TR diss
2469

Particle models for transport in three-dimensional shallow water flow

Deeltjesmodellen voor transport in
driedimensionaal ondiep water stroming

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College van Dekanen aangewezen,
op dinsdag 22 november 1994 te 13.30 uur

door

Daniël Willem DUNSBERGEN
geboren te Waalwijk
doctorandus in de wiskunde

1994

Dit proefschrift is goedgekeurd door de promotor:

Prof.dr.ir. G.S. Stelling

Cover designer: Ted van Slingerland

Published and distributed by:

Delft University Press
Stevinweg 1
2628 CN Delft
The Netherlands

Telephone +31 15 783254
Fax +31 15 781661

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Dunsbergen, D.W.

Particle models for transport in three-dimensional shallow water flow /
D.W. Dunsbergen. - Delft : Delft University Press. - Ill.
Thesis Delft University of Technology. - With ref. - With summary in Dutch.
ISBN 90-407-1054-6
NUGI 841
Subject headings: particle model; transport simulation; three-dimensional shallow water flow

Copyright © 1994 by D.W. Dunsbergen

All rights reserved.
No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without permission from the publisher: Delft University Press, Stevinweg 1, 2628 CN Delft, The Netherlands.

Printed in the Netherlands

This thesis is also published in the series 'Communications on Hydraulic and Geotechnical Engineering' of the Faculty of Civil Engineering as Report No. 94-8, ISSN 01690-6548.

Abstract

The present study focuses on three-dimensional numerical modelling of transport processes in shallow water. The transport process is assumed to be governed by the advection-diffusion equation. Flow data obtained by running a hydrodynamic model are used as input for the transport model. For accurate modelling of complex geometries with free water surface coordinate transformations are applied in the hydrodynamic model as well as in the transport model. Before examining transport simulation the hydrodynamic model is described. Two numerical transport simulation methods are studied. One method makes use of finite difference approximations, the other simulates the transport process with a particle model. The particle model has been described in detail. It deals with particle displacements in curved grids. Discrete flow data obtained by the hydrodynamic model are used consistently. The treatment of boundary and initial conditions are discussed. Accurate algorithms are presented for the computation of the advective and diffusive displacement. Three algorithms are given to convert particle information into, for instance, a concentration distribution.

It turned out that especially in point source applications the particle model is attractive in regions where the concentration distribution contains steep gradients. These steep gradients are accurately resolved. A finite difference method is favourable in cases where the concentration distribution remains smooth. This observation resulted in the development of a combined finite difference / particle method. Such a model yields accurate answers in an efficient way.

Applications focus on reservoir calculations, instantaneous and continuous releases, heat transport in a square model harbour and salt intrusion in the Mekong estuary.

Samenvatting

In deze studie staat de driedimensionale numerieke modellering van transportprocessen in ondiep water centraal. Er is verondersteld dat het transportproces beschreven kan worden met de advectie-diffusie vergelijking. Het transportmodel maakt gebruik van stromingsgegevens die afkomstig zijn van een hydrodynamisch model. Voor een nauwkeurige modellering van gecompliceerde geometrieën met een vrij wateroppervlak is gebruik gemaakt van coördinatentransformaties. Zowel het hydrodynamische model als het transportmodel zijn aan deze transformaties onderworpen. Alvorens in te gaan op transportsimulatie in ondiep water wordt het gebruikte hydrodynamische model beschreven. Twee numerieke transportsimulatiemethoden zijn bestudeerd. De ene maakt gebruik van eindige-differentie-schema's, terwijl de andere het transportproces simuleert met behulp van een deeltjesmodel. Het deeltjesmodel is in detail beschreven. Het behandelt deeltjesverplaatsingen in kromlijnige roosters. Daarbij is op consistente wijze gebruik gemaakt van discrete stromingsgegevens afkomstig uit het hydrodynamische model. De randvoorwaarden- en beginvoorwaardenafhandeling is uiteengezet en nauwkeurige algorithmes voor advectieve en diffusieve verplaatsingen zijn beschreven. Verder zijn drie algorithmes behandeld om gesimuleerde deeltjesposities om te zetten in bijvoorbeeld een concentratieverdeling.

Het is gebleken dat met name in de simulatie van lozingen, het deeltjesmodel aantrekkelijk is in gebieden waar de concentratiegradiënt groot is. Optredende steile concentratiegradiënten worden nauwkeurig opgelost. In gebieden waar de concentratieverdeling glad is geniet het eindige-differentiemodel de voorkeur. Deze observatie heeft geleid tot de ontwikkeling van een gecombineerd eindige-differentie/deeltjesmodel. Een dergelijk model levert op efficiënte wijze nauwkeurige resultaten.

Toepassingen concentreren zich op reservoirberekeningen, instantane en continue lozingen, warmtetransport in een vierkante modelhaven en zoutindringing in het Mekong estuarium.

Contents

Abstract iii
Samenvatting iv
Contents v
Nomenclature vii

1 Introduction 1

2 Flow model 5
2.1 Introduction 6
2.2 Three-dimensional shallow water equations 6
2.3 Turbulence modelling 10
2.4 Shallow water equations in transformed coordinates 12
2.5 Description of a three-dimensional shallow water solver 20
2.6 Concluding remarks 25

3 Three-dimensional transport modelling 27
3.1 Introduction 28
3.2 The advection-diffusion equation 28
3.3 Discretization scheme for the advection-diffusion equation 31

4 Particle methods 37
4.1 Introduction 38
4.2 Theoretical aspects 38
  4.2.1 Stochastic modelling; particle models 39
  4.2.2 Stochastic modelling; boundary conditions 44
  4.2.3 The random walk in transformed coordinates 49
4.3 Numerical aspects of particle models 55
  4.3.1 Derivation of the concentration distribution from a discrete mass density 56
  4.3.2 Computation of particle displacements 64
  4.3.3 Representation of the initial condition 77
  4.3.4 The advection algorithm 81

4.3.5 Implementation of exponential decay and source terms 92
4.3.6 Implementation of boundary conditions 93
4.4 Numerical experiments 101
4.4.1 The advection step 101
4.4.2 The diffusive step 111
4.4.3 The estimated mass flux procedure at boundaries 121
4.4.4 Physical experiments 128

5 The combined Particle/Euler transport model 159
5.1 Introduction 160
5.2 Particle/Euler description 161
5.3 Particle/Euler simulations 164

6 Concluding remarks and discussion 177

References 183

Appendices 193
Appendix A: Transformation relations (I) 194
Appendix B: Transformation relations (II); a curvilinear transformation in the horizontal plane and the sigma-transformation in the vertical plane 196
Appendix C: Transport equation in transformed coordinates 206
Appendix D: Advection schemes 209
Appendix E: Reflection principles at closed boundaries 224

Dankwoord 229
Curriculum Vitae 230

Nomenclature

This section summarizes the most common symbols used in this thesis. Certain symbols have more than one meaning. Explanation is then ruled by the local context.

Roman letters

A	cross-sectional area
$\underline{A}$	drift vector
B	noise tensor
C	Chezy coefficient; concentration
C_E	dimensionless parameter measuring the influence of curvature
C_e	equilibrium bed concentration
C_I	initial concentration
C_D, c_v, c_v'	model parameters (turbulence models)
d	bottom profile
dW	Wiener increment
$\mathrm{d}(.,.)$	distance between both arguments
D	diffusion tensor
D	spatial difference operator
D_H	horizontal diffusion coefficient
D_{ij}	eddy diffusivity in x_i - direction due to the component of concentration gradient in x_j - direction
D_V	vertical diffusion coefficient
D_ξ	diffusive term in momentum equation in ξ - direction
D_η	diffusive term in momentum equation in η - direction
$\underline{e}_i$	unit vector in Cartesian space (x_1,x_2,x_3)
$E\,[.]$	mathematical expectation
$E^{x,t}\,[.]$	expectation with respect to process X started at x at time t
f	Coriolis parameter; frequency
$F_{\partial\Omega}$	mass flux through boundary $\partial\Omega$
g	acceleration due to gravity; noise tensor
$\sqrt{G_{\xi\xi}}$	metric relation (related to Δx)

$\sqrt{G_{\eta\eta}}$	metric relation (related to Δy)
$G_{\xi\eta}$	off-diagonal element of two-dimensional metric tensor
G	entropy
$\sqrt{G}$	Jacobian of horizontal transformation (2DH)
$\underline{h}$	drift vector
H	water depth
i	$\sqrt{-1}$
I_j	grid cell with index j
$\mathrm{Im}(.)$	imaginary part
J	Jacobian
k	turbulent kinetic energy
K	number of layers
l_m	mixing length
L	characteristic length scale
m	modulus of random number generator
M	number of intervals in x - direction; mass
$\underline{n}$	normal vector to boundary
n_p	number of released particles
N	number of time increments
N^c	number of grid cells
$O(.)$	order of magnitude
p	probability density function
$p(.\,\|\,.)$	conditional probability density function
P	probability; hydraulic pressure
P_a	atmospheric pressure
q	mass flux
Q	discharge; volume flux; sink term
Q	maximum number of iterations
r	boundary point; smoothness coefficient
R	uniform random number in $(-1,1)$

Re	Reynolds number
Re(.)	real part
s	time transformation
s_i	i^{th} moment
S	source term; step function
t	time
T	simulation time; flux
T	temperature
T_L	Lagrangian time scale
$\underline{u} = (u_1,u_2,u_3)$	velocity vector in Cartesian space
u	velocity component in x - direction
u_ξ	velocity component in $\underline{a}^1$ - direction
u_η	velocity component in $\underline{a}^2$ - direction
U	contravariant component; characteristic horizontal velocity

$\underline{U} = (U^1,U^2,U^3)$	contravariant velocity vector
v	velocity component in y - direction
V	contravariant component; velocity of substance-water mixture
V_F	frozen advection constant
V_j	volume of cell with index j
w	velocity component in z - direction
w_a	artificial vertical velocity relative to sigma-plane
w_s	settling velocity
W	contravariant component; characteristic vertical velocity
$W(t)$	Wiener process
$\underline{x} = (x_1,x_2,x_3)$	Cartesian coordinate system
(x,y,z)	Cartesian coordinate system
X^p	particle position

Greek letters

γ	decay coefficient
Γ_{ij}^k	Christoffel symbol of the second kind
Γ	stochastic Langevin force
Δ	increment; Laplace operator
Δt	time increment
Δx	spatial increment in x - direction
Δy	spatial increment in y - direction
$\Delta \xi$	increment in ξ - direction
$\Delta \eta$	increment in η - direction
$\Delta \sigma_k$	relative layer thickness of layer k
ϵ	dissipation of turbulent kinetic energy; error measure; reflection coefficient
ζ	water level elevation
θ	integration constant ($0 \leq \theta \leq 1$)
κ	von Karman constant (≈ 0.4)
λ	Lagrange multiplier
μ_p	mass of particle with index p
μ_{tot}	total mass of computational particles
ν_m	kinematic viscosity
ν_t	turbulent eddy viscosity
$\underline{\xi} = (\xi^1,\xi^2,\xi^3)$	curvilinear; general coordinate system

(ξ,η,σ)	curvilinear; general coordinate system
π	3.1415926536
ρ	water density; propagation factor
ρ_{air}	air density
σ	wave speed; standard deviation
σ_t	turbulent Prandtl number
τ	shear stress; time increment used in advection algorithm; exit time; $\frac{1}{2}\Delta t$
τ_{bottom}	bottom friction
τ_c	constituent return time
ϕ	latitude; indicator function
φ	angle; arbitrary function; point spread function
χ	indicator function
Ψ	stream function
ω	vertical velocity component relative to σ - plane
$\underline{\Omega}$	earth's rotation vector
Ω	angular speed of earth; physical domain
∇	nabla operator
$\partial\Omega$	boundary of physical domain

Acronyms

AD	advection-diffusion operator	MED	maximum entropy distribution
A.D.I.	alternating direction implicit	NDD	numerically defined distribution
Det{.}	determinant	pdf	probability density function
diag(.)	matrix with diagonal structure	PSF	point spread function
DP	diffusive part	2DH	two-dimensional horizontal
GPS	grid-point speed	2DV	two-dimensional vertical

Subscripts

c	corrected	t	turbulent quantity
C	curvature	U	local flow velocity
D	Dirichlet boundary: b_D; diffusive	ξ	in ξ - direction
i	covariant vector: $\underline{a}_i$;	η	in η - direction
	i th component of vector: $(\underline{a})_i$	$(\cdot)_{\underline{\xi}}$	quantity at a fixed point in $\underline{\xi}$ - space
m,n,k	grid function defined at centre of cell (m,n,k)		

Superscripts

H	horizontal	$[q]$	iteration index q
i	contravariant vector: $\underline{a}^i$	T	transpose of vector: $\underline{a}^T$
l	time level	V	vertical
n, (n)	time level	0	initial value
p	index of particle	$*$	intermediate model result at $t = (l+½)\Delta t$
$pred$	predictor		

Overlines, primes, etcetera

overline	ensemble averaged; cell-averaged; depth-averaged; refers to numerical method	$\vec{AB}$	vector AB
		$(\cdot)$ and $\cdot$	scalar product
		$\times$	vector product
prime	turbulent fluctuation	$\Vert . \Vert$	Euclidean norm of vector
tilde	transformed quantity: $\tilde{\varphi}(\xi) = \varphi(x(\xi))$	$\vert_x$	at position x
		$[\cdot]$	truncated integer
underline	vector		

Chapter 1

Introduction

1 Introduction

Nowadays simulation techniques play an important role in the assessment of control strategies in natural water bodies such as rivers, lakes, estuaries and coastal seas. Simulation in this sense generally consists of two parts:

(i) simulation of flow, and

(ii) simulation of transport of dissolved or suspended matter.

This simulation procedure is schematized in figure (1.1).

Flow simulation in essence is based upon mathematical elaboration of conservation laws of mass and momentum. In water flow simulation the hydrostatic pressure assumption is taken into account.

Transport simulation is based upon conservation laws or balance principles, taking into account advective and diffusive transport as well as creation, destruction and internal generation of transported substance. Releases (discharges) are treated as input due to point sources.

In shallow water flows the transport of various physical quantities are distinguished, such as salinity, heat, silt, oxygen, bacteria, etcetera. Different transport processes give rise to different numerical simulation techniques. Two different viewpoints can be distinguished, that is the Lagrangian and the Eulerian viewpoint. The Lagrangian method of analysis describes the behaviour of discrete particles, or point masses, as they move in space. The Eulerian method describes what happens at a fixed point or in a fixed region in space. It allows the observation of phenomena at points of interest rather than trying to follow the particle throughout a region in space. The adoption of a transport module in one viewpoint or the other should yield accurate answers to physical problems.

Development of an Eulerian or Lagrangian transport model is motivated by the type of source that induces the transport process. The nature of the source can be either continuous or instantaneous in space or time. Sources continuous in space and time are found in for instance salt intrusion applications in estuaries. This type of transport is adequately simulated in an Eulerian frame of reference. With respect to point sources that are instantaneous in space and time, as can be recognized in for instance dissolved matter transport due to calamities such as ship accidents, the concentration distribution is characterized by steep gradients just after the release. The use of an Eulerian simulation method is often inaccurate. Simulation within a Lagrangian frame of reference is usually an adequate alternative. However, for a big dispersing cloud many Lagrangian particles need to be evaluated. Also

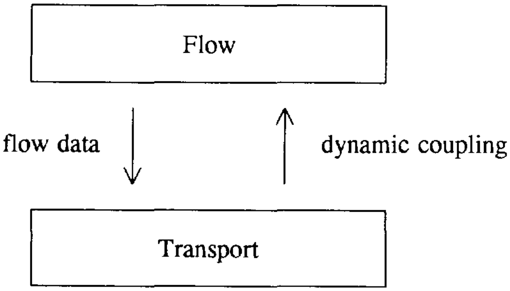


Figure (1.1): Simulation procedure of flow and transport.

in point sources that are continuous in time (continuous discharge of industrial waste) the Lagrangian approach is justifiable near the discharge location (steep gradients are accurately resolved), while far away form the discharge location the Eulerian viewpoint seems to be appropriate.

Throughout this thesis the course sketched in figure (1.1) is followed. The aim is to devise a computer model which can effectively and efficiently be applied for modelling shallow water flow in conjunction with transport of dissolved or suspended substance. In correspondence with figure (1.1) the model consists of two modules. The present study contains:

(1) description of a numerical solution technique for flow simulation,
(2) characteristic distinctions between the Eulerian and the Lagrangian approach in transport modelling, and
(3) the description of flow data transfer between the flow and the transport module.

The organization of this thesis disintegrates into two parts. The numerical approximation technique for the flow simulation is described first continued with the investigation of three different transport models that use the flow data as input. These models are classified as (i) finite difference model (Eulerian approach), (ii) particle model (Lagrangian approach), and (iii) combined finite difference/particle model.

Flow simulation is discussed in chapter 2. To account for accurate schematization of complicated geometries the equations are written with respect to boundary-fitted coordinates. Chapter 2 discusses general notions like horizontal and vertical transformation relations and describes an approximation method for the solution of the shallow water equations in transformed coordinates. The hydrodynamic model TRISULA, made available by Delft Hydraulics, forms the basis of the numerical study. It is emphasized that the flow model is only described to generate input data for the transport model. Within the scope of this thesis accuracy and stability aspects of the flow model are of secundary interest. Detailed considerations about numerical schemes are therefore moved to appendices.

General considerations about transport modelling in shallow areas are outlined in chapter 3. The transport process is described with the so-called advection-diffusion equation including source terms and boundary conditions. Chapter 3 goes into traditional finite difference transport simulation. The remaining chapters discuss the use of particle models.

Especially for applications where concentration gradients are too steep to be solved accurately the particle model is applied. This model is also referred to as the random walk model. Chapter 4 gives the description of a particle model using the available discrete flow information with respect to a transformed grid. The model is defined such that consistency with the advection-diffusion equation is guaranteed. The flow data, used as input, are the result of the flow simulation. Three algorithms are discussed that convert particle distributions to concentrations. Numerical aspects of advective as well as diffusive displacement of particles are discussed and illustrated by means of numerical experiments. An advection algorithm is developed which traces particles along streamlines. The procedure makes use of a velocity field that satisfies the continuity equation everywhere in space. Consequently strict mass conservation is obtained. Computation of particle displacements in transformed domains introduces correction terms recognized as an additional drift contribution. These correction terms are necessary to maintain consistency with the advection-diffusion equation. Also boundary condition treatment in particle models is discussed. An algorithm is described that allows application of various boundary conditions. Special attention is paid to reflection procedures to be used at closed boundaries. Finally the implementation of decay and source contributions is discussed.

In chapter 5 a combined finite difference/particle model is proposed. It gathers the favourable aspects of both simulation techniques. The main advantage of this approach is that the model is constructed such that the finite difference simulation and the particle simulation run simultaneously. The contribution of each model in the simulated concentration distribution is determined by accuracy arguments rather than the judgement of the user.

Finally the conclusions from the present study are summarized in chapter 6.

Chapter 2

Flow model

2.1 Introduction

Transport models need flow data as input. This chapter gives a mathematical-physical description of a shallow water flow model that is used to generate flow data. In addition a finite difference discretization method of the flow model is given.

The governing equations, describing (shallow) water flow, are derived from the Navier-Stokes equations. Some simplifying assumptions are applied. These assumptions, related to shallow water flow, will be discussed in section 2.2. The governing equations result from these assumptions. One of these assumptions is related to turbulence modelling. Some remarks about turbulence models are summarized in section 2.3. Considering numerical models, shallow water equations are transformed such that the boundaries of the numerical grid follow the boundaries of the flow region. For that purpose the horizontal coordinates are transformed according to a so-called boundary-fitted coordinate transformation. In the vertical a transformation is applied such that a transformed water column is characterized by a constant water depth. The equations in transformed coordinates are discussed and explicitly given in section 2.4. Detailed mathematical derivations are moved to appendices. Section 2.5 discusses the discretized equations that are implemented in the simulation program called TRISULA. This shallow water flow solver has been made available by Delft Hydraulics. This chapter concludes in section 2.6 with concluding remarks.

2.2 Three-dimensional shallow water equations

Shallow water flows are classified as flows, in which the horizontal length scale is much larger than the vertical one. These flows are found in for instance rivers, lakes, estuaries and coastal zones. The assumption of shallowness gives rise to simplified flow equations. The main characteristic is that the pressure is, in first order, hydrostatically distributed along the vertical. The hydrostatic pressure relation is obtained by neglecting vertical acceleration.

The three-dimensional shallow water equations are derived from the Navier-Stokes equations and the equation of continuity. With the assumption of water being incompressible the latter equation reduces to

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.1}$$

with velocity components u, v and w in x-, y-, and z- direction respectively. Density variations do not violate the incompressible continuity equation (2.1). These variations will affect the velocity through the momentum equations. In particular it is assumed that density variations are small in comparison to a basic profile. The so-called Boussinesq approximation assumes that the influence of density variations is only envisaged in the buoyancy term. With this assumption the momentum equations are given by

$$\begin{aligned}
\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial x} \qquad - 2\left(\Omega_2 w - \Omega_3 v\right) + \nu_m \Delta u \\
\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial y} \qquad - 2\left(\Omega_3 u - \Omega_1 w\right) + \nu_m \Delta v \qquad (2.2) \\
\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial z} - g - 2\left(\Omega_1 v - \Omega_2 u\right) + \nu_m \Delta w
\end{aligned}$$

ρ denotes the water density, P the hydraulic pressure, ν_m the kinematic viscosity, Δ the Laplace operator, $\underline{\Omega} = \left(\Omega_1, \Omega_2, \Omega_3\right)^T$ the earth's rotation vector and g the acceleration due to gravity. $\underline{\Omega}$ is assumed to be independent of time. g is assumed constant. Equation (2.2) characterizes conservation of momentum. The time rate of change of the velocity is related to (i) a pressure force, (ii) a gravitational force (buoyancy), (iii) a Coriolis contribution (rotating coordinate system), and (iv) diffusion of momentum due to molecular exchanges between a fluid particle and the (Newtonian) fluid surrounding it.

