time scale is a measure of how long the particle takes to lose memory of its initial turbulent velocity. Existing particle models are generally unable to accurately model the dispersion process of a cloud of pollutants from time 0 to T_L. In order to solve this problem, we have developed a particle model called the random flight model which uses a correlated process known as coloured noise as a driving force. In this model the increments of the coloured noise are correlated in time for $t \ll T_L$. Thus, in the beginning the movements (increments) of a particle depends not only on the flow velocity but also the velocity of the particle. In this way we have been able to take into account the correlation behaviour of dispersion of a cloud of pollutants shortly after its deployment for $t \ll T_L$.

The ultimate goal of this thesis was to develop an efficient particle model for sediment transport in shallow water. To this end we have modified the existing particle model which was developed for the prediction of pollutants dispersion in coastal water. We have introduced in the particle model extra equations for the erosion/suspension using a probabilistic concepts (the Poisson distribution function) to determine the actual number of particles which are suspended in each grid cell. The settling/deposition is modelled

by using an exponential decaying ordinary differential equation.

Furthermore, we have also derived the two dimensional modified Fokker-Planck equation in relation to the developed particle model for sediment transport. This has been done by adding the two extra terms for erosion and deposition. The Fokker-Planck equation describes the evolution of the probability density of particles with time and space. The probability that a particle ends up in a certain location is of course a description of the concentration in that location.

Particle models can be used to calculate/predict the concentration of pollutants. However, for a long simulation period in which case the particles are spread over a large area, many particles are needed to obtain the accurate results. Consequently, the problems of high computational costs plus memory problem can go beyond the capacity of a singel sequential computer. We have addressed this problem by developing a parallel particle model which can be used on a parallel computer and/or a cluster of PCs.

In order to improve the efficiency and computing speed, we have considered, the use of variable integration time step sizes instead of fixed time step. The use of fixed time step sizes in the numerical methods in particle models have limitations when, for example, the SDEs being solved are stiff as this forces the step size to be very small. Fixed small step sizes in the numerical approximation of SDEs may introduce unnecessary huge computational cost. An adaptive scheme has been developed in our particle model with variable time step sizes. In this advanced scheme during the simulation an optimal step size for a given error tolerance is estimated. The adaptive scheme has been applied in the Dutch coastal water to simulate the transport of pollutants. The results show that the smaller step sizes are needed to stably integrate the movement of a particle in highly irregular regions. The adaptive scheme is also fully incorporated into the parallel implementation of the particel model.

Samenvatting

In dit proefschrift worden de resultaten van het onderzoek naar deeltjesmodellen voor de simulatie van het transport van verontreinigingen en sediment in ondiepe wateren beschreven. Om grootschalige complexe milieu systemen beter te kunnen begrijpen is het toepassen van high performance computing technieken bij de simulatie van Lagrangiaanse modellen van groot belang. Het onderzoek richt zich in de eerste instantie tot het verbeteren van de bestaande deeltjesmodellen om het korte termijngedrag van de verspreiding van een wolk van verontreinigingen in ondiep water nauwkeuriger te beschrijven.Vervolgens wordt een parallel deeltjesmodel voor sediment transport ontwikkeld.

De verplaatsing van een deeltje kan worden verdeeld in twee componenten: een advective component en een diffusive component. Bij het numeriek modelleren van de verplaatsing van een deeltje wordt de diffusive component vaak geimplementeerd door een random getal met een overeenkomstige standaard afwijking. Dit random proces in het deeltjesmodel wordt het Wiener proces genoemd. Het is bekend dat deeltjesmodellen de dispersie van een deeltjeswolk in turbulente stroming nauwkeurig beschrijven als de deeltjeswolk al langer dan een bepaald Lagrangiaanse tijd (T_L) in het water is. De Lagrangiaanse tijdschaal is een maat voor de tijd die deeltjes nodig hebben om hun geheugen' van de initiële turbulente snelheid te verliezen.

Bestaande deeltjesmodellen kunnen in het algemeen het dispersie proces van een wolk van verontreinigingen van tijd 0 tot T_L niet nauwkeurig beschrijven. Om dit probleem op te lossen hebben we een zogenaamde random flight model ontwikkeld dat de beweging van de deeltjes als een gecorreleerd proces (de gekleurde ruis) modelleerd. In dit model zijn de random veranderingen van de gekleurde ruis voor $t \ll T_L$ gecorreleerd. Dus de verplaatsing van een deeltje hangt niet alleen van de stromingsnelheid af, maar ook van de snelheid van het deeltje. Op deze manier kan het bekende fysische verschijnsel van de dispersie van een deeltjeswolk in de begintijd beter worden gemodelleerd.

Het uiteindelijke doel van dit promotieonderzoek is het ontwikkelen van een efficiënt deeltjesmodel voor sediment transport in ondiep water. Om dit te bereiken hebben we het bestaande deeltjesmodel voor de voorspelling van de dispersie van verontreinigingen in kustwateren aangepast. We hebben daarvoor de vergelijkingen voor de beschrijving van erosie/suspensie aan het bestaande deeltjesmodel toegevoegd. Het actuele aantal zwevende' deeltjes in iedere grid cel wordt door middel van een Poisson distributiefunctie bepaald. Het proces van sedimentatie wordt gemodelleerd met een differentiaalvergelijking die een proces met exponentieel uitdemping beschrijft.

We hebben ook de twee dimensionale aangepaste Fokker-Planck vergelijkingen met betrekking tot het deeltjesmodel voor sediment transport afgeleid. Dit wordt gedaan met de toevoeging van de twee termen: erosie en sedimentatie. De Fokker-Planck vergelijking beschrijft de evolutie van de kansdichtheid van deeltjes als functie van tijd en ruimte. De kans dat een deeltje zich in een bepaalde locatie bevindt, kan naar een concentratiewaarde in die locatie worden vertaald.