To resolve all eddies present in the flow an extremely fine resolution in space and time is required especially if the Reynolds number, $Re = UL/\nu_m$, is large (U and L represent a characteristic velocity and length scale respectively). To avoid such an extremely fine resolution in numerical simulations an additional simplification is applied. It models the influence of small scale motion on the large scale motion (turbulence modelling). In that case, following Reynolds, the actual velocity is decomposed as

$$u = \bar{u} + u' \qquad (2.3)$$

assuming that the actual velocities, u, are close to the basic profile, $\bar{u}$. The turbulent fluctuation, given by u', is superimposed on this basic profile. The basic turbulent flow equations of motion arise if (i) (2.3) is inserted into (2.2) with respect to all three velocity components, (ii) the molecular stresses are neglected, and (iii) the resulting equations are (ensemble) averaged. Doing so, with averaged quantities specified with an overbar ($-$), equation (2.2) becomes

$$\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{w}}{\partial z} = 0 \qquad (2.4a)$$

$$\begin{aligned}
\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} + \bar{v}\frac{\partial \bar{u}}{\partial y} + \bar{w}\frac{\partial \bar{u}}{\partial z} + \frac{\partial \overline{u'u'}}{\partial x} + \frac{\partial \overline{u'v'}}{\partial y} + \frac{\partial \overline{u'w'}}{\partial z} &= -\frac{1}{\rho}\frac{\partial \bar{P}}{\partial x} + \bar{f}_x \\
\frac{\partial \bar{v}}{\partial t} + \bar{u}\frac{\partial \bar{v}}{\partial x} + \bar{v}\frac{\partial \bar{v}}{\partial y} + \bar{w}\frac{\partial \bar{v}}{\partial z} + \frac{\partial \overline{v'u'}}{\partial x} + \frac{\partial \overline{v'v'}}{\partial y} + \frac{\partial \overline{w'w'}}{\partial z} &= -\frac{1}{\rho}\frac{\partial \bar{P}}{\partial y} + \bar{f}_y \qquad (2.4b) \\
\frac{\partial \bar{w}}{\partial t} + \bar{u}\frac{\partial \bar{w}}{\partial x} + \bar{v}\frac{\partial \bar{w}}{\partial y} + \bar{w}\frac{\partial \bar{w}}{\partial z} + \frac{\partial \overline{w'u'}}{\partial x} + \frac{\partial \overline{w'v'}}{\partial y} + \frac{\partial \overline{w'w'}}{\partial z} &= -\frac{1}{\rho}\frac{\partial \bar{P}}{\partial z} + \bar{f}_z - g
\end{aligned}$$

with Coriolis term $(\bar{f}_x,\bar{f}_y,\bar{f}_z)^T = -2\mathbf{\Omega} \times (\bar{u},\bar{v},\bar{w})^T$. The arising so-called Reynolds stresses ($\overline{u'u'}$, $\overline{u'v'}$, etcetera) are responsible for a loss of momentum in the mean motion. These stresses are a-priori unknown. To obtain a well-posed problem additional equations are needed. Applying the eddy viscosity concept the Reynolds stresses are modelled according

$$\overline{u'v'} = \overline{v'u'} = -\nu_t\left(\frac{\partial \bar{v}}{\partial x} + \frac{\partial \bar{u}}{\partial y}\right) \qquad (2.5)$$

Equation (2.5) introduces the so-called eddy viscosity ν_t. Similar expressions hold for the other five stress contributions. Substitution of (2.5) into (2.4b) yields three model equations. From these equations together with the continuity equation (2.4a) the mean flow velocity components $\bar{u}$, $\bar{v}$ and $\bar{w}$ as well as the pressure $\bar{P}$ can only be solved if the eddy viscosity parameters ν_t are being specified. Many methods exist for the determination of ν_t - see section 2.3 or RODI (1984).

In shallow water applications the basic turbulent flow equations are once more simplified. Some terms are neglected as a result of dimension analysis. The following characteristic relations are distinguished:

1) $\frac{H}{L} << 1$	The characteristic horizontal length scale is much larger than the water depth (H).
2) $W = \frac{H}{L}U$	The characteristic vertical velocity component is small in comparison with the characteristic horizontal velocity component.
3) $\frac{\lvert\tau\rvert}{\rho g H} << 1$	The characteristic shear stress $\lvert\tau\rvert$ introduced by the main driving force of the fluid flow is small in comparison with the characteristic hydraulic pressure.

Under these additional assumptions the vertical momentum equation reduces to the hydrostatic pressure relation - see JIN (1993). As a consequence of 1) and 2) the terms involving $\partial\bar{w}/\partial x$ and $\partial\bar{w}/\partial y$ in the horizontal momentum equations are being neglected. The difference between the characteristic horizontal and vertical length scales gives rise to the distinction between two eddy viscosities ν_t^H and ν_t^V denoting the horizontal and vertical eddy viscosity respectively. Finally, the three-dimensional shallow water equations are resumed, dropping the overbar (-)

The continuity equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.6a}$$

The momentum equation in x - direction

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial x} + f v \\ &+ 2\frac{\partial}{\partial x}\left(\nu_t^H \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\nu_t^H \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right) + \frac{\partial}{\partial z}\left(\nu_t^V \frac{\partial u}{\partial z}\right) \end{aligned} \tag{2.6b}$$

The momentum equation in y - direction

$$\begin{aligned} \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial y} - f u \\ &+ \frac{\partial}{\partial x}\left(\nu_t^H \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right) + 2\frac{\partial}{\partial y}\left(\nu_t^H \frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial z}\left(\nu_t^V \frac{\partial v}{\partial z}\right) \end{aligned} \tag{2.6c}$$

The hydrostatic pressure relation (as a result of the simplified momentum equation in z - direction)

$$\frac{1}{\rho}\frac{\partial P}{\partial z} = -g \tag{2.6d}$$

In (2.6b) and (2.6c) a Coriolis parameter f is introduces. It is defined as

$$f = 2\Omega \sin(\phi) \tag{2.6e}$$

with Ω the angular speed of earth and ϕ the latitude. The water level is denoted with $z = \zeta$, the bottom with $z = -d$. Then, by definition, the water depth equals

$$H = \zeta + d \tag{2.7}$$

In case of constant density equation (2.6d) becomes

$$P(z) = P_a + \rho g(\zeta - z) \tag{2.8}$$

with P_a the atmospheric pressure at the water surface. The characteristic hydraulic pressure is recognized after insertion of $z = -d$ in equation (2.8).

The equations given in (2.6) are ready for numerical implementation as soon as the eddy viscosities ν_t^H and ν_t^V are quantified and the boundary conditions are given. The boundary conditions specify additional driving forces of the fluid flow, such as wind shear stresses exerted on the free water surface or tidal influences specified at open boundaries. Some frequently applied descriptions of eddy viscosities are reflected in the following section.

2.3 Turbulence modelling

The Navier-Stokes equations (2.2) are supposed to describe both laminar and turbulent flows. Numerical solution methods approximate these equations with respect to a predefined grid. To obtain consistent approximations the grid must be chosen such that all the characteristic motions are resolved. In practice however this is an unreasonable requirement. For instance, in computer models in ocean or continental shelf engineering the horizontal grid size is typically orders of magnitude larger than the smallest physical characteristic length scale (a typical grid size is about a few kilometres). Consequently, all significant processes with a length scale smaller than the chosen grid size must be subgrid processes that are to be parametrized. Following the procedure sketched in the previous section these processes are modelled by the eddy viscosity concept. A thorough survey of existing turbulence models is beyond the scope of this thesis. The reader is referred to e.g., LESIEUR (1990) or RODI (1984) for practical applications. In this section some possible choices for the arising eddy viscosities are outlined.

Especially in shallow water embayment, such as continental shelves, the effect of the horizontal eddy viscosity (ν_t^H) on the flow pattern is much smaller than the effect of the vertical one (ν_t^V). The effect of small horizontal viscosity terms in numerical models is often obscured by numerical diffusion. Many shallow water flow solvers presume

$$\nu_t^H = \text{constant} \tag{2.9}$$

The vertical eddy viscosity is usually expressed as the product of a length scale and a velocity scale - see PRANDTL (1925). This mixing-length argument is written as

$$\nu_t^V = l_m \, v \tag{2.10}$$

l_m denotes the mixing length. The velocity v characterizes the turbulent fluctuations. To form a good compromise between generality and economy of use for hydrodynamic problems, expression (2.10) is usually submitted to one of the following procedures - see RODI (1984):

1) zero-equation model (algebraic model)
   The square root of the turbulent kinetic energy (per unit mass), $\sqrt{k}$, is related to the velocity scale v, such that

$$\nu_t^V = c_\nu' l_m \sqrt{k} \tag{2.11}$$

   c_ν' arises as a model parameter. Equation (2.11) is known as the Kolmogorov-Prandtl expression. In zero-equation models algebraic expressions are used for l_m and k. One of the simplest choices is given by the Bakhmetev (length scale) distribution which yields a parabolic vertical eddy viscosity

$$\nu_t^V = \kappa z \sqrt{1 - \frac{z}{H}} \sqrt{k} \tag{2.12}$$

   with κ the von Karman constant ($\kappa \approx 0.4$). An algebraic expression for k involves the friction velocity at bottom and water surface. Then, (2.12) is referred to as the Prandtl mixing length hypothesis.
2) one-equation model ($k-L$ model)
   As in the zero-equation model the length scale distribution is prescribed. However, k is solved from a transport equation. This equation models the rate of change of turbulent kinetic energy by convective and diffusive transport, production by shear, and viscous dissipation. It is noted that in shallow water flow the exchange between k and potential energy is explicitly taken into account (buoyancy force) - see e.g., RODI (1980). The solution of the transport equation depends on the boundary conditions. Especially in three-dimensional models with a small vertical length scale an incorrect boundary condition at the free water surface affects the vertical distribution. A correct description of boundary conditions is not obvious.
3) two-equation model ($k-\epsilon$ model)
   In two-equation models two transport equations are solved for turbulent quantities. The most widely used model uses transport equations for k (turbulent kinetic energy) and ϵ (dissipation of turbulent kinetic energy), in which ϵ is defined as

$$\epsilon = C_D \frac{k^{\frac{3}{2}}}{l_m} \tag{2.13}$$

With definition (2.13) the expression for the vertical eddy viscosity becomes, with empirical model parameter c_ν

$$\nu_t^V = c_\nu \frac{k^2}{\epsilon} \tag{2.14}$$

The defined eddy viscosities can be inserted in the three-dimensional shallow water equations (2.6b) and (2.6c). It results in a closed set of equations from which the velocity field as well as the pressure distribution can be solved numerically. The effect of density differences is taken into account if two additional transport equations are solved. These equations simulate heat and salt transport. The water density is given by an algebraic expression (equation of state), relating water density to temperature and salinity. It is noted that in such stratified flows the mixing length is suppressed. This phenomenon is modelled in the zero-equation model by the introduction of a damping function to be appointed in (2.11). This function depends on the gradient Richardson number defined as the ratio between the buoyancy and shear production terms. In the $k-\epsilon$ model the density influences are modelled by the buoyancy terms in the transport equation for k.

Returning to numerical simulation of three-dimensional shallow water flow, it is noted that most engineering applications deal with complicated flow regions such as irregular boundaries in the horizontal plane. Moreover, numerical problems are due to the free water surface (moving boundary). The numerical treatment is based upon boundary-fitted coordinates. For that purpose the shallow water equations are transformed. This transformation will be outlined in the next section.

2.4 Shallow water equations in transformed coordinates

For numerical simulations, the shallow water flow region is covered with a computational mesh. Here a mesh in Cartesian coordinates is defined as a set of connecting cubes. Distorted grids (curved or stretched) are called transformed grids. A mesh in Cartesian coordinates yields for some geometries typical zigzag boundaries. Such a representation of boundaries gives rise to inaccurate simulations. The main motivation for dealing with transformed grids is based on continuity arguments. For instance, the grid drawn in figure (2.1) yields velocity vectors in the shaded grid cells with inaccurate orientation. Moreover, curved channel flows covered with a mesh in Cartesian coordinates is not able to give an accurate representation of the stream width which affects the accuracy of the volume flux through a cross section. A second order effect is coming from the boundary treatment. Near zigzag boundaries artificial friction is introduced and a special discretization method is necessary for the advection terms in the momentum equations - see STELLING (1984). For these reasons the use of boundary-fitted coordinates is often more efficient. In three-dimensional simulations the representation of complex bottom topographies becomes accurate if the sigma transformation is applied. This transformation, as introduced in PHILLIPS (1957), stretches the vertical direction, such that the transformed water depth is constant in space and time. Since the water level usually changes in time, the sigma transformation is time dependent. Illustrations of numerical grids are sketched in figure (2.2).

Subsequently, the three-dimensional shallow water equations will be subjected to an orthogonal curvilinear transformation in the horizontal plane and the sigma transformation in the vertical. The equations in transformed coordinates will be derived. Two transformation routes are possible. The first route treats the horizontal and the vertical transformation successively. The second route treats both transformations simultaneously. Here the second alternative is considered.

The Cartesian coordinate system is denoted with (x,y,z). The transformed coordinate system is given by (ξ,η,σ). The transformation is expressed by

$$\begin{cases} x = x(\xi,\eta) \\ y = y(\xi,\eta) \\ z = \sigma H(x,y,t) + \zeta(x,y,t) \end{cases} \tag{2.15}$$

in which ζ denotes the water level elevation above the plane of reference $z = 0$. The bottom topography is expressed by d below the same plane of reference - see equation (2.7). The sigma transformation is such that the transformed water level coincides with $\sigma = 0$, while the transformed bottom topography is given by $\sigma = -1$ - see figure (2.2b). The first two

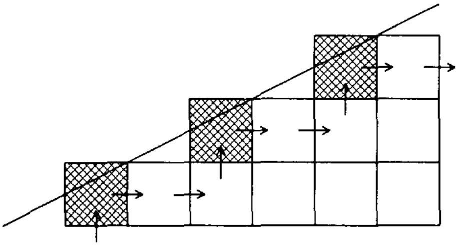


Figure (2.1): Zigzag boundary.

Direction of resulting velocity vectors in shaded cells equals ≈ 45° instead of the tangent to the physical boundary.

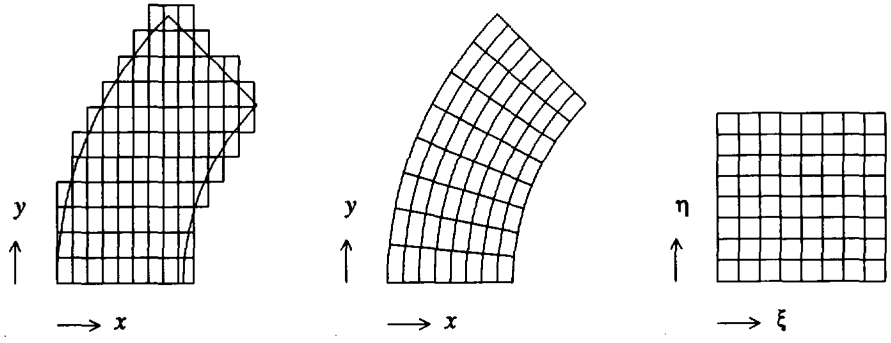


Figure (2.2a): Typical meshes in the horizontal plane (e.g., part of river bent).

LEFT: mesh with zigzag boundaries,

CENTRE: boundary fitted (orthogonal) mesh,

RIGHT: boundary fitted mesh in transformed space.

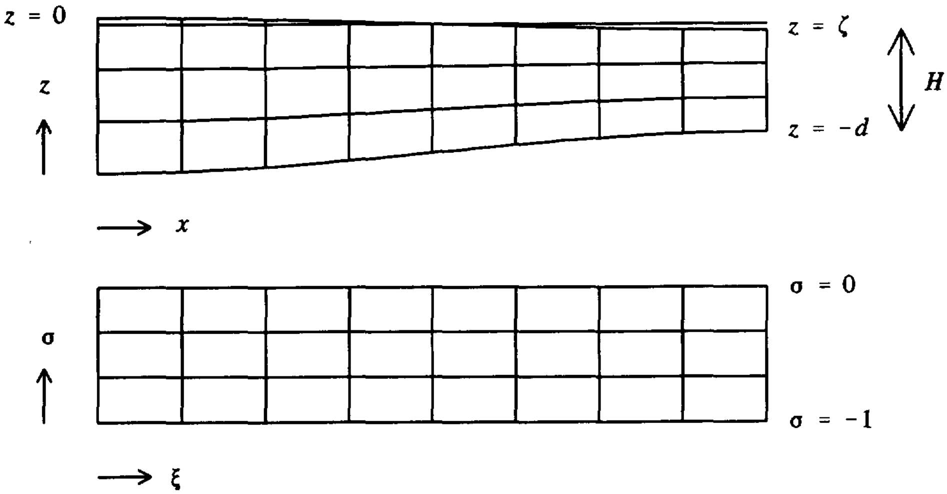


Figure (2.2b): Computational mesh in the vertical plane with sigma transformation $\sigma = \frac{z - \zeta}{H}$.

TOP: sigma-grid in physical space with horizontal plane of reference $z = 0$,

BOTTOM: sigma-grid in transformed space.

relations of (2.15) account for the orthogonal transformation in the horizontal plane.

The coordinates (x,y,z) carry dimension meter (m). The coordinates (ξ,η,σ) are dimensionless. The change in coordinate system introduces transformed velocity components. In the transformed space these components are measured along the transformed coordinate axes. The usual transformation relations derived from tensor analysis introduce so-called contravariant velocity components U, V and W - see THOMPSON *et al.* (1985). These components carry dimension s^{-1}. With $\underline{e}_i$ (i=1,2,3) the usual basis elements and $\underline{a}_i$ (i=1,2,3) the so-called covariant base vectors, i.e., the tangent vectors to the three coordinate lines, a velocity vector is expressed as

$$u\underline{e}_1 + v\underline{e}_2 + w\underline{e}_3 = U\underline{a}_1 + V\underline{a}_2 + W\underline{a}_3 \tag{2.16}$$

By definition - see figure (2.3),

$$\underline{a}_1 = \begin{pmatrix} \dfrac{\partial x}{\partial \xi} \\ \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial z}{\partial \xi} \end{pmatrix} \qquad \underline{a}_2 = \begin{pmatrix} \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \eta} \\ \dfrac{\partial z}{\partial \eta} \end{pmatrix} \qquad \underline{a}_3 = \begin{pmatrix} \dfrac{\partial x}{\partial \sigma} \\ \dfrac{\partial y}{\partial \sigma} \\ \dfrac{\partial z}{\partial \sigma} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dfrac{\partial z}{\partial \sigma} \end{pmatrix} \tag{2.17}$$

The contravariant velocity components U, V and W are related to u, v and w by the introduction of the contravariant base vectors $\underline{a}^i$ (i=1,2,3). These vectors are defined as normal vectors to the three coordinate surfaces. Therefore,

$$\underline{a}_i \cdot \underline{a}^j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \tag{2.18}$$

For numerical simulation applied to three-dimensional shallow water flow, different velocity components are introduced. These components are chosen such that:

(i) each component has dimension m/s, and

(ii) volume fluxes through grid cell sides are easily computed numerically. For orthogonal horizontal transformations two independent velocity components are defined perpendicular to the vertical sides of a grid cell. The remaining velocity component is vertically orientated.

Metric relations contribute to requirement (i). In the horizontal plane the following metric relations are defined

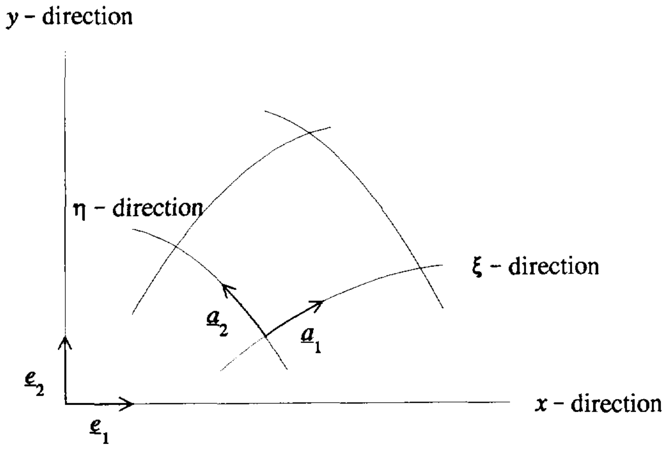


Figure (2.3): Covariant base vectors in a two-dimensional nonorthogonal curved grid.

$$\left.\begin{aligned}\sqrt{G_{\xi\xi}} &= \sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2}\\ \sqrt{G_{\eta\eta}} &= \sqrt{\left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2}\end{aligned}\right\}\quad \text{horizontal measures of lengths} \tag{2.19}$$

$$\left.\sqrt{G} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right\}\quad \text{Jacobian of horizontal transformation} \tag{2.20}$$

Subsequently, the following velocity components are defined

$$\left\{\begin{aligned} u_\xi &= \frac{1}{\|\underline{a}^1\|}U\\ u_\eta &= \frac{1}{\|\underline{a}^2\|}V\\ \omega &= \|\underline{a}_3\|W \end{aligned}\right. \tag{2.21}$$

These components satisfy the mentioned requirements. An illustration for a nonorthogonal transformation is given in figure (2.4). If the horizontal transformation is orthogonal the orientation of the U and V components coincide with those of the u_ξ and u_η components respectively.

Now that the reference velocities (2.21) and metric relations (2.19) are defined the shallow water equations are transformed by extensive use of the chain rule. Details of the derivations

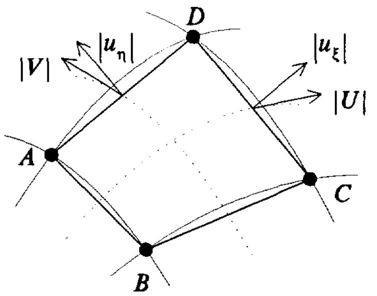


Figure (2.4a): Velocity in horizontal plane.

$\|\vec{CD}\| = \sqrt{G_{\eta\eta}}$, $\|\vec{AD}\| = \sqrt{G_{\xi\xi}}$.

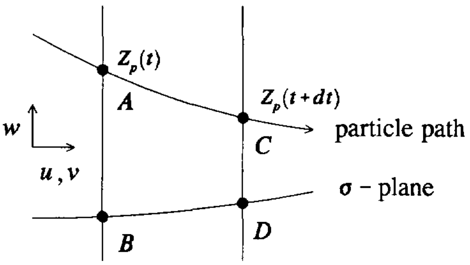


Figure (2.4b): Velocity in vertical plane.

$w dt = Z_p(t+dt) - Z_p(t)$,

$\omega dt = \|\vec{CD}\| - \|\vec{AB}\|$.

are moved to appendices. Here the final result is given.

In a moving frame of reference, induced by the sigma transformation, equation (2.1) is transformed according - see appendix A,

$$\frac{1}{J}\left(\frac{\partial J}{\partial t} + \frac{\partial}{\partial \xi}(JU) + \frac{\partial}{\partial \eta}(JV) + \frac{\partial}{\partial \sigma}(JW)\right) = 0 \tag{2.22}$$

in which J represents the Jacobian of the overall transformation. J is expressed in terms of H and $\sqrt{G}$ in equation (B.4) of appendix B. Inserting $J = H\sqrt{G}$ together with the velocity components of (2.21) yields

The continuity equation

$$\frac{\partial H}{\partial t} + \frac{1}{\sqrt{G}}\left(\frac{\partial}{\partial \xi}\left(H\sqrt{G_{\eta\eta}}u_\xi\right) + \frac{\partial}{\partial \eta}\left(H\sqrt{G_{\xi\xi}}u_\eta\right)\right) + \frac{\partial \omega}{\partial \sigma} = 0 \tag{2.23a}$$

It is noted that equation (2.23a) is also valid if the horizontal transformation is nonorthogonal. A positive volume flux through a water column results in a water level reduction - see figure (2.5). For that reason u_ξ and u_η are measured in horizontal direction, i.e., normal to the vertical sides of the water column. The definition of u_ξ and u_η given in (2.21) meets this requirement.

The derivation of the shallow water equations in transformed coordinates is given in appendix B. Here the considerations mentioned above are used to obtain the transformed equations in case of an orthogonal curvilinear transformation in the horizontal plane.

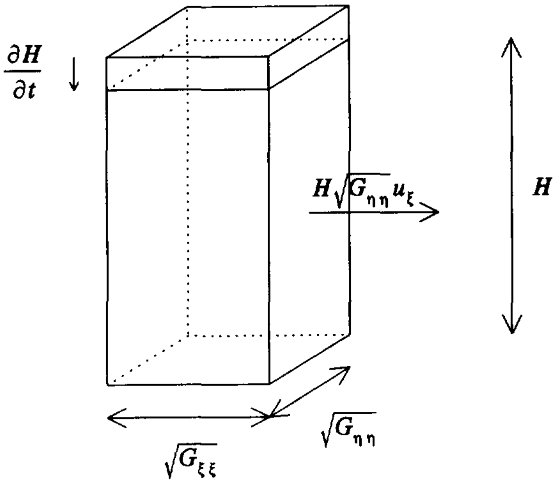


Figure (2.5): Water column. Height of water column depends on volume flux through vertical boundaries.

The momentum equation in ξ - direction

$$\begin{aligned}
&\frac{\partial u_\xi}{\partial t} + \frac{u_\xi}{\sqrt{G_{\xi\xi}}}\frac{\partial u_\xi}{\partial \xi} + \frac{u_\eta}{\sqrt{G_{\eta\eta}}}\frac{\partial u_\xi}{\partial \eta} + \frac{\omega}{H}\frac{\partial u_\xi}{\partial \sigma} \\
&+ \frac{u_\xi u_\eta}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial \eta}\left(\sqrt{G_{\xi\xi}}\right) - \frac{{u_\eta}^2}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial \xi}\left(\sqrt{G_{\eta\eta}}\right) \\
&= -\frac{1}{\tilde{\rho}}\frac{1}{\sqrt{G_{\xi\xi}}}\left(\frac{\partial \tilde{P}}{\partial \xi} + \frac{\partial \tilde{P}}{\partial \sigma}\frac{\partial \sigma}{\partial \xi}\right) + f u_\eta \\
&+ D_\xi\left(\nu_t^H\right) + \frac{1}{H^2}\frac{\partial}{\partial \sigma}\left(\nu_t^V\frac{\partial u_\xi}{\partial \sigma}\right)
\end{aligned} \tag{2.23b}$$

The momentum equation in η - direction

$$\begin{aligned}
&\frac{\partial u_\eta}{\partial t} + \frac{u_\xi}{\sqrt{G_{\xi\xi}}}\frac{\partial u_\eta}{\partial \xi} + \frac{u_\eta}{\sqrt{G_{\eta\eta}}}\frac{\partial u_\eta}{\partial \eta} + \frac{\omega}{H}\frac{\partial u_\eta}{\partial \sigma} \\
&+ \frac{u_\xi u_\eta}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial \xi}\left(\sqrt{G_{\eta\eta}}\right) - \frac{{u_\xi}^2}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial \eta}\left(\sqrt{G_{\xi\xi}}\right) \\
&= -\frac{1}{\tilde{\rho}}\frac{1}{\sqrt{G_{\eta\eta}}}\left(\frac{\partial \tilde{P}}{\partial \eta} + \frac{\partial \tilde{P}}{\partial \sigma}\frac{\partial \sigma}{\partial \eta}\right) - f u_\xi \\
&+ D_\eta\left(\nu_t^H\right) + \frac{1}{H^2}\frac{\partial}{\partial \sigma}\left(\nu_t^V\frac{\partial u_\eta}{\partial \sigma}\right)
\end{aligned} \tag{2.23c}$$

The hydrostatic pressure relation

$$\tilde{P} = P_a + gH\int_{\sigma}^{0} \tilde{\rho}(\xi,\eta,\sigma',t)\,d\sigma' \tag{2.23d}$$

The contribution of (i) advection, (ii) hydraulic pressure, (iii) Coriolis, (iv) horizontal eddy viscosity, and (v) vertical eddy viscosity are recognized. In addition curvature terms are included. These terms arise due to the definition of the transformed velocity components. Furthermore it is noted that the transformation is time dependent. Therefore the time derivative has also been transformed. It yields a contribution of the grid-point speed $(\partial z/\partial t)_{(\xi,\eta,\sigma)}$. This contribution is absorbed in the definition of the ω - velocity. The horizontal viscosity terms $D_\xi\left(\nu_t^H\right)$ and $D_\eta\left(\nu_t^H\right)$ are not given explicitly. The amount of terms appears to be disorderly - see appendix B and C. At this stage it is reminded that transformed equations are derived to simplify the numerical treatment. For that reason the horizontal viscosity terms will be treated with respect to the Cartesian coordinate system. For a non-equidistant rectangular grid the expressions for the horizontal eddy viscosity terms $D_\xi\left(\nu_t^H\right)$ and $D_\eta\left(\nu_t^H\right)$ remain orderly. Then $\partial y/\partial\xi = \partial x/\partial\eta = 0$ and all curvature terms vanish. The expressions become

$$\begin{cases} D_\xi = \dfrac{1}{\sqrt{G_{\xi\xi}}}\left(\dfrac{\partial\tau_{\xi\xi}}{\partial\xi} + \dfrac{\partial\tau_{\xi\xi}}{\partial\sigma}\dfrac{\partial\sigma}{\partial\xi}\right) + \dfrac{1}{\sqrt{G_{\eta\eta}}}\left(\dfrac{\partial\tau_{\xi\eta}}{\partial\eta} + \dfrac{\partial\tau_{\xi\eta}}{\partial\sigma}\dfrac{\partial\sigma}{\partial\eta}\right) \\[2ex] D_\eta = \dfrac{1}{\sqrt{G_{\xi\xi}}}\left(\dfrac{\partial\tau_{\xi\eta}}{\partial\xi} + \dfrac{\partial\tau_{\xi\eta}}{\partial\sigma}\dfrac{\partial\sigma}{\partial\xi}\right) + \dfrac{1}{\sqrt{G_{\eta\eta}}}\left(\dfrac{\partial\tau_{\eta\eta}}{\partial\eta} + \dfrac{\partial\tau_{\eta\eta}}{\partial\sigma}\dfrac{\partial\sigma}{\partial\eta}\right) \end{cases} \tag{2.24}$$

The arising transformed Reynolds stresses satisfy (also with respect to a rectangular coordinate system in the horizontal plane)

$$\begin{cases} \tau_{\xi\xi} = \dfrac{2\nu_t^H}{\sqrt{G_{\xi\xi}}}\left(\dfrac{\partial u_\xi}{\partial\xi} + \dfrac{\partial u_\xi}{\partial\sigma}\dfrac{\partial\sigma}{\partial\xi}\right) \\[2ex] \tau_{\xi\eta} = \dfrac{\nu_t^H}{\sqrt{G_{\eta\eta}}}\left(\dfrac{\partial u_\xi}{\partial\eta} + \dfrac{\partial u_\xi}{\partial\sigma}\dfrac{\partial\sigma}{\partial\eta}\right) + \dfrac{\nu_t^H}{\sqrt{G_{\xi\xi}}}\left(\dfrac{\partial u_\eta}{\partial\xi} + \dfrac{\partial u_\eta}{\partial\sigma}\dfrac{\partial\sigma}{\partial\xi}\right) \\[2ex] \tau_{\eta\eta} = \dfrac{2\nu_t^H}{\sqrt{G_{\eta\eta}}}\left(\dfrac{\partial u_\eta}{\partial\eta} + \dfrac{\partial u_\eta}{\partial\sigma}\dfrac{\partial\sigma}{\partial\eta}\right) \end{cases} \tag{2.25}$$

Finally, it is assumed that the bottom does not change due to morphological changes. Then the water depth satisfies

$$\frac{\partial H}{\partial t} = \frac{\partial \zeta}{\partial t} \tag{2.26}$$

This expression can be inserted into the continuity equation. With equation (2.23) a complete mathematical-physical description of the three-dimensional shallow water equations is given. The description is such that its numerical implementation may make use of a boundary-fitted mesh. Further numerical considerations will be discussed in the following section.

2.5 Description of a three-dimensional shallow water solver

This section discusses a finite difference discretization method for the shallow water equations in transformed coordinates. The discretized equations are implemented within the simulation program called TRISULA. This shallow water solver has been made available by Delft Hydraulics. This system is based upon finite difference approximation methods. These methods involve the following aspects:

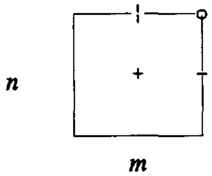
- choice of a grid,
- spatial discretization, and
- time integration.

The used scheme is mainly based on studies of LEENDERTSE (1989) and STELLING (1984). Discretization in space and time are discussed separately. A discussion about the numerical treatment of advection terms is given in appendix D.

Grid choice

To solve shallow water equations numerically a staggered boundary-fitted grid is introduced. A grid cell is labelled with (m,n,k). $k = 1$ represents the top layer. The notation is according to the notation of section 2.4. The following grid staggering is used:

- in the (ξ,η)-plane with σ is constant



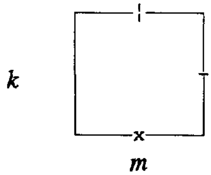
- location of $(u_\xi)_{m+\frac{1}{2},n,k}$ and $\sqrt{G_{\eta\eta}}\big|_{m+\frac{1}{2},n,k}$

| location of $(u_\eta)_{m,n+\frac{1}{2},k}$ and $\sqrt{G_{\xi\xi}}\big|_{m,n+\frac{1}{2},k}$

+ location of $\zeta_{m,n}$ and $\left(\nu_t^H\right)_{m,n,k}$

o location of $d_{m+\frac{1}{2},n+\frac{1}{2}}$ with $d = H - \zeta$

- in the (ξ,σ)-plane with η is constant



– location of $(u_\xi)_{m+\frac{1}{2},n,k}$

¦ location of $\omega_{m,n,k-\frac{1}{2}}$

x location of $\nu_{t\ m,n,k+\frac{1}{2}}^{V}$

- the staggering in the (η,σ)-plane, with ξ is constant, is identical.

At the u_ξ and u_η location also roughness quantities are defined. These quantities characterize the status of the grid point. A physically realistic value of the roughness parameter indicates that the location is wet. A negative value denotes that the position is dry, i.e., left out of computations.

The choice of the location of the discrete variables is mainly based on accuracy considerations. To assure that the numerical solution of $\nabla P = 0$ does not show nonphysical oscillations the pressure is computed at the centre of each grid cell, while the velocity components are ordered halfway the grid cell sides - see FLETCHER (1988). In shallow water flow modelling the pressure is related to the water level elevation ζ. Consequently, ζ is also computed at centred locations. Vertical viscosity terms arise in the momentum equations in ξ - and η - direction. Therefore $\partial/\partial\sigma\left(\nu_t^V\left(\partial u_\xi/\partial\sigma\right)\right)$ must be evaluated at every u_ξ - location. A first order discretization for a physical quantity φ,

$$\left.\frac{\partial\varphi}{\partial\sigma}\right|_k = \frac{\varphi_{k-\frac{1}{2}} - \varphi_{k+\frac{1}{2}}}{\Delta\sigma_k} \tag{2.27}$$

with $\Delta\sigma_k$ the relative layer thickness of layer k, immediately gives rise to position ν_t^V in the horizontal plane $k+\frac{1}{2}$. In general it is not known whether $\nu_t^V\left(\partial u_\xi/\partial\sigma\right)$ or $\nu_t^V\left(\partial u_\eta/\partial\sigma\right)$ is dominant. So, ν_t^V is located at the centre of the bottom side, i.e., $\nu_{t\ m,n,k+\frac{1}{2}}^{V}$.

The grid is related to the physical domain such that the boundary of the physical domain intersects the u_ξ - and u_η - locations. Doing so, boundary conditions are easily imposed. Finally, the grid staggering is also responsible for an acceptable computing time.

Spatial discretization

Spatial derivatives are approximated by means of finite differencing. Each equation is approximated at fixed points within the Eulerian mesh. Average operators have to be used to address a physical quantity to the point under consideration. For instance,

$$\left(u_\xi\right)_{m,n,k} = \tfrac{1}{2}\left(\left(u_\xi\right)_{m+\frac{1}{2},n,k} + \left(u_\xi\right)_{m-\frac{1}{2},n,k}\right) \tag{2.28a}$$

$$\zeta_{m+\frac{1}{2},n} = \tfrac{1}{2}\left(\zeta_{m,n} + \zeta_{m+1,n}\right) \tag{2.28b}$$

Expression (2.28b) uses information of two adjacent grid cells.

It is already mentioned in section 2.4 that the implementation of the transformed horizontal viscosity term may give rise to numerical problems. If the fully transformed expressions of the horizontal viscosity terms are discretized numerical truncation errors are responsible for artificial vertical spreading. This is especially recognized in vertically homogenously mixed reservoirs with constant salt density. Due to truncation errors a density gradient (which should be absent) is computed. This numerical artifact causes a vertical momentum transfer. Consequently, a density-driven flow is initiated. It results in a vertical nonphysical circulation. For that reason it is important to treat horizontal viscosity terms with minimal artificial spreading. If the horizontal viscous fluxes are totally performed in the Cartesian coordinate system these numerical artifacts are overcome. The same argument holds for the numerical treatment of the pressure contribution. It is also computed in Cartesian coordinates. The detailed numerical treatment is found in STELLING and KESTER (1994).

The numerical treatment of advection terms is discussed in appendix D.

With $\Delta\xi = \Delta\eta = 1$, the increments of arc length $\sqrt{G_{\xi\xi}}$ and $\sqrt{G_{\eta\eta}}$ represent the distance between depth locations measured along the lines η = constant and ξ = constant respectively. These quantities are computed at u_η- and u_ξ-locations respectively. Discretization of curvature terms in orthogonal grids becomes straightforward.

Integration in time

To obtain an efficient numerical scheme only tridiagonal sets of equations are considered. This is realized by a two step method in which the equations are solved along lines parallel to the ξ- and η-axes at the first and second step respectively. The procedure marches forward in discrete time steps with increment Δt. Grid quantities are computed at time levels $t + \Delta t$ as well as at intermediate time levels $t + \tfrac{1}{2}\Delta t$. Both steps are treated implicitly except for the horizontal viscous contributions. Implicit numerical treatment of nonlinear terms gives rise to an iteration procedure, again to assure that only tridiagonal matrix systems have to be considered. The procedure is summarized as follows:

Step 1: Marching from time level t to time level $t + \tfrac{1}{2}\Delta t$.

(i) First, u_η - velocities are computed implicitly from the momentum equation in η - direction. All other quantities are considered at the old time level.

(ii) Second, the momentum equation in ξ - direction and the continuity equation are integrated over the layer thickness. The result of summation of the layer-equations is interpreted as depth-averaged equations. These nonlinear equations are treated implicitly. An iterative procedure yields the approximate water level elevation.

(iii) Third, the u_ξ - velocities are computed implicitly from the momentum equation in ξ - direction.

(iv) Fourth, ω - velocities are determined from the continuity equation.

Step 2: Marching from time level $t + ½\Delta t$ to time level $t + \Delta t$.

The second step is similar to the first, except that u_ξ and u_η as well as ξ and η are interchanged.

The discretization method is second order accurate in space and time, except for the ω - velocities. The ω - velocities are first order accurate. Due to the explicit treatment of the horizontal viscous terms the scheme looses its property of being unconditionally stable.

Detailed discretized equations are reported in the TRISULA documentation guide - see TRISULA (1988) and STELLING and KESTER (1994). For later reference the continuity equation will be discussed in more detail.

The discretized continuity equation

Since u_ξ, u_η and ζ are computed at stage (i), (ii) and (iii) of both steps, the ω - velocities are simply computed from the continuity equation. In consideration of the derivation of a Lagrangian advection solver to be used in the particle transport model of section 4.2.4, the precise discretized continuity equation will be given. The continuity equation is evaluated at the centre of cell (m,n,k). Since the momentum equation is nonlinear and only tridiagonal matrix systems are considered, an iteration procedure is applied. The iteration index is denoted with superscript $[q]$. The discretization scheme is similar to the one reported in STELLING (1984).