Deeltjesmodellen kunnen worden gebruikt voor het berekenen/voorspellen van de concentratie van bepaalde verontreinigingen. In een simulatie voor een lange periode waar de deeltjes over een groot oppervlak zijn verspreid, moeten er heel veel deeltjes in het model worden doorgerekend om een nauwkeurig resultaat te krijgen. Zo'n situatie vraagt heel veel rekentijd en geheugen. We hebben daarom ook een parallel deeltjesmodel voor parallelle computers of cluster van PCs ontwikkeld om dit probleem op te lossen.

Om de efficiëncy en rekensnelheid te verhogen is ook het gebruik van variabele integratie tijdstappen in plaats van een constante tijdstap bestudeerd. Het gebruik van een constante tijdstap in numerieke methoden bij deeltjesmodellen heeft beperkingen. Bijvoorbeeld, wanneer de SDEs stijf zijn, moet een zeer kleine integratie tijdstap worden gebruikt. Het gebruik van een constante kleine tijdstap in de numerieke integratie van SDEs leidt tot onnodig grote rekenkosten. We hebben daarom een adaptief schema met variabele tijdstappen ontwikkeld voor het deeltjesmodel. In dit geavanceerde schema wordt tijdens de simulatie eerst een optimale tijdstap geschat bij een gegeven fout tolerantie. Het adaptieve schema is toegepast op de simulaties van het transport van verontreinigingen langs de Nederlandse kust. De resultaten laten zien dat kleine tijdstappen nodig zijn voor een stabiele integratie bij de berekening van de verplaatsing van de deeltjes in onregelmatige gebieden. Het adaptieve schema is ook volledig ingevoerd in de parallelle implementatie van het deeltjesmodel.

Acknowledgements

I would like to thank the UDSM for the study leave and financial supports which were given to me. Many thanks to Prof V. G. Masanja and Dr Alphonce for encouraging me join the Ph.D research under MHO project. I would also like to thank Mr. Elco van Noort, Christel Crone and CICAT staff for all administrative issues. I'm highly indebted to my promotor Prof A.W. Heemink for his great guidance, insight, and many useful ideas since I came to Delft. His useful discussions, corrections, encouragement and careful reading of this thesis during my research, he supported me with great enthusiasm. Working with him has always been a wonderful experience. Many thanks also to my supervisor Dr. H.X. Lin for introducing me to the subject of parallel processing and useful discussions and guidance. Many thanks to Dr. Hans van Weide for the many useful and helpful discussions. I thank the late Dr. G. C. Herman for the role he played in the initiation of MHO project. At Rijkswaterstaat especially from Jan du Pree, I gratefully enjoyed much support for the Wadden sea data for the implementation of the particle model in this thesis. I thank Dr. M. Verlaan for sharing his practical knowledge as well as useful discussions. Many thanks to Kees Lemmes and Eef Hartman for introducing me to the world of Linux and many other useful computer skills. I would also like to express my very special gratitude to Ewout van den Berg for the very useful discussions and suggestions about the implementation of the parallel particle model in this thesis. I wish to thank Dr E.S. Massawe for his support and encouragement. I would like to thank all people in the Delft Institute of Applied Mathematics(DIAM) in general and Fahmi as well as Daria in particular for the useful discussions. I would like to thank my office mates Svetlan Ponomareva and Xin. I sincerely would like to thank the international student Chaplaincy under Fr. Ben and Rev. Waltraut Stroh for their moral support. Many thanks to my friends, my brothers and my beloved wife during my absence for this research, she supported me in many unfathomed ways. I would also like to thank the following friends, Ineke, Elly van Wille, Paul .G. Cool and Hurry for supporting the water project in my village. Finally I would like to thank my parents notably my late father who constantly encouraged me and was anxious for my success.

Curriculum Vitae

Wilson Mahera Charles was born on October 20, 1969 in Musoma Rural district in Mara region of Tanzania. He studied at Musoma secondary school from 1986-1989. He later studied at Usagara High school, where in 1992 he obtained the advanced certificate in advanced physics, chemistry and mathematics. Between June 1992 and may 1993 he joined the National service at Buhemba Camp under the so called operation Multi-partyism and obtained a certificate of attendance. From September 1993 to September 1995 he worked as accounts assistant for Mara Cooperative Union(MCU). In October 1995, he joined the University of Dar-es-salaam for a bachelor of science with education and graduated in 1999 with BSc(educ.)Hons. In August 1999 he got a NUFU(Norwegian University Funding) scholarship for a Southern African Postgraduate Training Programme in Mathematical Modelling for Southern African Development Community (SADCC). This was based in the Department of Mathematics at the University of Zimbabwe, in August 2001 he obtained MSc.(Mathematics). Upon his arrival back to Tanzania, in October 2001 he was employed by the University of Dar-es-salaam(UDSM) as an assistant Lecturer in the Department of Mathematics (DoM). Under the auspicious of the MHO project which was jointly ran between the University of Dar-es-salaam and Delft University of Technology, in April 2002 he started a Ph.D research programme at the Delft University of Technology in the Delft Institute of Applied Mathematics(DIAM). His work focused on the stochastic modelling of transport problems in shallow waters largely by using stochastic differential equations and parallel processing of the particle transport model. He worked under the supervision of Dr.ir. H. X. Lin and Prof. A. W. Heemink as his promoter. The research was jointly financed by the Dutch Government(Nuffic,TUDelft) and the Tanzania Government(the University of Dar-es-salaam).