Step 1, with $\zeta^{[0]} = \zeta^l$ and $u_\xi^{[0]} = u_\xi^l$, $q = 1,..,Q$:

$$\begin{aligned}
&\frac{\zeta^{[q]}_{m,n} - \zeta^{l}_{m,n}}{\tfrac{1}{2}\Delta t} + \frac{1}{\sqrt{G}\big|_{m,n}} \left[\left(H\sqrt{G_{\xi\xi}}\,u_\eta\right)^{l}_{m,n+\frac{1}{2},k} - \left(H\sqrt{G_{\xi\xi}}\,u_\eta\right)^{l}_{m,n-\frac{1}{2},k} \right] \\
&+ \frac{1}{\sqrt{G}\big|_{m,n}} \left[\left(\left(d\sqrt{G_{\eta\eta}}\,u_\xi\right)^{[q]}_{m+\frac{1}{2},n,k} - \left(d\sqrt{G_{\eta\eta}}\,u_\xi\right)^{[q]}_{m-\frac{1}{2},n,k} \right) \right. \\
&\qquad + \zeta^{[q-1]}_{m,n} \left(\left(\sqrt{G_{\eta\eta}}\,u_\xi\right)^{[q]}_{m+\frac{1}{2},n,k} - \left(\sqrt{G_{\eta\eta}}\,u_\xi\right)^{[q]}_{m-\frac{1}{2},n,k} \right) \\
&\qquad \left. + \left(\sqrt{G_{\eta\eta}}\,u_\xi\right)^{[q-1]}_{m,n,k} \left(\zeta^{[q]}_{m+\frac{1}{2},n} - \zeta^{[q]}_{m-\frac{1}{2},n} \right) \right] \\
&+ \frac{\omega^{[q]}_{m,n,k-\frac{1}{2}} - \omega^{[q]}_{m,n,k+\frac{1}{2}}}{\Delta\sigma_k} \\
&= 0
\end{aligned} \tag{2.29}$$

Step 2, with $\zeta^{[0]} = \zeta^{l+\frac{1}{2}}$ and $u^{[0]}_\eta = u^{l+\frac{1}{2}}_\eta$, $q = 1,..,\mathrm{Q}$:

$$\begin{aligned}
&\frac{\zeta^{[q]}_{m,n} - \zeta^{l+\frac{1}{2}}_{m,n}}{\tfrac{1}{2}\Delta t} + \frac{1}{\sqrt{G}\big|_{m,n}} \left[\left(H\sqrt{G_{\eta\eta}}\,u_\xi\right)^{l+\frac{1}{2}}_{m+\frac{1}{2},n,k} - \left(H\sqrt{G_{\eta\eta}}\,u_\xi\right)^{l+\frac{1}{2}}_{m-\frac{1}{2},n,k} \right] \\
&+ \frac{1}{\sqrt{G}\big|_{m,n}} \left[\left(\left(d\sqrt{G_{\xi\xi}}\,u_\eta\right)^{[q]}_{m,n+\frac{1}{2},k} - \left(d\sqrt{G_{\xi\xi}}\,u_\eta\right)^{[q]}_{m,n-\frac{1}{2},k} \right) \right. \\
&\qquad + \zeta^{[q-1]}_{m,n} \left(\left(\sqrt{G_{\xi\xi}}\,u_\eta\right)^{[q]}_{m,n+\frac{1}{2},k} - \left(\sqrt{G_{\xi\xi}}\,u_\eta\right)^{[q]}_{m,n-\frac{1}{2},k} \right) \\
&\qquad \left. + \left(\sqrt{G_{\xi\xi}}\,u_\eta\right)^{[q-1]}_{m,n,k} \left(\zeta^{[q]}_{m,n+\frac{1}{2}} - \zeta^{[q]}_{m,n-\frac{1}{2}} \right) \right] \\
&+ \frac{\omega^{[q]}_{m,n,k-\frac{1}{2}} - \omega^{[q]}_{m,n,k+\frac{1}{2}}}{\Delta\sigma_k} \\
&= 0
\end{aligned} \tag{2.30}$$

From these equations $\omega_{m,n,k+\frac{1}{2}}$ is solved for $k = 1,..,K-1$, starting at the toplayer and proceeding downward with computation. It takes into account the kinematic boundary condition at the free water surface. This condition expresses that the vertical velocity and acceleration are in phase with those at the water surface - see appendix B,

$$\omega_{m,n,\frac{1}{2}} = \omega\big|_{\text{water level}} = 0 \tag{2.31}$$

The continuity equation is solved correctly if (2.29) and (2.30) yield $\omega_{m,n,K+\frac{1}{2}} = \omega\big|_{\text{bottom}} = 0$. Because of the iteration procedure small deviations may arise. This is overcome by a water level correction (continuity correction). Then $u^{[Q]}$ is accepted as the correct velocity component and once more the depth averaged continuity equation is solved for $\zeta^{l+\frac{1}{2}}$ in step 1 and for ζ^{l+1} in step 2.

The numerical scheme outlined above is implemented in the simulation program TRISULA. This source is used to generate the necessary input for transport simulation.

2.6 Concluding remarks

The three-dimensional shallow water equations are derived taking into account the Boussinesq approximation and the hydrostatic pressure assumption. Turbulence is modelled using the eddy viscosity concept. The arising eddy viscosities arise from either zero-, one- or two-equation models. The governing equations are subjected to a boundary-fitted coordinate transformation to obtain the shallow water equations in transformed coordinates. Two of the used velocity components match the contravariant velocity components of the orthogonal horizontal transformation, while in the vertical the ω - velocity is introduced. A finite difference discretization method is described using a spatial staggered grid.

Chapter 3

Three-dimensional transport modelling

3.1 Introduction

To analyze environmental water problems, ranging from water pollution due to sewer systems or other outfall problems, sedimentation, oil spill problems, and so on, transport models are required to predict the advection and dispersion of pollutants. These models are developed to support control strategies for problems related to surface water quality, salt intrusion, siltation, and so forth.

Section 3.2 gives the mathematical-physical description of transport processes including boundary conditions in relation to different examples. The arising partial differential equation can not be solved analytically and as a consequence numerical solution techniques must be considered. Basically two mathematical simulation techniques exist, classified as Eulerian methods and Lagrangian methods. The Eulerian method describes what happens at a fixed point in space. The Lagrangian method implies a moving coordinate system where the coordinates move along with the flow. In that respect a particle method, the subject of chapter 4, is a Lagrangian method. Eulerian methods cover the flow region with a computational mesh. Each cell may be regarded as a control volume upon which conservation principles are applied. Using this control volume approach and Gauss' divergence theorem the so-called finite volume methods arise. It is also possible to discretize time and spatial coordinates as they arise in the differential equation that governs the conservation principle. It leads to the construction of finite difference methods. Section 3.3 discusses such an approach. The discretization method and its numerical consequences will be summarized.

3.2 The advection-diffusion equation

The mathematical description of transport processes is based upon principles applied to a control volume. The time rate of change of the concentration of transported substance within this control volume is the net result of (i) concentration fluxes through the sides of the control volume, (ii) production, and (iii) decay in the control volume. Fluxes are described in terms of advection and diffusion. Production and decay depend on physical characteristics. As a result a (generalized) advection-diffusion equation is assumed. In three dimensions it is given by

$$\frac{\partial C}{\partial t} = -\sum_{i=1}^{3} \frac{\partial}{\partial x_i}(u_i C) + \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial C}{\partial x_j}\right) + S + Q \tag{3.1}$$

with C the concentration of transported substance, u_i is the velocity component in x_i - direction, D_{ij} the eddy diffusivity coefficient in x_i - direction due to the component of concentration gradient in x_j - direction, S and Q are terms describing sinks and sources.

The advection-diffusion equation is widely applied in a variety of engineering applications. Equation (3.1) with $S = Q = 0$ arises as a salt balance equation in estuaries - see DYER (1977). It is also used for simulation purposes of dissolved matter transport. Transport of nonconservative substances is often studied in surface water quality problems. Again equation (3.1) is considered with $S = 0$ and $Q = -\gamma C$ where γ represents a positive exponential decay coefficient - see ONISHI (1981). In suspended sediment transport problems a settling velocity w_s is introduced. Equation (3.1) is then used with $S = -\partial(w_s C)/\partial x_3$ - see KRANCK (1984). Examples of suspended (cohesive) sediment models that make use of the advection-diffusion equation with some appropriate source term are given in e.g., COLE and MILES (1983), HAYTER and MEHTA (1986) and O'CONNER and NICHOLSON (1986). Suspended and dissolved impurities may enter the water due to natural sources such as rainfall. It gives rise to a production Q that might be continuous or instantaneous both in space and/or time. Source terms arising from chemical industrial waste from a dye plant or calamities are recognized as point sources that are possibly continuous in time.

In addition to equation (3.1) 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) boundaries positioned at for instance tidal inlets. Dirichlet boundary conditions are often imposed at for instance bottom or open boundaries to prescribe a fixed concentration. In regions far away from discharge locations it is sometimes justified to prescribe $C = 0$ at open boundaries. If an equilibrium bed concentration C_e is assumed a bottom boundary condition in sediment transport problems may be $C = C_e$. At closed boundaries a Neumann boundary condition is often prescribed which excludes mass transfer through such a boundary. Mathematically this is denoted with $\partial C/\partial n = 0$ with n the normal vector to the boundary. Other bottom boundary conditions address deposition, erosion and consolidation - see RIJN (1989).

Initial conditions address a concentration distribution measured at initial state and account for instantaneous discharges of, for instance, industrial waste.

It is clear that numerical investigations are only useful if a proper representation of the corresponding physical processes are addressed. Within this context FISCHER (1981) states: "*It also seems likely, however, that predictive ability is not strongly dependent on the choice of numerical algorithm, since the limits we find on predictive ability are related primarily to gaps in physical understanding and ability to model detail, rather than the choice of numerical method*". However, although numerical methods are always constructed such that the approximation is consistent with the solution of the mathematical-physical description, different numerical models may give rise to different accuracy properties. Accuracy arguments should play a part in the choice of the numerical model. The choice of a numerical method that solves equation (3.1) together with initial and boundary conditions is also based on consistency, stability, efficiency and robustness. With efficiency it is meant that an increase of computational effort decreases the error at least proportionally. A robust model always yields physical realistic solutions.

In case of Eulerian methods a partial differential equation is approximated by means of a set of algebraic equations. Solving these algebraic equations yield concentration values at discrete positions in a predefined grid. Although the transport equation in discrete form is consistent with the differential equation the fixed grid size is responsible for discretization errors. This is especially envisaged in applications with steep concentration gradients, where the magnitude of the error may become comparable with that of the solution itself. Steep gradients arise in discharge applications modelled with continuous (or instantaneous) source terms. To resolve these gradients accurately the mesh must satisfy certain requirements. One requirement results from the possible presence of spurious oscillations (wiggles) that may overshadow the physical realistic solution. Another requirement restricts the grid size in order to guarantee relatively small truncation errors. Alternative methods have been developed to allow larger grid sizes. For instance, fractional step methods separates the advective and diffusive contributions into two different model equations - see YANENKO (1971). Nonlinear alternatives are found in the use of filter techniques that reduce the largest amplitudes in arising wiggles - see e.g., FORESTER (1977) or ENGQUIST *et al.* (1989) - or other nonlinear techniques that introduce artificial (numerical) diffusion - see e.g., BORIS and BOOK (1976). However, numerical diffusion affects accuracy. Still, in advection dominated flows the numerical diffusion must be negligible in comparison with physical diffusion.

A Lagrangian method is known as another alternative. In a Lagrangian frame of reference particle displacements are computed. Then the process is modelled similar to Brownian motion. Each particle displacement consists of a deterministic displacement and a stochastic displacement. This so-called random walk simulation is formulated such that the obtained approximation is consistent with the solution of equation (3.1). Such a method guarantees positive concentrations. Numerical diffusion is minimal and solutions are nonoscillating. The efficiency of a random walk method may be less competitive in comparison with Eulerian methods when dealing with big dispersing clouds since many particles need to be evaluated. Computer limitations may restrict the applicability of the model. It is stated that particle models are favourite in modelling point discharges. The Eulerian approach is preferred if (i) the physical process implies dynamic coupling with the flow simulation, or (ii) the concentration distribution remains sufficiently smooth.

The choice of a numerical method distinguishes far from near field applications. Far field transport is defined as transport of substance with sufficiently smooth concentration gradients which gives rise to the Eulerian approach. In near field applications the concentration distribution is characterized by steep gradients such that the particle method is adequate.

If the transport process implies turbulent mixing the advection-diffusion equation is obtained from the eddy diffusivity concept. The validity of this procedure is limited - see e.g., CORRSIN (1974) or BERKOWICZ and PRAHM (1980). For instance, at short times, when the diffusion distance is short compared with the Lagrangian integral scale T_L, the variance of a dispersing cloud is proportional to $E[\underline{u}\underline{u}]\, t^2 = O(t^2)$, showing the influence of the turbulence intensity, while at high times the variance of a dispersing cloud is proportional to $2E[\underline{u}\underline{u}]T_L\, t = O(t)$ - see LESIEUR (1990). Consequently, the advection-diffusion equation will not be expected to be successful at short times. In contrast to Eulerian methods the random walk method can be improved to account for this phenomenon. Then the movement of particles is determined by a deterministic and a stochastic velocity component. It results in a so-called random flight model - see e.g., HEEMINK (1990). This observation provides another argument to use particle models in point source applications.

The study of numerical simulation techniques for transport models employed in this thesis investigates the Eulerian and the Lagrangian viewpoint in solving the advection-diffusion equation. The flow data obtained with the flow simulation of chapter 2 is used as input for the transport simulation. Since the flow data is given in transformed coordinates, the transport model will also be derived considering transformed coordinates. Within the Eulerian frame of reference, to be discussed in section 3.3, the transport equation is transformed and written in discretized form. In chapter 4 the particle model will be discussed in detail. It considers transformed particle displacements. The transformation serves the implementation of the available flow data as input. Moreover, given the Eulerian mesh used during flow simulation concentrations are easily deduced from the Lagrangian simulation results. To overcome computer storage and computation time limitations a mixed particle/finite difference model is developed in chapter 5, in which the advantageous aspects of both alternatives are gathered. This model distinguishes between far field and near field transport. Far field transport is simulated with the finite difference solution technique of section 3.3, while near field transport is simulated with the particle model of chapter 4.

3.3 Discretization scheme for the advection-diffusion equation

Approximations are obtained from the discretized form of the transformed advection-diffusion equation. The transformed equation arises from the transformation relations given in the appendices. For an orthogonal transformation in the horizontal plane and the sigma transformation in the vertical plane, as introduced in (2.15), the following equation is obtained - see appendix C,

$$\frac{\partial(HC)}{\partial t} + \frac{1}{\sqrt{G}}\left(\frac{\partial}{\partial\xi}\left(H\sqrt{G_{\eta\eta}}u_\xi C\right) + \frac{\partial}{\partial\eta}\left(H\sqrt{G_{\xi\xi}}u_\eta C\right)\right) + \frac{\partial(\omega C)}{\partial\sigma} - \frac{1}{H}\frac{\partial}{\partial\sigma}\left(D_V\frac{\partial C}{\partial\sigma}\right) = H \times \mathrm{DP}(D_H) \tag{3.2}$$

The right-hand side of equation (3.2) denotes the contribution of the horizontal eddy diffusivity.

The grid under consideration coincides with the one introduced in section 2.5. The grid function used to approximate the concentration distribution defines the concentration value at the centre of each grid cell ($C_{m,n,k}$).

Due to stability requirements it is likely to apply an implicit time integration method. An unconditionally stable numerical scheme is also attractive from a computational point of view. Then, relatively large time increments can be used. A fully implicit scheme, however, couples the following unknowns at new time levels in a system of algebraic equations: (i) $C_{m-i,n,k}$, $C_{m,n,k}$, $C_{m+i,n,k}$, (ii) $C_{m,n-i,k}$, $C_{m,n,k}$, $C_{m,n+i,k}$, and (iii) $C_{m,n,k+i}$, $C_{m,n,k}$, $C_{m,n,k-i}$. The index $i = 1, 2, \dots, I$ depends on the order of consistency. The corresponding matrix system is hard to solve efficiently due to a large number of unknowns (>3 per equation). A tridiagonal matrix system (three unknowns) is usually efficiently solved. To construct a numerical method that only solves such tridiagonal matrix systems the algebraic equations are decoupled. For instance, to decouple the ξ - and η - direction an A.D.I. scheme is applied. This procedure is also recognized in the time splitting scheme of chapter 2 for flow simulation. To decouple the horizontal direction from the vertical direction an iteration procedure will be applied.

So, time integration of (3.2) is performed with a time splitting procedure. The spatial discretization is performed with a finite volume approach. Consequently, equation (3.2) is integrated in space over a grid cell (m,n,k) with volume $\Delta\xi\,\Delta\eta\,\Delta\sigma_k$. Then, as a result of Gauss' divergence theorem the advection contributions give rise to expressions in terms of fluxes through grid cell sides.

Subsequently, the time splitting scheme is summarized introducing the iteration index $[q]$. u_ξ, u_η, ω and H result from flow simulation.

Step 1, with $C^{[0]} = C^{l}$, $q = 1,..,Q$:

$$\Delta\sigma_k \frac{H^{l+½}_{m,n} C^{[q]}_{m,n,k} - H^{l}_{m,n} C^{l}_{m,n,k}}{½\Delta t} = \mathrm{AD}_1 \tag{3.3}$$

Step 2, with $C^{[0]} = C^{l+½}$, $q = 1,..,Q$:

$$\Delta\sigma_k \frac{H^{l+1}_{m,n} C^{[q]}_{m,n,k} - H^{l+½}_{m,n} C^{l+½}_{m,n,k}}{½\Delta t} = \mathrm{AD}_2 \tag{3.4}$$

in which

$$H_{m,n} = \zeta_{m,n} + \frac{1}{4}\left(d_{m+½,n+½} + d_{m-½,n+½} + d_{m+½,n-½} + d_{m-½,n-½}\right) \tag{3.5}$$

The spatial difference operators AD_1 and AD_2 contain the advective and diffusive contributions. Skipping the detailed formulas these operators include the following:

- Vertical advection and diffusion are treated according the Crank-Nicolson method (time averaging) - see LAMBERT (1973).

- Horizontal advection is treated with the (semi-implicit) A.D.I. scheme, given as an iteration procedure. In ξ – direction the advection approximation becomes

$$\Delta\sigma_k \frac{\partial}{\partial\xi}\left(H\sqrt{G_{\eta\eta}}\,u_\xi C\right) = \frac{T_{m+½} - T_{m-½}}{\Delta\xi} \tag{3.6}$$

with flux $T_{m+½}$ given by

$$T_{m+½} = \Delta\sigma_k\, H_{m+½,n} \left.\sqrt{G_{\eta\eta}}\right|_{m+½,n} \Delta\eta\, \left(u_\xi\right)_{m+½,n,k} C_{m+½,n,k} \tag{3.7}$$

Since the concentration values are only defined at the centre of each grid cell, the concentration at the u_ξ – location must be approximated by inter- or extrapolation. In step 1 an upwind scheme is used (implicit with iteration index $[q-1]$), while in step 2 a central difference scheme is used (explicit). This procedure is in accordance with advection scheme 5 of appendix D, here applied to concentration. This scheme provides adequate propagation properties with minimal numerical diffusion. In coming chapters this scheme will be used in numerical experiments. To illustrate the phenomenon of artificial diffusion the scheme will be compared with first order upwind differences analogue to scheme 1 of appendix D.

It is noted that due to the A.D.I. method the advection in η – direction interchanges the above described procedures for step 1 and step 2. That is, an explicit central difference scheme in step 1 and an implicit (iterative) upwind scheme in step 2.

- The horizontal diffusion term requires a special treatment. In Cartesian coordinates

$$\mathrm{DP}\left(D_H\right) = \frac{\partial}{\partial x}\left(D_H \frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(D_H \frac{\partial C}{\partial y}\right) \tag{3.8}$$

The complete transformed transport equation is given in appendix C. The amount of terms becomes disorderly. Each term has to be discretized to yield consistent approximations. However, due to truncation errors artificial diffusion is introduced. If only the horizontal transformation is considered the following expression arises

$$\mathrm{DP}\left(D_H\right) = \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\left(\frac{\partial}{\partial \xi}\left(D_H \frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}} \frac{\partial C}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(D_H \frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}} \frac{\partial C}{\partial \eta}\right)\right) \tag{3.9}$$

This expression is incomplete in a sense that it does not account for the σ – transformation. Implementation of this expression yields an apparent flux in vertical direction. This phenomenon is described in KESTER and UITTENBOGAARD (1990) and appendix C. To obtain accurate approximations with minimal vertical artificial diffusion the authors mentioned above propose a different numerical treatment. $\mathrm{DP}\left(\nu_t^H\right)$ is treated explicitly. The coordinates ξ and η, as they arise in equation (3.9), are regarded as strictly horizontal coordinates. The horizontal diffusion term gives rise to horizontal flux contributions through vertical grid cell sides. These contributions are evaluated with respect to each grid cell. This treatment yield accurate approximations. The explicit treatment affects the property of unconditional stability. The numerical experiments of section 4.4.2 will include an experiment with the incomplete transformation to show the consequences.

- Finally, a comment is made about the computation of the Jacobian of the horizontal transformation, $\sqrt{G}$, as it arises in equation (3.2). $\sqrt{G}$ expresses the local area of a grid cell projected on a horizontal plane. In case of orthogonality $\sqrt{G} = \sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}$. So, $\sqrt{G}$ is obviously approximated with either the exact area of the projected grid cell or with

$$\sqrt{G}\Big|_{m,n} = \frac{1}{4}\left(\sqrt{G_{\eta\eta}}\Big|_{m-\frac{1}{2},n} + \sqrt{G_{\eta\eta}}\Big|_{m+\frac{1}{2},n}\right) \times \left(\sqrt{G_{\xi\xi}}\Big|_{m,n-\frac{1}{2}} + \sqrt{G_{\xi\xi}}\Big|_{m,n+\frac{1}{2}}\right) \tag{3.10}$$

In perfect orthogonal grids these two alternatives coincide. However, in almost orthogonal grids (and nonorthogonal grids) the method that uses the projected area is favourite. This statement anticipates the results of section 4.4.1 that describes a Lagrangian advection algorithm.

Although the propagation properties of the advection scheme are adequate an adverse aspect is that the scheme does not guarantee positive approximations. Especially in regions where steep concentration gradients are present negative concentrations might be generated. An accepted procedure is to apply a filter technique to suppress the amplitude of the oscillations that are responsible for negative contributions. The filter technique used here is known as the Forester filter as described in FORESTER (1977). It is reminded that the finite difference method will mainly be applied in far field simulations. In that case negative values are unnoticed. For that reason a detailed description of the Forester filter will not be given. In chapter 5 finite difference methods are compared with other methods in a one-dimensional environment. In this comparison study the one-dimensional Forester filter will be outlined.

Chapter 4

Particle methods

4.1 Introduction

For a three-dimensional transport model flow velocities, water levels and mixing coefficients must be known a priori. These quantities are obtained by a three-dimensional hydrodynamic model. The hydrodynamic model yields flow information at discrete locations with respect to a predefined Eulerian grid. A question that arises is how to use this information correctly in a transport model. With correctness it is meant that (i) physical assumptions made in the hydrodynamic model are not violated by the transport model, and (ii) the transport equations and boundary conditions are simulated accurately.

A particle model is constructed such that a consistent approximation of the solution of the advection-diffusion equation is obtained. The following issues are discussed seperately:

- computation of particle displacements relative to transformed Eulerian grids,
- description of an accurate advection algorithm,
- numerical simulation of the dispersive step,
- conversion algorithms from (discrete) particle information to a (continuous) concentration distribution,
- boundary treatment, and
- implementation of sources.

The particle model is described in section 4.2. Computational issues regarding the model's implementation are discussed in section 4.3. Section 4.4 contains numerical experiments.

4.2 Theoretical aspects

Durbin states *"It is natural to look to the theory of stochastic differential equations for insight inside the phenomenon of the turbulent transport"* - DURBIN (1983). The resemblance between the algorithm and the underlying physical processes motivates its development through intuitive reasoning. In section 4.2.1 some basic concepts of particle models are summarized in connection with the assumed mathematical-physical description of transport processes as given by equation (3.1) of section 3.2. Boundary conditions simulated in particle models are discussed in section 4.2.2. For an efficient use of discrete flow data in particle models the simulation will be performed in transformed coordinates. The description of such a particle model will be given in section 4.2.3.

4.2.1 Stochastic modelling; particle models

Modelling diffusion with particle models requires the statistical theory of Brownian motion. An introduction to stochastic processes is necessary. The mathematical description of the transport process will be written in terms of a stochastic differential equation. The difference between the stochastic and deterministic calculus will be outlined to provide some useful tools for the implementation of the particle method on a digital computer.

- Particle displacements

When particles are moving in a fluid, the molecules of the surrounding fluid will collide with the particles causing random displacements of the particles. The migration and fluctuation in the displacement of particles are described with a so-called Langevin equation, written as a (Markovian) stochastic differential equation, i=1,2,3 - see SCHUSS (1980)

$$\frac{d}{dt}(X_i) = \underbrace{h_i(\underline{X},t)}_{\text{deterministic}} + \underbrace{\sum_{j=1}^{3} g_{ij}(\underline{X},t)\Gamma_j(t)}_{\text{stochastic}} \tag{4.1}$$

with $\underline{X} = (X_1,X_2,X_3)^T$ the position of a particle, $\underline{h}$ the drift vector and g the noise tensor. The superscript T indicates the transpose of the vector. The stochastic Langevin force Γ_j is assumed to have zero mean and a Gaussian distribution with δ-correlation, i.e., Gaussian white noise - see KAMPEN (1981). This so-called white noise process appears to be a useful mathematical idealization for describing random influences. It is noted that in the absence of the stochastic contribution equation (4.1) resembles a streamline equation. In that case the particle model resembles to the method of characteristics used to solve hyperbolic equations. The Markov property implies that if the present state is known, the future will be independent of the past. Thus, the process is completely determined by the initial condition given at $t = t_0$. The process expressed in (4.1) can also be presented as a fluctuation equation written as, i=1,2,3

$$dX_i = h_i(\underline{X},t)dt + \sum_{j=1}^{3} g_{ij}(\underline{X},t)\; dW_j(t) \tag{4.2}$$

In (4.2) the Wiener increment $dW_j(t) = \Gamma_j(t)dt$ is used, which expresses the stochastic influence of the process. The corresponding Wiener process is defined as

$$W_j(t) = \int_{t_0}^{t} dW_j(t) \tag{4.3}$$

and is called the continuous drunkard's walk, since it describes the position of a drunkard after time $t - t_0$ who has been undergoing successive independent random steps in x_j-direction - see DURBIN (1983). It is also referred to as the Brownian motion process - see JAZWINSKI (1970). It represents a Gaussian process with mean zero and variance $t - t_0$. The increments are also Gaussian distributed, stochastically independent with mean zero and variance dt - see PAPOULIS (1965)

$$\text{mean:} \quad E\left[dW_j(t)\right] = 0 \tag{4.4a}$$

$$\text{covariance:} \quad E\left[dW_i(t_1)\, dW_j(t_2)\right] = \begin{cases} 0 & i \neq j \vee t_1 \neq t_2 \\ dt & i = j \wedge t_1 = t_2 \end{cases} \tag{4.4b}$$

It is reminded that the position of a particle is assumed to evolve in a Markovian manner with independent increments. The assumption about uncorrelated increments in time is justified if the root mean square displacement is proportional to $\sqrt{t - t_0}$, which characterizes Brownian motion. However, in homogeneous isotropic turbulent flows, this is only true if $t - t_0 \gg T_L$, with T_L the Lagrangian integral time scale - see LESIEUR (1990). T_L is a measure of the period that a particle takes to lose memory of its initial turbulent velocity - see FISCHER *et al.* (1979). Consequently, the discussion here is restricted to relatively long time periods, i.e., $t - t_0 \gg T_L$[†]. Moreover, for such relatively long time periods the assumption about the validity of the advection-diffusion equation is accepted.

The solution of the stochastic differential equation determines the transition probability density function of the random process $\underline{X}(t)$. With the assumption of equation (4.1) being

---

[†] To account for inhomogeneities, unsteadiness or non-Gaussianity in the turbulent velocity distribution and to become applicable in cases where the travel time of particles is much less than the Lagrangian integral time scale, other kind of random walk models are developed - see THOMSON (1987). Then the evolution of $(\underline{X},\underline{V})$, assumed Markovian, is described by the stochastic differential equation

$$\begin{cases} dV_i = a_i(\underline{X},\underline{V},t)dt + \sum_{j=1}^{3} b_{ij}(\underline{X},\underline{V},t)dW_j(t) \\ dX_i = V_i dt \end{cases}$$

An illustration is given in HEEMINK (1990), in which a (two-dimensional) shallow water flow is considered. His paper describes a particle method that converges for long diffusion times, $t - t_0 \gg T_L$, to the advection-diffusion equation (i.e., the eddy diffusivity concept). Short diffusion time behaviour is controlled by an exponential Lagrangian autocorrelation function. Consequently, initially, the root mean square displacement increases linearly with the time of travel, which corresponds with the statistical theory of turbulence - see MONIN and YAGLOM (1975).

Markovian the evolution equation of the corresponding probability density function, p, satisfies the so-called Kramer-Moyal expansion - see RISKEN (1984). In one dimension this expansion is written as

$$\frac{\partial p}{\partial t} = \sum_{n=1}^{\infty} \frac{1}{n!}\left(-\frac{\partial}{\partial x}\right)^{n}\left[\tilde{a}^{(n)}p\right] \tag{4.5}$$

with $\tilde{a}^{(n)}$ the rate of increase of the n^{th} moment of step dX. It matches the advection-diffusion equation only if the first two terms of (4.5) are taken into account. Truncation of (4.5) up to second order yields the so-called Fokker-Planck equation. Pawula argues that truncation after third order implies the existence of negative solutions. Consequently, either the advection-diffusion equation or the Kramer-Moyal expansion with infinite number of terms must be considered - see PAWULA (1967). The latter alternative is inefficient in a sense that infinite number of terms requires infinite number of boundary conditions. It is reminded that the transport process is assumed to be governed correctly by the advection-diffusion equation. So, a correct analogue with the random walk is obtained if higher order derivatives ($n \geq 3$) in the Kramer-Moyal expansion vanish. Further comments are reported in UFFINK (1990) and CORRSIN (1974).

- Simulation of the concentration distribution

To simulate the mass distribution that corresponds with the advection-diffusion process at least two alternatives exist to proceed:

(i) Derive the evolution equation of the probability density function which defines the probability to find a particle in an infinitesimal interval at time t per unit volume. The time evolution of the conditional probability density function of the particle positions, given the initial condition, is expressed by the Fokker-Planck equation. The solution of this equation is associated with the concentration distribution to be simulated. The solution method solves equation (4.5) directly, assuming $\tilde{a}^{(n)} = 0$ for $n \geq 3$.

(ii) For each individual particle solve the fluctuation equation numerically. The simulation is performed by injection of a finite number of particles, where the initial position of each particle is determined by the initial concentration distribution. Particle trajectories are constructed as a time-evolving simulation obeying equation (4.2). The outcome is a list of particle positions and corresponding mass elements from which a mass distribution can be obtained. Obviously, the concentration distribution is associated with this mass distribution. Since each particle displacement consists of a random contribution the method is often referred to as the random walk method.

In particle models the second alternative is used for simulation purposes. The first alternative will be used to define the drift and noise components such that consistency with the advection-diffusion equation is obtained.

- Itô versus Stratonovich

In order to compute particle displacements equation (4.1) has to be integrated. If (4.1) is expressing a deterministic differential equation time integration will be straightforward. Then the interval (t_0,t) is divided into N sub-intervals (t_n,t_{n+1}) of length Δt with $t_n = t_0 + n\Delta t$, $t_N = t$ and $N\Delta t = t - t_0$. The solution is obtained by successive summation of all n integrands evaluated at $t_n + \theta\Delta t$, $0 \le \theta \le 1$, and multiplied by Δt. If the sum converges to a certain limit irrespective of θ (i.e., irrespective of the evaluating point inside the interval) the integrand is called Riemann integrable. However, equation (4.1) represents a stochastic differential equation (instead of a deterministic differential equation). Here the limit that determines the statistical properties of the process depends on the choice of θ and equation (4.1) becomes meaningful if the integration method (θ) is prescribed. The choice of this θ leads to the Itô-Stratonovich dilemma - see KAMPEN (1981). The integration according to Itô defines $\theta = 0$, while the integration according to Stratonovich sets $\theta = 0.5$. The difference between the Itô and the Stratonovich procedure can be understood by looking at equation (4.1). Since the strength of the fluctuation in the position of the particle depends on the particle's trajectory, one has to decide whether the jump depends on the position before the jump (Itô), after the jump or, for instance, the mean of both values (Stratonovich) - see DURBIN (1983).

The integration rule determines the properties of a stochastic process. Since the Itô interpretation yields an explicit numerical method ($\theta = 0$), the Itô calculus has been chosen to compute the particle displacements. Unfortunately, if the stochastic differential equation is nonlinear, the Itô calculus, in particular the stochastic integration, differs from the classical (Riemann) calculus. It yields a Fokker-Planck equation that depends on the chosen integration rule.

- The Fokker-Planck equation

The three-dimensional Fokker-Planck equation is given in FELLER (1971)

$$\frac{\partial p}{\partial t} = -\sum_{i=1}^{3} \frac{\partial}{\partial x_i}(A_i p) + \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial^2}{\partial x_i \partial x_j}(B_{ij} p) \tag{4.6}$$

with $p = p(\underline{X},t\,|\,\underline{X}(t_0),t_0)$ the transition probability density function and $\underline{X}(t_0)$ the initial condition at time $t = t_0$. A_i represents the mean displacement (drift) in x_i - direction of the particle per unit time and the coefficients B_{ij} represent the increase of the (co-)variance per unit time. The integration rule determines the relation between $\underline{A}$ and B on the one hand and $\underline{h}$ and g on the other. Using the Itô integration rule, this relation becomes - see UFFINK (1990)

$$\begin{cases} A_i = h_i \\ B_{ij} = \tfrac{1}{2}\sum_{k=1}^{3} g_{ik}g_{jk} \end{cases} \tag{4.7}$$

With relation (4.7), equation (4.6) should be referred to as the Itô-Fokker-Planck equation. It is noted that the Stratonovich integration rule leads to different A_i and B_{ij} in terms of h_i and g_{ij}. Both rules are correct in the sense that both can be used for simulation purposes - see KLOEDEN and PLATEN (1992).

- Consistency with the advection-diffusion equation

By matching the Fokker-Planck equation with the advection-diffusion equation the consistency between the particle model and the advection-diffusion equation is guaranteed. This procedure yields

$$\begin{cases} A_i = u_i + \sum_{j=1}^{3} \dfrac{\partial D_{ij}}{\partial x_j} \\ B_{ij} = D_{ij} \\ p = C \end{cases} \tag{4.8}$$

Equation (4.7) relates the drift and noise components with the local flow velocity and the diffusion coefficients

$$\begin{cases} h_i = u_i + \sum_{j=1}^{3} \dfrac{\partial D_{ij}}{\partial x_j} \\ \tfrac{1}{2}\sum_{k=1}^{3} g_{ik}g_{jk} = D_{ij} \end{cases} \tag{4.9}$$

The drift component h_i consists of a contribution due to the local flow velocity and a contribution due to the space-varying diffusivity. The latter is referred to as the noise-induced drift component. The hydrodynamic flow model of chapter 2 gives rise to a simplified diffusion tensor. It is isotropic in the horizontal plane. The off-diagonal elements of the diffusion tensor are set to zero, while the diagonal elements equal $D_{11} = D_{22} = D_H$ and

$D_{33} = D_V$. For this case the drift and noise components will be given explicitly. Now, equation (4.9) reduces to

$$\begin{cases} h_1 = u_1 + \dfrac{\partial D_H}{\partial x_1} \\ h_2 = u_2 + \dfrac{\partial D_H}{\partial x_2} \\ h_3 = u_3 + \dfrac{\partial D_V}{\partial x_3} \end{cases} \qquad g = \begin{pmatrix} \sqrt{2D_H} & 0 & 0 \\ 0 & \sqrt{2D_H} & 0 \\ 0 & 0 & \sqrt{2D_V} \end{pmatrix} \tag{4.10}$$

Equation (4.10) will be applied if the Eulerian mesh with flow data is given in Cartesian coordinates.

4.2.2 Stochastic modelling; boundary conditions

For particle models particle tracks are computed. The origin of a particle track is determined by the initial concentration distribution. The simulation marches forward in time. After time t the positions of particles determine the concentration distribution. This approach simulates a transport process written as an initial value problem. Section 4.2.1 does not give a definite answer in applying boundary conditions or point sources in terms of particle distributions. This section will concentrate on these issues. For that purpose the stochastic representation of the solution will be discussed. Within a stochastic framework the concentration is expressed in terms of an expectational value - see e.g., DYNKIN and JUSCHKEWITSCH (1969), FRIEDMAN (1975) or ØKSENDAL (1992). This expectational value is then obtained by running a Monte-Carlo simulation - see e.g., SABELFELD (1992) or RUBINSTEIN (1981). The difference between a Monte-Carlo simulation and the particle model of section 4.2.1 is that the Monte-Carlo simulation marches backward in time. It will be shown that such a simulation may apply for boundary conditions. The concepts are transformed such that the particle model, marching forward in time, also applies for conditions on the boundary.

- Monte Carlo samples

The stochastic approach for the analysis of transport formulates the mean concentration $\overline{C}$ at the point $(\underline{x},t)$ in terms of an ensemble of random trajectories of particles, starting at some point $\underline{x}_0$, by taking into account the initial concentration C_I and the probability $P(\underline{x},t\,|\,\underline{x}_0)$ that a particle is observed at position $\underline{x}$ at time t started at $\underline{x}_0$. This formulation is valid for an initial value diffusion problem in an unbounded domain. Then the mean concentration is expressed by the formulae - see DURBIN (1983)

$$\overline{C}(\underline{x},t) = \int\int\int P(\underline{x},t\,|\,\underline{x}_0)C_I(\underline{x}_0)\; d\underline{x}_0 \tag{4.11}$$

It is to be noted that the initial position $\underline{x}_0$ is random, described by the transition probability P. Equivalently, equation (4.11) is written as

$$C(\underline{x},t) = E^{\underline{x},t}\left[C_I(\underline{X}(0))\right] \tag{4.12}$$

where $E^{\underline{x},t}$ denotes the expectation with respect to the (diffusion) process $\underline{X}(t)$, started at position $\underline{x}$ at time t. The evaluation of the expectational value is obtained by running a Monte Carlo simulation. In that case particles are released at position $\underline{x}$ at time t. The simulation marches backward in time. The position of a particle at time $t = 0$ is determined by the inverse flow direction and the diffusion coefficients. (4.12) is approximated with

$$E^{\underline{x},t}\left[C_I(\underline{X}(0))\right] = \frac{1}{\mathrm{n_p}}\sum_{p=1}^{\mathrm{n_p}} C_I\left(\underline{X}^p(0)\right) \tag{4.13}$$

with $\mathrm{n_p}$ the number of released particles - see DYNKIN and JUSCHKEWITSCH (1969). The accuracy of such an initial value diffusion simulation in an unbounded domain is determined by the number of trials to estimate an ensemble average. The difference between a backward and a forward simulation is illustrated in figure (4.1).

- Stochastic representation of the concentration distribution

The extension to diffusion processes including production/source terms and boundary conditions, is obtained by looking at the stochastic representation of initial-boundary value diffusion problems. Here an advection-diffusion problem is considered with exponential decay and the presence of source terms. For convenience of notation the flow region is denoted with Ω, its boundary with $\partial\Omega$. The corresponding equation becomes

$$\begin{cases} \dfrac{\partial C}{\partial t} + \mathrm{L}C = S(\underline{x},t) & (\underline{x},t) \in \Omega \times [0,T) \\ C(\underline{x},T) = C_I(\underline{x}) & \underline{x} \in \Omega \\ C(\underline{r},t) = b_D(\underline{r},t) & (\underline{r},t) \in \partial\Omega \times [0,T) \end{cases} \tag{4.14a}$$

S represents the source terms and b_D denotes the Dirichlet boundary value. The operator L expresses the advection, dispersion and exponential decay of substance. It is written as

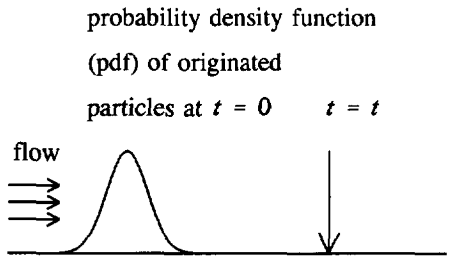


Figure (4.1a): The backward approach (Monte Carlo simulation).

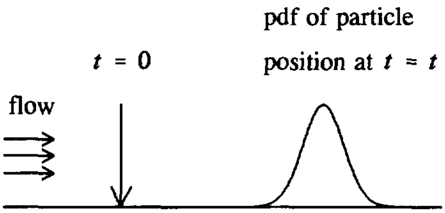


Figure (4.1b): The forward approach (particle model).

$$L = \sum_{i=1}^{3} -V_i \frac{\partial}{\partial x_i} + \sum_{i=1}^{3}\sum_{j=1}^{3} D_{ij} \frac{\partial^2}{\partial x_i \partial x_j} - \gamma \tag{4.14b}$$

with V_i the local velocity of the substance-water mixture in x_i-direction. γ denotes the decay coefficient and is assumed to be positive, i.e., $\gamma \geq 0$.

Equation (4.14) is recognized as a "backward" equation by (i) the presence of a " + "-sign at the left-hand side of equation (4.14a), and (ii) the initial condition given at time $t = T$, instead of at time $t = 0$. The stochastic representation of the solution of equation (4.14) becomes - see FRIEDMAN (1975)

$$\begin{aligned} C(x,t) &= E^{x,t}\left[C_I(X(T)) \exp\left(\int_t^T \gamma(X(s),s)\, ds \right) \chi_{\tau=T} \right] \\ &+ E^{x,t}\left[b_D(X(\tau),\tau) \exp\left(\int_t^\tau \gamma(X(s),s)\, ds \right) \chi_{\tau<T} \right] \\ &- E^{x,t}\left[\int_t^\tau S(X(s),s) \exp\left(\int_t^s \gamma(X(\lambda),\lambda)\, d\lambda \right) ds \right] \end{aligned} \tag{4.15}$$

τ represents the exit time, the time needed to reach the boundary. If a particle stays inside the domain during execution, τ is set to the simulation time T. The indicator function χ is defined as

$$\chi_{\{\text{condition}\}} = \begin{cases} 1 & \text{if condition} \\ 0 & \text{if not condition} \end{cases} \tag{4.16}$$

Similar to (4.13), the formulation in terms of expectational values, as in equation (4.15),

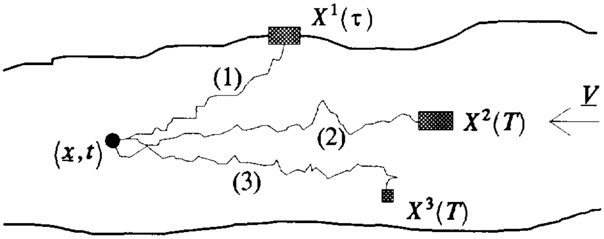


Figure (4.2): Illustration of Monte Carlo simulation to estimate the concentration at $(\underline{x},t)$.

(1) particle track accounts for the boundary condition,

(2) particle track accounts for a point source,

(3) particle track accounts for the initial condition.

gives rise to a Monte Carlo simulation, where information is transferred through the interior. Particles are released at position $\underline{x}$ at time t. After a (reversed) simulation time $T - t$, the end points of the particle tracks are evaluated with respect to (i) the initial condition, (ii) the (Dirichlet) boundary condition, and (iii) the presence of point sources - see figure (4.2). Decay is determined along each path.

An analogy with a forward simulation is obtained if in the particle model at initial state particle tracks start at (i) positions where $C_I(\underline{x}) > 0$, (ii) boundaries, and (iii) point source locations. The remainder of this section discusses the analogy in more detail. Besides the prescription of initial particle positions, the mass of each particle must be addressed.

- Particle attributes

To simulate the concentration distribution each particle represents a certain amount of mass. So, a mass element, say μ_p, is assigned to each particle. This mass element is called an attribute of the particle. Due to exponential decay the evolution in time of the mass of a particle is given by

$$\mu_p(t) = \mu_p(0) \exp\left[-\int_0^t \gamma(\underline{X}(s),s)\, ds\right] \tag{4.17}$$

Particle models are easily extended to simulate the dispersion of many different noninteracting constituents. Then, in addition, a specific name is assigned to each particle. It is noted that also other quantities can be assigned as particle attributes. For instance, GHONIEM and SHERMAN (1985) discuss the use of the concentration gradient as a possible particle attribute (gradient random walk).

- Particle positions initiated by the initial concentration distribution and point sources

As in section 4.2.1, particles are released inside the flow region. At initial state the number of particles and their mass elements are easily determined by integration of the initial concentration distribution over a unit volume. In numerical simulations this unit volume corresponds with a grid cell of the Eulerian mesh. The numerical implementation is examined in section 4.3.3. Analogous to this procedure, point sources determine particle injection during simulation.

- Boundary conditions in particle models

Equation (4.15) indicates that in the backward approach during simulation, information (here, mass elements) is transferred from the boundary to the interior. In the forward approach particles, originated by the initial concentration distribution and point sources, are withdrawn as they reach the boundary. As seen from equation (4.15) a complete particle withdrawal yields a zero concentration boundary. To account for boundary conditions other than a zero concentration boundary, additional particles are introduced at the boundary. The number of particles that has to be injected needs to be specified. This number is determined by the expected mass flux through the boundary per time unit dt. This flux is defined as

$$F_{\partial\Omega}(t) = \int_{t}^{t+dt} \left[\int_{\partial\Omega} \underline{q}\cdot\underline{n}\, dA \right] ds \tag{4.18}$$

with $\underline{n}$ the normal vector to the boundary $\partial\Omega$ pointing outwards and $\underline{q}$ the mass flux where each component (i=1,2,3) is defined as

$$q_i = V_i C - \sum_{j=1}^{3} D_{ij} \frac{\partial C}{\partial x_j} \tag{4.19}$$

Equation (4.18) is evaluated using the imposed boundary condition. For instance, if $\partial C/\partial x_j$ is prescribed the procedure requires the estimation of C near the boundary to estimate $F_{\partial\Omega}$. $F_{\partial\Omega}$ determines the intensity of a point source positioned at the boundary. It is noted that this procedure can be applied for inflow as well as outflow boundaries.

The procedure is illustrated as follows. Suppose that $F_{\partial\Omega}$ resembles an outflow of 100 particles. If the simulation of particles displacements inside the flow region results in a particle withdrawal of 150 particles, the intensity of the source at the boundary becomes simply 50 particles.

It is noted that for zero flux boundary conditions the procedure is easily simplified. A zero flux boundary condition means that transfer of mass through such a boundary is prohibited. This is realized by reflection of particles. None of the simulated particles will be withdrawn. The detailed numerical implementation of boundary conditions is discussed in section 4.3.6.

4.2.3 The random walk in transformed coordinates

The particle model requires flow velocity and diffusion coefficients as input. These physical quantities are obtained from a numerical hydrodynamic model. Such a model could be based upon curvilinear grids for a more accurate schematization of complicated geometries. It is convenient to compute the particle displacements in the corresponding computational space. For that purpose the fluctuation equation will be transformed accordingly. To obtain the transformed fluctuation equation the stochastic chain rule must be considered. This stochastic chain rule, the "Itô formula", will be discussed first.

- The stochastic chain rule

The stochastic chain rule is derived by looking at the following one-dimensional transformation

$$\xi = \xi(x,t) \tag{4.20}$$

The chain rule relates $d\xi$ with dx and dt. The corresponding expression is found by looking at the first few terms in the Taylor expansion for ξ in

$$\Delta\xi = \xi(x + \Delta x, t + \Delta t) - \xi(x,t) \tag{4.21}$$

in which

$$x + \Delta x = x(t + \Delta t) \tag{4.22}$$

and similar to equation (4.2)

$$\Delta x = h\Delta t + g\Delta W \tag{4.23}$$

Inserting a Taylor series expansion in equation (4.21) yields

$$\begin{aligned}\Delta\xi &= \frac{\partial\xi}{\partial t}\Delta t + \frac{\partial\xi}{\partial x}\Delta x + \tfrac{1}{2}\Delta x\frac{\partial^2\xi}{\partial x^2}\Delta x + \dots \\ &= \left(\frac{\partial\xi}{\partial t} + \frac{\partial\xi}{\partial x}h\right)\Delta t + \frac{\partial\xi}{\partial x}g\Delta W + \tfrac{1}{2}g\frac{\partial^2\xi}{\partial x^2}g\Delta W^2 + \dots\end{aligned} \tag{4.24}$$

Now it is reminded that $E[\Delta W^2] = \Delta t$, such that the $O(\Delta W^2)$-terms in (4.24) have to be taken into account[†]. Finally, the stochastic chain rule becomes

$$d\xi = \left(\frac{\partial\xi}{\partial t} + \frac{\partial\xi}{\partial x}h + \tfrac{1}{2}g\frac{\partial^2\xi}{\partial x^2}g\right)dt + \frac{\partial\xi}{\partial x}g\,dW \tag{4.25}$$

Formally, the equality given in (4.25) is interpreted in the mean-square sense. A rigorous derivation of the Itô formula is given in e.g., PUGACHEV and SINITSYN (1987) or KLOEDEN and PLATEN (1992). It is noted that the difference with the deterministic analogue is expressed by $\tfrac{1}{2}g(\partial^2\xi/\partial x^2)g\,dt$.

In three dimensions the derivation is similar as in one dimension. With $\underline{\xi} = \underline{\xi}(\underline{x},t)$ and $\underline{\xi} = (\xi^1,\xi^2,\xi^3)^T$ the fluctuation equation becomes

$$d\xi^i = \tilde{h}_i(\underline{x},t)dt + \sum_{j=1}^{3}\tilde{g}_{ij}(\underline{x},t)dW_j \tag{4.26}$$

where

$$\left\{\begin{aligned}\tilde{h}_i &= \left(\frac{\partial\xi^i}{\partial t}\right)_{\underline{x}} + \sum_{j=1}^{3}\frac{\partial\xi^i}{\partial x_j}h_j + \tfrac{1}{2}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3}\frac{\partial^2\xi^i}{\partial x_j\partial x_k}g_{jl}g_{kl} \\ \tilde{g}_{ij} &= \sum_{k=1}^{3}\frac{\partial\xi^i}{\partial x_k}g_{kj}\end{aligned}\right. \tag{4.27}$$

The tilde refers to the transformed components. In the expression for $\tilde{h}_i$, the subscript $\underline{x}$ indicates the variable which is held constant.

- Comparison with the deterministic case

In the one-dimensional deterministic case the increment of a particle position becomes $\Delta x = u\Delta t$. Then the transformation, given in (4.20), leads to

---

[†] For nongaussian processes even higher order terms, $E[\Delta W^n]$, $n \geq 2$, have to be taken into account - see PAOLA and FALSONE (1993).

$$\Delta\xi = \underbrace{\frac{\partial\xi}{\partial t}\Delta t + \frac{\partial\xi}{\partial x}\Delta x}_{\text{influence of contravariant velocity}} + \underbrace{\tfrac{1}{2}\frac{\partial^2\xi}{\partial x^2}\Delta x^2}_{\substack{\text{curvature term}\\ \text{(grid stretching)}}} + \ldots \tag{4.28}$$

The influence of the contravariant velocity component is recognized using equation (A.14) and (A.6) of appendix A

$$\frac{\partial\xi}{\partial t}\Delta t + \frac{\partial\xi}{\partial x}\Delta x = \left[\frac{\partial\xi}{\partial t} + \frac{\partial\xi}{\partial x}u\right]\Delta t = \frac{\partial\xi}{\partial x}\left(u - \frac{\partial x}{\partial t}\right)\Delta t = U\Delta t \tag{4.29}$$

The order of magnitude of the curvature term in (4.28) equals $O(\Delta t^2)$. In the stochastic case described above the curvature term, with respect to the stochastic displacement, appears to be $O(\Delta W^2) = O(\Delta t)$. So, omission of the curvature terms in (4.27) implies that the error is of order $O(\Delta t)$, while the displacement itself is already $O(\Delta t)$. Consequently, from a consistency point of view these curvature terms have to be included. The numerical consequences will be discussed in section 4.3.2.

- The Fokker-Planck equation in transformed coordinates

The transformed drift components, $\tilde{h}_i$, and the transformed noise coefficients, $\tilde{g}_{ij}$, can also be obtained by looking at the Fokker-Planck equation in transformed coordinates. Obviously, this alternative derivation must lead to the same expressions. The necessary transformation relations are summarized in appendix A. Substitution of these relations into equation (4.6) gives, with J the Jacobian of the transformation

$$\begin{aligned}\frac{1}{J}\left(\frac{\partial(Jp)}{\partial t}\right)_{\boldsymbol{\xi}} = & -\frac{1}{J}\sum_{i=1}^{3}\frac{\partial}{\partial\xi^i}\left[\left(\sum_{j=1}^{3}\frac{\partial\xi^i}{\partial x_j}A_j + \left(\frac{\partial\xi^i}{\partial t}\right)_{\boldsymbol{x}}\right)Jp\right]\\ & + \frac{1}{J}\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\sum_{k=1}^{3}\sum_{l=1}^{3}\frac{\partial}{\partial\xi^k}\left[\frac{\partial\xi^k}{\partial x_j}\frac{\partial}{\partial\xi^l}\left(\frac{\partial\xi^l}{\partial x_i}B_{ij}Jp\right)\right]\right)\end{aligned} \tag{4.30}$$

By rearranging terms in the diffusive part, like

$$\begin{aligned}\sum_{k=1}^{3}\sum_{l=1}^{3}\frac{\partial}{\partial\xi^k}\left[\frac{\partial\xi^k}{\partial x_j}\frac{\partial}{\partial\xi^l}\left(\frac{\partial\xi^l}{\partial x_i}B_{ij}Jp\right)\right] = & \sum_{k=1}^{3}\sum_{l=1}^{3}\frac{\partial^2}{\partial\xi^k\partial\xi^l}\left(\frac{\partial\xi^k}{\partial x_j}\frac{\partial\xi^l}{\partial x_i}B_{ij}Jp\right)\\ & - \sum_{k=1}^{3}\frac{\partial}{\partial\xi^k}\left(\frac{\partial^2\xi^k}{\partial x_i\partial x_j}B_{ij}Jp\right)\end{aligned} \tag{4.31}$$

the transformed Fokker-Planck equation becomes

$$\frac{\partial \tilde{p}}{\partial t} = -\sum_{i=1}^{3} \frac{\partial}{\partial \xi^i}(\tilde{A}_i \tilde{p}) + \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial^2}{\partial \xi^i \partial \xi^j}(\tilde{B}_{ij} \tilde{p}) \tag{4.32}$$

in which

$$\begin{cases} \tilde{p} = Jp \\ \tilde{A}_i = \left(\dfrac{\partial \xi^i}{\partial t}\right)_{\underline{x}} + \displaystyle\sum_{j=1}^{3} \frac{\partial \xi^i}{\partial x_j} A_j + \sum_{j=1}^{3} \sum_{k=1}^{3} \frac{\partial^2 \xi^i}{\partial x_j \partial x_k} B_{jk} \\ \tilde{B}_{ij} = \displaystyle\sum_{k=1}^{3} \sum_{l=1}^{3} \frac{\partial \xi^i}{\partial x_k} \frac{\partial \xi^j}{\partial x_l} B_{kl} \end{cases} \tag{4.33}$$

The first equality of equation (4.33) expresses that volume elements are transformed according to the Jacobian J. It will now be shown that equation (4.26) corresponds with the transformed Fokker-Planck equation (4.32). For that purpose, assume that the following Itô stochastic differential equation corresponds with (4.32)

$$d\xi^i = \bar{h}_i(\underline{x},t)dt + \sum_{j=1}^{3} \bar{g}_{ij}(\underline{x},t)\, dW_j(t) \tag{4.34}$$

Similar as in (4.7)

$$\begin{cases} \tilde{A}_i = \bar{h}_i \\ \tilde{B}_{ij} = \frac{1}{2}\displaystyle\sum_{k=1}^{3} \bar{g}_{ik}\bar{g}_{jk} \end{cases} \tag{4.35}$$

Then, from equation (4.35), (4.33), (4.7) and (4.27) it is clear that

$$\begin{cases} \bar{h}_i = \tilde{h}_i \\ \frac{1}{2}\displaystyle\sum_{k=1}^{3} \bar{g}_{ik}\bar{g}_{jk} = \frac{1}{2}\sum_{k=1}^{3} \tilde{g}_{ik}\tilde{g}_{jk} \end{cases} \tag{4.36}$$

With (4.36) is has been proved that the transformed fluctuation equation, defined by (4.26) and (4.27), corresponds with the transformed Fokker-Planck equation, given in (4.32).

- Consistency with the advection-diffusion equation - a general coordinate system

By inserting (4.9) into equation (4.27) the transformed drift and noise components are defined in terms of the drift and noise components in Cartesian coordinates. It results in

$$\left\{\begin{aligned} \tilde{h}_i &= \left(\frac{\partial \xi^i}{\partial t}\right)_{\underline{x}} + \sum_{j=1}^{3} \frac{\partial \xi^i}{\partial x_j}\left(u_j + \sum_{k=1}^{3} \frac{\partial D_{jk}}{\partial x_k}\right) + \sum_{j=1}^{3}\sum_{k=1}^{3} \frac{\partial^2 \xi^i}{\partial x_j \partial x_k} D_{jk} \\ \tfrac{1}{2}\sum_{k=1}^{3} \tilde{g}_{ik}\tilde{g}_{jk} &= \sum_{k=1}^{3}\sum_{l=1}^{3} \frac{\partial \xi^i}{\partial x_k}\frac{\partial \xi^j}{\partial x_l} D_{kl} \end{aligned}\right. \qquad (4.37)$$

For computational purposes, equation (4.37) is likely to be expressed in terms of the contravariant velocity components, U^i, and the contravariant base vectors, $\underline{a}^i$. Using the transformation relations given in appendix A, (4.37) is rewritten as

$$\left\{\begin{aligned} \tilde{h}_i &= U^i + \sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} \left(\underline{a}^i\right)_j \left(\underline{a}^l\right)_k \frac{\partial D_{jk}}{\partial \xi^l} + \sum_{j=1}^{3}\sum_{k=1}^{3}\left(\sum_{l=1}^{3} \left(\underline{a}^l\right)_j \frac{\partial}{\partial \xi^l}\left[\left(\underline{a}^i\right)_k\right]\right) D_{jk} \\ \tfrac{1}{2}\sum_{k=1}^{3} \tilde{g}_{ik}\tilde{g}_{jk} &= \sum_{k=1}^{3}\sum_{l=1}^{3} \left(\underline{a}^i\right)_k \left(\underline{a}^j\right)_l D_{kl} \end{aligned}\right. \qquad (4.38)$$

Formally, $\tilde{h}_i$ in equation (4.38) is an abbreviation of $\tilde{h}_i(\underline{x},t)$, with $\underline{x}$ replaced by its transformation in terms of $\underline{\xi}$ and t. It is seen from equation (4.38) that the transformed drift component $\tilde{h}_i$ depends on (i) the local flow velocity relative to the coordinate system, (ii) the transformed noise-induced drift component, and (iii) the curvature of the transformation under consideration with respect to the stochastic displacement. The latter contribution results from the stochastic chain rule.

- Consistency with the advection-diffusion equation - an orthogonal curvilinear transformation in the horizontal plane and the σ - transformation in the vertical plane

In chapter 2 a hydrodynamic model has been discussed which introduces an orthogonal curvilinear transformation in the horizontal plane and the σ - transformation in the vertical plane. Also a simplified diffusion tensor has been specified. For this particular case, the drift and noise components will be given explicitly. It is argued in section 2.4 that the hydrodynamic model returns velocity components relative to the moving grid, u_ξ, u_η, and ω. These velocity components, together with the contravariant base vectors will be inserted into equation (4.37). The explicit expressions serve the simplicity of the numerical implementation of the particle model relative to the hydrodynamic model. The numerical implementation will be discussed in section 4.3.2.

For convenience, the transformed drift component is composed of three parts, defined in

(4.39). Each part will be treated separately.

$$\underset{\substack{\text{Influence of}\\ \text{transformed}}}{\tilde{h}_i} \;=\; \underbrace{\left(\tilde{h}_i\right)_U}_{\substack{\text{local flow}\\ \text{velocity}}} \;+\; \underbrace{\left(\tilde{h}_i\right)_D}_{\substack{\text{noise-induced}\\ \text{drift}}} \;+\; \underbrace{\left(\tilde{h}_i\right)_C}_{\substack{\text{noise-induced}\\ \text{curvature}}} \tag{4.39}$$

The noise-induced drift comes from consistency considerations. The term referred to as noise-induced curvature arises from the Itô integration rule. It is only envisaged in the presence of diffusion.

Due to orthogonality in the horizontal plane, the influence of the transformed local flow velocity becomes, for every component

$$\begin{cases} \left(\tilde{h}_1\right)_U = \dfrac{1}{\sqrt{G_{\xi\xi}}}\,u_\xi \\ \left(\tilde{h}_2\right)_U = \dfrac{1}{\sqrt{G_{\eta\eta}}}\,u_\eta \\ \left(\tilde{h}_3\right)_U = \dfrac{1}{H}\,\omega \end{cases} \tag{4.40}$$

The expressions for the second and third item of (4.39), become more complex due to nonorthogonality and anisotropy. To express the influence of nonorthogonality the scalar products of the contravariant base vectors are used. Furthermore, it is noted that - see appendix B,

$$\left(\underline{a}^1\right)_3 = 0 \qquad \left(\underline{a}^2\right)_3 = 0 \qquad \left(\underline{a}^1\cdot\underline{a}^2\right) = 0 \tag{4.41}$$

With (4.41) the influence of the transformed noise-induced drift becomes

$$\begin{cases} \left(\tilde{h}_1\right)_D = \dfrac{1}{G_{\xi\xi}}\dfrac{\partial D_H}{\partial \xi} + \left(\underline{a}^1\cdot\underline{a}^3\right)\dfrac{\partial D_H}{\partial \sigma} \\ \left(\tilde{h}_2\right)_D = \dfrac{1}{G_{\eta\eta}}\dfrac{\partial D_H}{\partial \eta} + \left(\underline{a}^2\cdot\underline{a}^3\right)\dfrac{\partial D_H}{\partial \sigma} \\ \left(\tilde{h}_3\right)_D = \dfrac{1}{H^2}\dfrac{\partial D_V}{\partial \sigma} + \left(\underline{a}^1\cdot\underline{a}^3\right)\dfrac{\partial D_H}{\partial \xi} + \left(\underline{a}^2\cdot\underline{a}^3\right)\dfrac{\partial D_H}{\partial \eta} \\ \qquad\qquad + \left[G_{\xi\xi}\left(\underline{a}^1\cdot\underline{a}^3\right)^2 + G_{\eta\eta}\left(\underline{a}^2\cdot\underline{a}^3\right)^2\right]\dfrac{\partial D_H}{\partial \sigma} \end{cases} \tag{4.42}$$

In the expression for the influence of the curvature term the contribution of the vertical diffusion coefficient D_V vanishes because of

$$(\underline{a}^1)_3 = 0 \qquad (\underline{a}^2)_3 = 0 \qquad \frac{\partial}{\partial \sigma}\left[(\underline{a}^i)_3\right] = 0 \quad , \; i = 1,2,3 \tag{4.43}$$

From these relations the following conservative form of $\left(\tilde{h}_i\right)_C$ is deduced, for i=1,2,3

$$\left(\tilde{h}_i\right)_C = \frac{D_H}{J}\sum_{j=1}^{3}\frac{\partial}{\partial \xi^j}\left[J\left(\underline{a}^i \cdot \underline{a}^j\right)\right] \tag{4.44}$$

Finally, the stochastic forcing components, $\tilde{g}_{ij}$, are found in

$$\tilde{g} = \begin{pmatrix} \frac{1}{\sqrt{G_{\xi\xi}}}\sqrt{2D_H} & 0 & 0 \\ 0 & \frac{1}{\sqrt{G_{\eta\eta}}}\sqrt{2D_H} & 0 \\ \left(\underline{a}^1 \cdot \underline{a}^3\right)\sqrt{G_{\xi\xi}}\sqrt{2D_H} & \left(\underline{a}^2 \cdot \underline{a}^3\right)\sqrt{G_{\eta\eta}}\sqrt{2D_H} & \frac{1}{H}\sqrt{2D_V} \end{pmatrix} \tag{4.45}$$

It is emphasized that the given expressions for the transformed drift and noise components depend on the assumptions that

(i) the curvilinear transformation in the horizontal plane is orthogonal, $\left(\underline{a}^1 \cdot \underline{a}^2\right) = 0$, and
(ii) the diffusion tensor expresses an isotropic diffusion in the horizontal plane. Moreover, the off-diagonal elements of the diffusion tensor are zero.

An orthogonal transformation simplifies the expressions given in (4.42), (4.44) and (4.45) considerably. If $\left(\underline{a}^i \cdot \underline{a}^j\right) = 0$ and $\partial / \partial \xi^j \left[J\left(\underline{a}^i \cdot \underline{a}^j\right)\right] = 0, \; \forall \; i,j$, then the transformed relations are very similar to those expressed in Cartesian coordinates - see equation (4.10).

4.3 Numerical aspects of particle models

This section gives a full description of the numerical implementation of the particle model.

Since a particle model describes discrete particle positions and corresponding mass elements, a conversion algorithm is needed to obtain a concentration distribution. This concentration distribution is regarded as an approximation of the solution of the advection-diffusion equation. Concepts of possible conversion algorithms are given in section 4.3.1.

The equations for the particle displacements in Cartesian coordinates, as well as in transformed coordinates, are discretized to obtain approximated sample-path solutions. The implementation is discussed in section 4.3.2.

Transport problems are written as initial value problems. Applying particle models the initial concentration distribution must be given within a stochastic framework. It will be illustrated in section 4.3.3.

Section 4.3.4 gives a detailed description of the advection step used in the particle model. It is developed such that consistency with the mass balance equation, as it is approximated numerically in the hydrodynamic model, is guaranteed. It provides accurate simulation of streamlines in structured grids.

The extension of the particle model to transport problems including (exponential) decay, additional source terms and various boundary conditions will be discussed in section 4.3.5 and 4.3.6 respectively.

4.3.1 Derivation of the concentration distribution from a discrete mass density

When solving the advection-diffusion equation with a finite difference method, concentration values are computed at discrete points in a grid. These concentration values can easily be used to construct a continuous concentration field which expresses the influence of dispersal on, for instance, the contaminant of interest. This concentration field is then readily visualized for further engineering purposes. The particle method does not produce concentration values. It returns discrete particle positions indicating the presence of a certain amount of mass defined by the corresponding mass element. The distribution of these mass elements must, for various reasons, be converted into a concentration field:

(i) Visualization of a concentration field can serve engineering purposes. Plotting the particle positions is known as an alternative visualization technique to gain insight in transport phenomena - see JONG and HEEMINK (1993).

(ii) A continuous concentration distribution simplifies the comparison between the simulation results of the particle method and other transport solvers.

(iii) Implementation of some specific boundary conditions in particle models requires the estimation of the mass flux through boundaries. Consequently, knowledge of the spatial derivative of the concentration distribution is necessary - see section 4.2.2. Such a derivative can be deduced from a continuous concentration distribution.

Three conversion algorithms are dealt with in more detail. The distributions obtained with the corresponding techniques are referred to as

- the numerical defined distribution,
- the point spread function, and
- the maximum entropy distribution.

The associated algorithms are described separately. Comparisons will be given based on numerical experiments.

- The numerical defined distribution (NDD)

The numerically defined distribution is obtained by counting the frequency of observations, i.e., particle positions, in predefined intervals. A scaling procedure corrects the area under the curve - see SIDDALL (1983). So, in one dimension, the computational domain should first be covered by a finite number of intervals, each equally spaced. This number is essentially based on judgement. Too many intervals will result in a considerable scatter from a fitted curve through a set of points at the midpoint of each interval corresponding to the frequencies, while too few intervals will not give a meaningful defined curve.

In three dimensions, the physical domain is covered with N^c grid cells I_{n^c}, $n^c = 1,..,N^c$. The concentration value associated with the numerically defined distribution, represents a cell-averaged concentration value, and equals

$$\overline{C}^n_{n^c} = C^{\mathrm{NDD}}(\underline{x},t_n)\Big|_{\underline{x}\in I_{n^c}} = \frac{1}{V_{n^c}} \sum_{\substack{p=1 \\ X^p(t_n)\,\in\, I_{n^c}}}^{\mathrm{n_p}^n} \mu^n_p \tag{4.46}$$

in which

$\mathrm{n_p}^n$ = total number of particles in the computational domain at time $t = t_n$,

V_{n^c} = volume of grid cell I_{n^c},

$X^p(t_n)$ = coordinate of particle with index p at time $t = t_n$, and

μ^n_p = mass of particle at $X^p(t_n)$.

Equation (4.46) immediately corresponds with the intuitive idea of a cell-averaged concentration value. It is obtained by counting the number of particles within a fixed grid, taking into account the mass of each particle. Equation (4.46) is a realization of the mean concentration regarded as an expectational value - see section 4.2.2.

It seems natural to choose the grid cells I_{n^c} in correspondence with the grid cells introduced by the hydrodynamic model. However, one should always be aware of the fact that the representation of the cell-averaged concentration values depends on the grid cell dimensions. Since the concentration distribution is written as a sum of concentration step functions, a meaningful approximation of a continuous concentration distribution is only obtained if the fraction of the original number, which will be found in a given grid cell, is large enough. On the other hand, this fraction may not be too large for certain grid cells. The introduction of a sub grid is sometimes necessary to derive an adequate approximation of the derivative of the concentration distribution. Such a sub grid uses smaller grid cell dimensions than the basic grid. If the number of particles with respect to this sub grid is too small, the simulation should be repeated with an increase of particles to be released initially. It is concluded that solutions obtained with this conversion algorithm are influenced by the judgement of the user in choosing an adequate mesh.

- The point spread function (PSF)

Another method to convert the mass density of the discrete particles into concentrations makes use of the so-called point spread function. The corresponding algorithm sums the significant particle influences in the vicinity of the calculation point $\underline{x}$ with a particular mass distribution φ centred on the particle locations $\underline{X}^p(t_n)$. φ is referred to as the point spread function. The associated concentration distribution is given by

$$C^{\mathrm{PSF}}(\underline{x},t_n) = \sum_{p=1}^{n_p^n} \mu_p^n \; \varphi\left(\underline{x}-\underline{X}^p(t_n) \,|\, \underline{\sigma}(\underline{x},t_n)\right) \tag{4.47}$$

An example of φ in three dimensions is given by

$$\varphi(\underline{x}\,|\,\underline{\sigma}) = \frac{1}{(2\pi)^{3/2}\sigma_1\sigma_2\sigma_3}\exp\left[-\frac{1}{2}\sum_{i=1}^{3}\left(\frac{x_i}{\sigma_i}\right)^2\right] \tag{4.48}$$

The point spread function φ, defined in (4.48), contains some degrees of freedom determined by $\underline{\sigma}$. The number of degrees of freedom depends on the dimensionality of the problem. The corresponding variances, σ_i^2, still have to be specified. It is noted that these variances affect the smoothness of the mass distribution. As illustrated by MONTMINY *et al.* (1992), small values of σ_i, does not yield a smooth concentration distribution, while large values of σ_i exaggerates the width of the global distribution.

A meaningful choice of $\underline{\sigma}$ is obtained by minimizing the following expression with respect to $\underline{\sigma}$ for given distribution C:

$$e\left(\underline{\sigma}^n \,|\, C^n, \mathrm{n_p}^n, t_n\right) = E\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[C(\underline{x},t_n) - \sum_{p=1}^{\mathrm{n_p}^n} \mu_p^n \,\varphi\left(\underline{x}-\underline{X}^p(t_n)\,|\,\underline{\sigma}(\underline{x},t_n)\right)\right]^2 d\underline{x}\right] \qquad (4.49)$$

Expression (4.49) minimizes the expected error in the approximation of C with C^{PSF}. Since the transport process represents a diffusion process, it is natural to minimize equation (4.49) with C a gaussian distribution. For simplicity, the mean is set to zero. The covariance matrix equals $2Dt_n$, with D a constant diffusion tensor: $D = \mathrm{diag}\left(D_1,D_2,D_3\right)$. Explicit expressions for optimal σ_i^n, i=1,2,3, are obtained from evaluation of equation (4.49), while the expectation E is regarded as an ensemble average of I samples of $\underline{X}^p\left(t_n\right)$. The optimal choice of σ_i^n depends on the variance of the diffusion process and the number of released particles. Such a dependency is recognized in BOOGAARD *et al.* (1993), where σ_i^n is written as

$$\sigma_i^n = \beta_i\left(\mathrm{n_p}^n\right)\sqrt{2D_i t_n} \qquad (4.50)$$

It is noted that (4.50) is only used in applications with $D_i > 0$. (4.50) is meaningless for advective transport without diffusion. The optimization procedure has been executed for the one-dimensional and the two-dimensional case with $I = 25$ samples. For simplicity $D_2 = D_1$ is assumed in the two-dimensional case, so that there is no reason to distinguish between σ_1^n and σ_2^n. The results are sketched in figure (4.3) and (4.4), where the coefficient β_i is plotted as a function of the number of particles. For large $\mathrm{n_p}^n$, it is deduced that:

$$\sigma_1^n = 0.959874\,\frac{\sqrt{2D_1 t_n}}{\left(\mathrm{n_p}^n\right)^{0.185440}} \qquad \text{(one-dimensional case)} \qquad (4.51a)$$

$$\sigma_i^n = 1.146612\,\frac{\sqrt{2D_i t_n}}{\left(\mathrm{n_p}^n\right)^{0.181091}} \qquad i = 1,2 \text{ (two-dimensional case)} \qquad (4.51b)$$

It is noted that C^{PSF} expresses a continuous concentration distribution that can be readily differentiated. The applicability of the point spread function algorithm is limited to regions away from open boundaries. Near boundaries the point spread function φ reaches outside the physical domain. If a zero flux boundary condition is applied at closed boundaries, C^{PSF} is written as the sum of two contributions. The first contribution is given by equation (4.47), while the second contribution consists of equation (4.47) evaluated at the image particle position. It is concluded that each specific application, involving boundary conditions, requires point spread functions that are restricted to the computational domain.

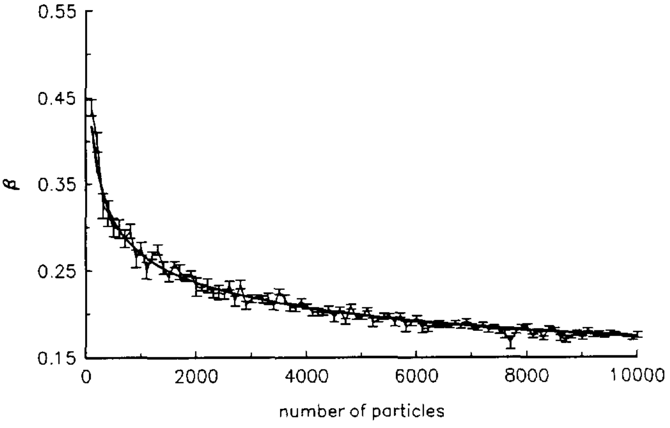


Figure (4.3): Optimal standard deviation of *one-dimensional* point spread function, scaled with the standard deviation of the diffusion process $\sqrt{2D_1 t}$. Error bars denote 3 times standard deviation of ensemble average. For large n_p:

$$\beta_1 = 0.959874 \left(n_p\right)^{-0.185440}$$

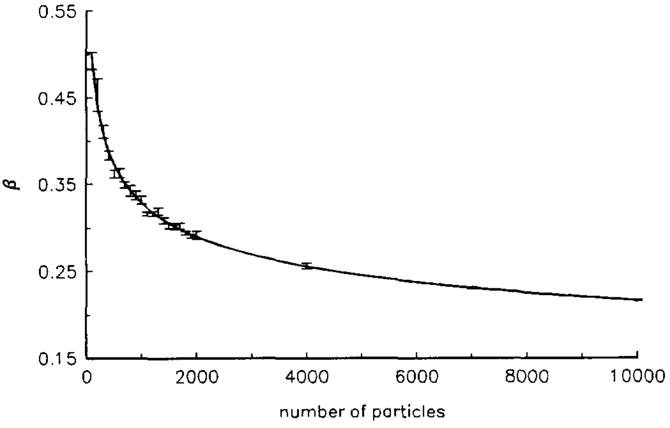


Figure (4.4): Optimal standard deviation of *two-dimensional* point spread function, scaled with the standard deviation of the diffusion process $\sqrt{2D_i t}$. Error bars denote 10 times standard deviation of ensemble average. Optimization procedure executed for n_p = 10000, 7000, 4000, 100j, j=1,..,20. For large n_p, i=1,2:

$$\beta_i = 1.146612 \left(n_p\right)^{-0.181091}$$

- The maximum entropy distribution (MED)

A third approach to estimate a continuous concentration distribution from a discrete mass distribution is based on information theory. It is used to obtain the maximum entropy estimates of the (mean) concentration distribution in the case where several low-order moments of the particle positions (i.e., $E\left[(X^p)^i\right]$, $i \geq 1$) are known. It is usually applied to obtain the probability density function of a scalar contaminant - see DERKSEN and SULLIVAN (1990), instead of the determination of the mean concentration distribution. JAYNES (1957) argues that "*It is the least biased estimate possible on the given information; i.e., it is maximally noncommittal with regard to missing information*". In the absence of theoretical or physical motivation for the selection of a particular (orthogonal) polynomial, the maximum entropy formulation is justified - see SHORE and JOHNSON (1980). The concept of the entropy method is described in SIDDALL (1983). The maximum entropy can also be applied directly to physical distributions other than probabilities - see SKILLING (1988). For simplicity, some one-dimensional results are given here.

For a continuous variable C^n, the entropy G is defined as

$$G(C^n) = - \int_{\mathbb{R}} C^n(x) \ln\left[C^n(x)\right] dx \qquad (4.52)$$

Again, the superscript n expresses time dependency. The entropy is interpreted as a measure of uncertainty for the whole range of values of x. The particle model returns particle positions. These locations express information about the concentration distribution for all x. From this information the moments are computed. If the entropy is maximized, subjected to the moment constraints, the maximum entropy distribution is obtained. These moment constraints are obtained by evaluation of the moments, s_i, defined as

$$s_i^n = \frac{1}{\mu_{\text{tot}}^n} \sum_{p=1}^{n_p^n} \mu_p^n \left[X^p(t_n)\right]^i \qquad \text{with} \qquad \mu_{\text{tot}}^n = \sum_{p=1}^{n_p^n} \mu_p^n \qquad (4.53)$$

In the maximum entropy procedure the normalized C^n is adjusted to achieve a maximum of

$$\bar{G}(C^n) = G(C^n) + (\lambda_0 + 1)\left[\int_{\mathbb{R}} C^n(x)dx - 1\right] + \sum_{i=1}^{m} \lambda_i \left[\int_{\mathbb{R}} x^i C^n(x)dx - s_i^n\right] \qquad (4.54)$$

The modified function $\bar{G}$ accounts for (i) the entropy G, (ii) a normalized concentration distribution ($\int_{\mathbb{R}} C^n(x)dx = 1$), and (iii) the moment constraints with λ_i the so-called Lagrange multipliers. The maximum entropy distribution, obtained by setting

$d\bar{G}/d(C^n(x)) = 0$, has a specific analytical form

$$C^{\mathrm{MED}}(x) = \exp\left(\lambda_0 + \sum_{i=1}^{m} \lambda_i x^i\right) \tag{4.55}$$

with m the number of moments to be used. To account for not normalized physical distributions λ_0 is adjusted. Although the algorithm is not restricted to diffusion processes, the analytical form makes it extremely suitable for simulations of such diffusion processes.

The procedure applied on a finite interval can be found in DOWSON and WRAGG (1973). In SIDDALL (1983) it is explained how to generate the Langrange multipliers. This source even includes a software package. Both references deal with one-dimensional applications. The three-dimensional maximum entropy distribution takes the form

$$C^{\mathrm{MED}}(x_1,x_2,x_3) = \exp\left(\sum_{i=0}^{m} \sum_{j=0}^{m} \sum_{k=0}^{m} \lambda_{ijk}\, x_1^i x_2^j x_3^k\right) \tag{4.56}$$

The optimization procedure that provides the Lagrangian multipliers λ_{ijk} is cumbersome. It is similar to the one-dimensional case and does not contribute additional information about this conversion algorithm. It will not be further discussed.

Just as in the point spread function algorithm this version of the maximum entropy distribution does not account for boundary conditions. If the procedure is subjected to boundary conditions, additional Lagrange multipliers need be introduced. It should result in an accurate global approximation of the continuous concentration distribution. This procedure is excluded from the experiments due to its computational burden.

- Illustration of the introduced conversion algorithms

A one-dimensional diffusion process is chosen to illustrate the conversion algorithms in an unbounded domain. The sample data exists of $n_p^0 = 1000$ realizations of $X^p(t)$. The results are plotted in figure (4.5). It is obvious that with 20 frequency intervals, the numerically defined distribution shows a considerable scatter from the gaussian distribution. The point spread function gives adequate results. Even if the number of realizations decreases, this method still remains satisfactory in producing a smooth approximate gaussian distribution. Best results are obtained with the maximum entropy distribution with two moment constraints. It is stated that the approximation becomes even better with increasing number of moment constraints.

Although the maximum entropy distribution gives relatively best results, it is not commonly

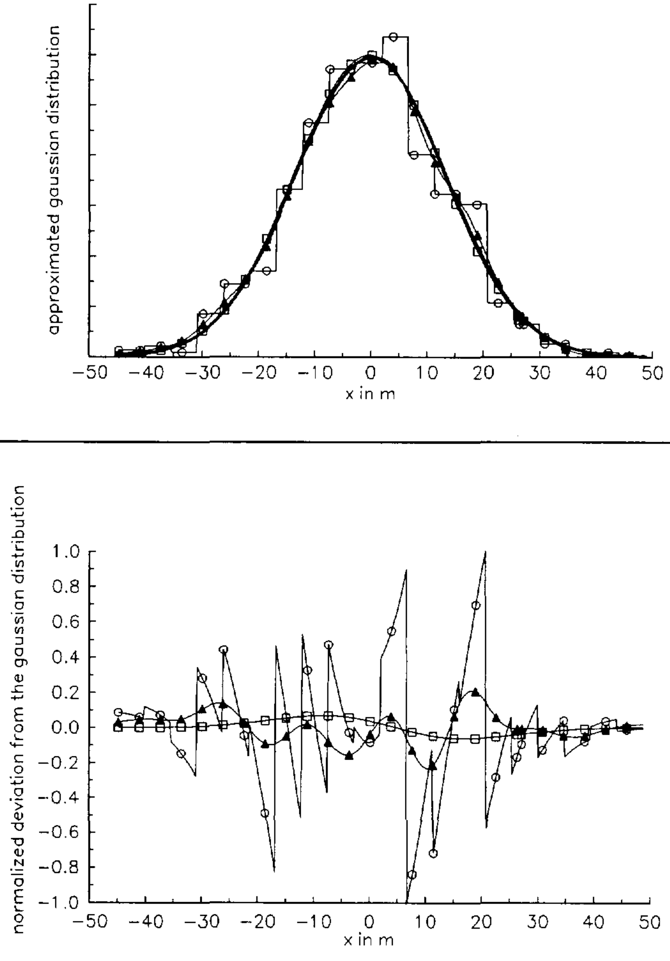


Figure (4.5): Result of three conversion algorithms to approximate a normalized gaussian distribution from a set of sample data.

TOP:	approximated probability density function,
BOTTOM:	deviation from gaussian distribution.

circle:	numerically defined distribution,
triangle:	point spread function,
box:	maximum entropy distribution with two moment constraints,
fat line:	gaussian distribution.

used in particle models. A possible explanation is that the corresponding optimization procedure is highly nonlinear. The nonlinearity affects the robustness of the method. Moreover, an additional number of moment constraints implies a considerable increase of computation time. Finally, the method becomes complicated in three-dimensional applications with a finite domain.

Insight in the use of the various methods was obtained on basis of several numerical experiments. It was found that the conversion algorithms can be distinguished with respect to the following characteristics:

(i) arbitrariness
(ii) complexity (i.e., computation time)
(iii) robustness
(iv) theoretical justification

Recommendations and algorithms are listed in descending order of priority.

	Unbounded regions		Bounded regions	
	one-dimensional	three-dimensional	one-dimensional	three-dimensional
1	C^{MED}	C^{PSF}	C^{NDD}	C^{NDD}
2	C^{PSF}	C^{NDD}	C^{PSF}	C^{PSF}
3	C^{NDD}	C^{MED}	C^{MED}	C^{MED}

4.3.2 Computation of particle displacements

In section 4.2.1 the equation for a particle displacement is expressed as a stochastic differential equation. This equation has been written in terms of local flow velocity components and diffusion coefficients. This section will focus on the numerical implementation of the corresponding random walk model relative to the hydrodynamic model as defined in chapter 2. For that purpose, the following stochastic Itô differential equation has to be integrated numerically

$$dX_i = h_i(X,t)dt + \sum_{j=1}^{3} g_{ij}(X,t)dW_j \tag{4.57}$$

with h_i and g_{ij} as expressed in (4.9) of section 4.2.1. For convenience the description in

Cartesian coordinates is considered first. Realizations of equation (4.57) are obtained if the initial position of particles is prescribed. The initial position is determined by the initial concentration distribution. Details about the representation of the initial condition is given in section 4.3.3. This section continues with some relevant aspects of numerical solution techniques for stochastic differential equations.

- Numerical solution techniques for stochastic differential equations

Various numerical solution techniques exist to obtain information about the solution of the stochastic differential equation (4.57). In general, these numerical techniques differ from the classical deterministic numerical integration methods, since the interpretation of the stochastic differential equation depends on the predefined integration rule. To develop a suitable numerical method to approximate the statistical properties of equation (4.57), various alternatives can be distinguished, such as

(i) The stochastic differential equation is transformed into the Fokker-Planck equation. It yields information about the probability density function associated with equation (4.57). The Fokker-Planck equation is solved using standard techniques such as a classical finite difference method - see RISKEN (1984).

(ii) The statistical properties are also given by the evolution of the moments. An alternative method is obtained if all moments ($E[X^k(t)]$, where X has been raised to the k^{th} power, $k=0,1,2,\ldots$) are approximated. This is realized by solving a set of so-called moment equations. The first moment equation is obtained by taking the expectational value of equation (4.57) using (4.4). The remaining moment equations are derived similarly. This procedure is outlined in DASHEVSKY (1975). This procedure is attractive in applications where the drift and noise components are written as polynomials, since the expectational value of the right-hand side of (4.57) is then expressed in terms of moments. An infinite set of deterministic equations remains in which the moments act as unknowns; higher order moments appear in the evolution equations for the lower order moments. This infinite set of interrelated equations can only be solved if a closure scheme is applied which constitutes a closed system of equations.

(iii) The stochastic differential equation is integrated directly, which means that statistically representative trajectories are generated. Since the process given in (4.57) is Markovian - see section 4.2.1, trajectories are constructed at discrete times with a procedure that marches forward in time. The characteristics of the process are obtained from the approximated constructed sample paths. An example of a possible algorithm is given in PALLESCHI and ROSA (1992). The method is referred to as the method of sample-path solutions - see KLOEDEN and PLATEN (1992) - or simply the random walk method.

The first and second method will not be illustrated here. The first method corresponds with the Eulerian approach and the second method is not attractive since (i) any closure scheme provides only limited information about the solution, and (ii) the method is only applicable to problems where the drift and noise components can be written as polynomials.

Here the third method is considered. It deals with the original stochastic differential equation, which has to be integrated numerically. To reduce equation (4.57) into an efficient random walk procedure some comments are made about the order of a numerical method. Some well-known integration procedures will be reflected to end up with a numerical integration procedure that will be used to simulate the particle trajectories.

- The order of convergence

In a deterministic case the considered differential equation is approximated with a so-called difference equation. The order of the associated solution method is related to the accumulated truncation error. This order can be determined by inserting Taylor series expansions. In the stochastic case the concept of convergence is generalized to random variables in several ways - see KARLIN and TAYLOR (1975) or KLOEDEN and PLATEN (1992). If one is interested in the approximate probability density function p associated with equation (4.57), then it is required that $E\left[p\left(\bar{\underline{X}},t\right)\right] \rightarrow E\left[p\left(\underline{X},t\right)\right]$ for $\Delta t \rightarrow 0$, where a bar indicates the numerical method. This type of convergence is called convergence in distribution. A stronger convergence criterium is observed if $E\left[\left|\bar{X}(t) - X(t)\right|^2\right] \rightarrow 0$ for $\Delta t \rightarrow 0$. This is called convergence in quadratic mean or mean square convergence and deals with the sample paths themselves. Examples of algorithms that converge in distribution are given in MILSTEIN (1978) or HAWORTH and POPE (1986). Algorithms with a strong convergence behaviour are obtained if the sample paths are expanded in Taylor series. Examples are given in RAO *et al.* (1974). Algorithms that converge in quadratic mean are given in GREINER *et al.* (1988).

It is reminded that the particle model is used as an alternative method for solving the advection-diffusion equation. It is not expected that a solution that converges only in distribution can compete with the accuracy that is obtained by using a traditional finite differencing method for solving the advection-diffusion equation. Therefore, the numerical solution technique that is used to formulate the particle model is chosen such that mean square convergence is guaranteed.

- In search for higher order methods

The choice of an efficient and accurate numerical solution method that converges in the mean square limit depends on - see GREINER *et al.* (1988) and RÜMELIN (1982):

(i) the dimensionality of the problem (here the dimension equals three), and
(ii) the assumptions about the diffusion tensor (anisotropy and dependency on $\underline{x}(t)$, the latter property is referred to as multiplicative noise).

In case of anisotropy and multiplicative noise numerical schemes are obtained if the sample paths are approximated considering

$$X_i(\Delta t) = X_i(0) + \int_0^{\Delta t} \underline{h}_i(\underline{X}(t),t)dt + \int_0^{\Delta t} \sum_{j=1}^{3} g_{ij}(\underline{X}(t),t)dW_j \tag{4.58}$$

Expansion of drift and diffusion terms about $t = 0$ yields expressions for the sample paths that consist of functionals of the Wiener process. For instance, $\int_0^{\Delta t} W_i dW_j$ is one of the terms that arise in the approximated sample path equation. It is noted that this particular nonlinear functional can not be integrated with an accuracy of $O(\Delta t^3)$ for $i \neq j$. Indeed, Rümelin proved that the highest order of convergence in the mean square sense is in general only $O(\Delta t^2)$, attained by the Euler method - see RÜMELIN (1982). In the one-dimensional case or in case of additive noise (that is, the diffusion tensor is independent of $\underline{x}(t)$) higher order methods are available. For instance, in the one-dimensional case the nonlinear functional $\int_0^{\Delta t} W dW = ½(W^2(\Delta t) - \Delta t)$, as a consequence of the Itô formulae. This expression is used to obtain the well-known Milstein method - see e.g., MIL'SHTEIN (1974).

It is noted that many methods claim to be of higher order. Actually, it means that either another type of convergence is considered or that the application is assumed to be one-dimensional or contains additive noise. With respect to convergence in distribution higher order methods can be obtained. Haworth and Pope argue that it is worthwhile studying such higher order methods to gain in efficiency - see HAWORTH and POPE (1986). Some simple methods are discussed in GREINER *et al.* (1988). Their paper encloses the (stochastic) Euler method, the Milstein method and the (stochastic) Heun method.

Here, in the three-dimensional multiplicative case the Euler method is chosen to simulate sample-path solutions for equation (4.57).

- The random walk procedure; the Euler method.

In section 4.2.1 it is discussed that the interpretation of a stochastic differential equation depends on the prescribed integration method. As a consequence, the numerical integration of equation (4.57) must be chosen in accordance with the sense in which the stochastic differential equation is understood. For the Itô equation (4.57) an explicit numerical method remains. In fact, the stochastic part of the Itô equation may only be integrated by an explicit scheme, since any other integration procedure corresponds to the solution of (4.57) interpreted in a different sense (e.g., the sense of Stratonovich). To obtain an implicit method the Itô equation is transformed into a Stratonovich equation. In accordance, this transformed equation is subjected to an implicit numerical scheme. Such a procedure is illustrated in e.g., RÜMELIN (1982), applied to the one-dimensional stochastic Heun scheme.

The numerical solution technique for equation (4.57) will now be given, bearing in mind that:

(i) Numerical integration of the stochastic part of equation (4.57) must be chosen in accordance with the Itô interpretation.

(ii) Convergence in the mean square sense is desired, since the numerical method provides the behaviour of the realizations of solutions of equation (4.57).

(iii) From a computational point of view implicit schemes are less efficient than explicit methods. Moreover, fully implicit schemes require a transformation of equation (4.57).

(iv) In three-dimensional applications with multiplicative noise the highest order of convergence in the mean square sense is only $O(\Delta t^2)$, attained by the Euler method.

These aspects motivate the following numerical method. The simulation is performed at discrete times, say $t_n = t_0 + n\Delta t$. Without loss of generality t_0 is set to zero. The increment of $\underline{X}(t)$ is determined by the transition from state $n\Delta t$ to state $(n+1)\Delta t$. The transition is approximately equal to, i=1,2,3,

$$\Delta X_i^{(n)} = X_i^{(n+1)} - X_i^{(n)} = \int_{t_n}^{t_{n+1}} h_i(\underline{X}(t),t)dt + \sum_{j=1}^{3} g_{ij}(\underline{X}^{(n)},t_n)\Delta W_j \tag{4.59}$$

with

$$\Delta W_j = W_j(t_{n+1}) - W_j(t_n) \tag{4.60}$$

The superscript between parentheses denotes the number of evolved time steps; $X_i^{(n)} = X_i(n\Delta t)$. It is noted that the stochastic part of equation (4.59) corresponds with the Euler method.

For a direct implementation the following issues need to be considered:

-1- the evaluation of $g_{ij}\left(X^{(n)},t_n\right)$,

-2- the realization of ΔW_j,

-3- the approximation of the deterministic drift as it appears in equation (4.59), and

-4- the computation of particle displacements in the (transformed) computational space.

The evaluation of $g_{ij}\left(X^{(n)},t_n\right)$, in three dimensions, is obtained by a trilinear interpolation procedure. For the realization of ΔW_j it is necessary to generate random processes on a digital computer. For that purpose random number generators are required. The approximation of the deterministic integral consists of two contributions, i.e., the local flow velocity and the noise-induced drift - see equation (4.9) of section 4.2.1. Integration of the local flow velocity field will be discussed in detail in section 4.3.4. The influence of the noise-induced drift is well-known in literature. Omission of the noise-induced drift results in an artificial migration of particles and it is concluded that this term should be taken into account to avoid artificial sources/sinks - see e.g., UFFINK (1990). This section continues with remarks about random number generators and some considerations about the computation of particle displacements in the (transformed) computational space. Finally, the complete numerical procedure is summarized in case of an orthogonal curvilinear transformation in the horizontal plane and the σ - transformation in the vertical plane.

- Random number generators

To simulate the (random) transformed particle position it is necessary to simulate the increment ΔW_j at each time step. The statistical characteristics of ΔW_j are derived from equation (4.4) of section 4.2.1. As a consequence, ΔW_j represents a Gaussian random number with mean zero and variance Δt. It can be realized with, for instance, the Box-Muller scheme, as described in RIPLEY (1987). It is however noted that the numerical procedure produces an approximate realization of the particle position that corresponds with the chosen realization of the process $W_j(t)$. The Euler integration method, expressed in (4.59), is only first order accurate with respect to Δt. A first order accurate realization of ΔW_j is therefore sufficient. This is achieved by

$$\Delta W_j = \sqrt{3\Delta t}\; R_j \tag{4.61}$$

R_j represents a uniform random number in $(-1,1)$, where the mean and variance of the random variable at the right-hand side of equation (4.61) equals the mean and variance of the Wiener increments. This approximation benefits the computational efficiency without affecting the order of convergence.

Since R_j is produced by a digital computer, and therefore defined in a completely deterministic way, it is called a pseudo random number. These pseudo random numbers have to be uniformly distributed, stochastically independent, reproducible, easy to compute and stored with a minimum of memory capacity. Since a random number is machine-dependent, a "good" random number does not have to be "good" on every computer. For instance, the limited precision of the multiply operator of a 16-bits register computer differs from the one of a 32-bits register machine.

Section 4.4 reports some numerical experiments. These experiments are executed with the following mixed congruental generator

$$\begin{cases} r_j = \left(a r_{j-1} + c\right) \bmod m \\ r_0 = 0 \end{cases} \tag{4.62}$$

with

$$\begin{cases} a = 5243 \\ c = 55397 \\ m = 262139 \end{cases} \quad \text{and} \quad R_j = 2\frac{r_j}{m} - 1 \tag{4.63}$$

It has been observed that the artificial drift (that is, realization of $E\left[R_j\right]$ deviates from zero) introduced by this type of pseudo random number generator is negligible compared with other numerical errors. More details about the properties of this type of generator together with the stochastic and theoretical tests are described in e.g., LEWIS and ORAV (1989). The multiplicative congruental random number generators (i.e., $c = 0$) are more efficient from a computational point of view. These generators are tested in FISHMAN and MOORE (1985) for $m = 2^{31} - 1$. Such a large modulus is only effective on computers with big word sizes.

- Sample-path solutions in the original (physical) space versus sample-path solutions in the computational (transformed) space.

The computation of sample paths can be executed in a (transformed) computational space. The transformed analogue of (4.59) becomes

$$\Delta \xi^{i^{(n)}} = \int_{t_n}^{t_{n+1}} \tilde{h}_i(X(\underline{\xi}(t)),t)dt + \sum_{j=1}^{3} \tilde{g}_{ij}\left(X\left(\underline{\xi}^{(n)}\right),t_n\right)\Delta W_j \tag{4.64}$$

with the transformed drift components, $\tilde{h}_i$, and transformed noise components, $\tilde{g}_{ij}$, given in section 4.2.3. The procedure is as complex as in the Cartesian case as soon as $\tilde{h}_i$ and $\tilde{g}_{ij}$ are known. These quantities are written in terms of the local flow velocities and the diffusion coefficients obtained by running a hydrodynamic model - see section 4.2.3. Simulation according (4.64) is straightforward. The computations are performed totally in the transformed space without using of the inverse transformation.

On the other hand, if the local flow velocities and diffusion coefficients are known in the physical space, i.e., with respect to the Cartesian coordinate system, while particle displacements still have to be computed in the (transformed) computational space, then one should be careful in applying the correct procedure. Intuitively, the following procedure may seem natural, but appears to be incomplete:

(i) Transform the initial position of the particle in the computational space into an initial position in the physical space.
(ii) Compute the contribution of the particle displacement due to drift (local flow velocity and noise-induce drift) and the contribution of the particle displacement due to the stochastic forcing (diffusivity).
(iii) Add these two contributions (that is, compute the result of equation (4.59))
(iv) Transform the resulting displacement vector to obtain the contravariant displacement vector.
(v) Compute the approximated particle position after simulation time Δt.

This procedure yields consistent results in the deterministic case only. In the stochastic case an inconsistency is introduced. This is explained by the Itô calculus. In section 4.2.3 it is elucidated that transformation of the physical space into some computational space introduces a noise-induced curvature term. Such a curvature term arises in the deterministic case only if higher order methods are applied. In the stochastic case such a curvature term reveals as a first order term. As a consequence, formally this term should be taken into account. The intuitive method described above fails at stage (iv), where the resulting displacement vector is being transformed. Disregarding curvature terms introduces an additional numerical error. The difference between a deterministic displacement and a displacement which contains a stochastic contribution is illustrated in figure (4.6). It is shown that the orientation of the displacement vector changes with varying time step.

From an engineering point of view it is important to know the impact of the noise-induced curvature term. Omission of this term reduces the computational effort considerably. It is already mentioned that omission of the noise-induced *drift* term introduces numerical sinks. Now the question arises if the absence of the noise-induced curvature term also introduces numerical sinks, or if it is just an additional numerical error which reduces the accuracy. In the latter case the relative importance of this term can be expressed in a dimensionless parameter. In search for such a dimensionless parameter a simple one-dimensional

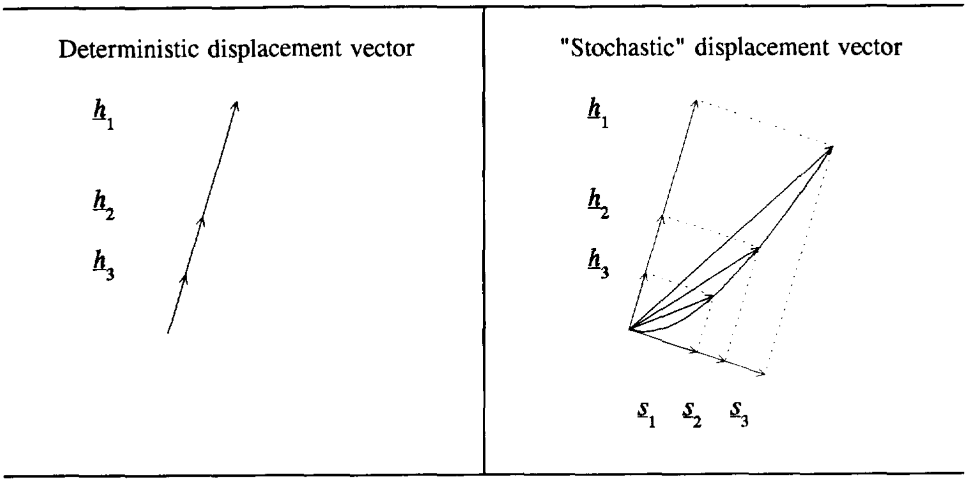

Deterministic displacement vector
"Stochastic" displacement vector
$\underline{h}_1$
$\underline{h}_2$
$\underline{h}_3$
$\underline{h}_1$
$\underline{h}_2$
$\underline{h}_3$
$\underline{s}_1$ $\underline{s}_2$ $\underline{s}_3$

Figure (4.6): Deterministic versus "stochastic" displacement vectors, with

$$\underline{h}_i = \underline{h}\frac{\Delta t}{2^{i-1}} \qquad \underline{s}_i = \underline{s}\sqrt{\frac{\Delta t}{2^{i-1}}} \quad (\underline{s}\ \text{fixed}) \qquad , i=1,2,3.$$

numerical experiment is executed.

A one-dimensional experiment approximates one sample path started at some initial location $x^{(0)}$. N successive positions of one single particle are computed. For simplicity, the flow velocity and the diffusivity are taken constant. The following transformation is considered

$$\xi = x^2 \tag{4.65}$$

This transformation gives rise to the following one-dimensional contravariant base "vector"

$$a = \frac{\partial \xi}{\partial x} = 2x = 2\sqrt{\xi} \tag{4.66}$$

Three sample-path solutions are computed:

(i) sample-path solution in the physical space

$$\begin{cases} x^{(n+1)} = x^{(n)} + u\Delta t + \sqrt{6D\Delta t}\,R \\ \xi^{(n+1)} = \left(x^{(n+1)}\right)^2 \end{cases} \tag{4.67i}$$

(ii) sample-path solution obtained using the inverse transformation (in the absence of the curvature term)

$$\begin{cases} \bar{\xi}^{(n+1)} = \bar{\xi}^{(n)} + 2\sqrt{\bar{\xi}^{(n)}}\left(u\Delta t + \sqrt{6D\Delta t}\,R\right) \\ \bar{x}^{(n+1)} = \sqrt{\bar{\xi}^{(n+1)}} \end{cases} \tag{4.67ii}$$

(iii) sample-path solution in the computational space

$$\begin{cases} \hat{\xi}^{(n+1)} = \hat{\xi}^{(n)} + 2\sqrt{\hat{\xi}^{(n)}}\left(u\Delta t + \sqrt{6D\Delta t}\,R\right) + 2D\Delta t \\ \hat{x}^{(n+1)} = \sqrt{\hat{\xi}^{(n+1)}} \end{cases} \tag{4.67iii}$$

R represents a uniform random number in $(-1,1)$. It is expected that the relative largest errors appear for $u = 0$ m/s. The following normalized error measures are introduced, where the suffix $\{u = 0\}$ accounts for the simulation in the absence of a current

$$\bar{\epsilon}_N = \frac{\left|\xi^{(N)} - \bar{\xi}^{(N)}\right|}{\max\limits_{n\in\{1,..,N\}} \left\{ \left|\xi^{(n)}_{\{u=0\}} - \bar{\xi}^{(n)}_{\{u=0\}}\right|, \left|\xi^{(n)}_{\{u=0\}} - \hat{\xi}^{(n)}_{\{u=0\}}\right| \right\}} \tag{4.68a}$$

and

$$\hat{\epsilon}_N = \frac{\left|\xi^{(N)} - \hat{\xi}^{(N)}\right|}{\max\limits_{n\in\{1,..,N\}} \left\{ \left|\xi^{(n)}_{\{u=0\}} - \bar{\xi}^{(n)}_{\{u=0\}}\right|, \left|\xi^{(n)}_{\{u=0\}} - \hat{\xi}^{(n)}_{\{u=0\}}\right| \right\}} \tag{4.68b}$$

The dimensionless parameter, which relates the influence of the noise-induced curvature with the influence of the local flow velocity, is defined as

$$C_E = \left|\frac{\tilde{h}_C}{\tilde{h}_U}\right| = \left|\frac{\dfrac{\partial a}{\partial x}D}{au}\right| \tag{4.69}$$

Figure (4.7) shows the influence of the curvature term. The intersection of both error measure curves occurs at $C_E \approx 2$. The vertical position of the intersection depends on the physical data. The behaviour of the error measures for $C_E < 1$ is dominated by the chosen time step. Large flow velocities require a small time step.

In general it is expected that ignorance of the curvature term reduces the overall accuracy and introduces numerical sources/sinks in strongly curved or stretched grids. It is recommended to include the curvature term in applications where

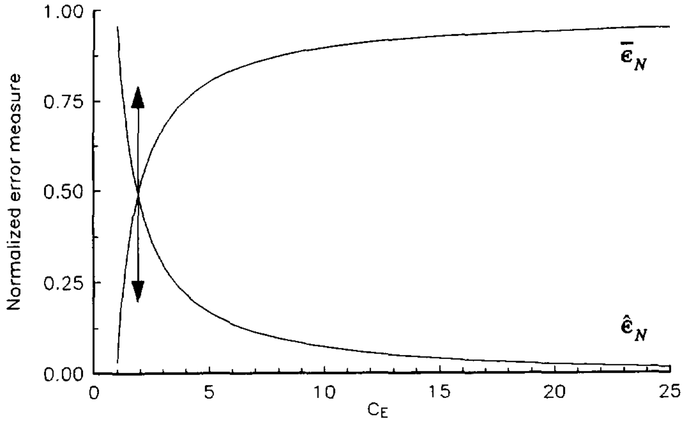


Figure (4.7): The influence of the curvature (C_E) on the accuracy of approximated sample-path solutions. A bar refers to sample-path solutions computed using the inverse transformation. A hat is written to indicate the procedure that is totally performed in the computational space. The arrow shows that the intersection of both error measure curves depends on the flow conditions and the numerical grid. However, the critical point equals $C_E \approx 2$.

$$C_E = \frac{\|\tilde{h}_C\|}{\|\tilde{h}_U\| + \|\tilde{h}_D\|} > 2 \tag{4.70}$$

The influence of the curvature term in two-dimensional applications will be illustrated in section 4.4.2.

- Numerical approximation of sample paths considering an orthogonal curvilinear transformation in the horizontal plane and the σ - transformation in the vertical plane.

Finally, a summary of the computation of particle displacements in a transformed grid is given. The transformation corresponds with the transformation introduced in section 2.4, where an orthogonal curvilinear transformation in the horizontal plane, and the σ - transformation in the vertical plane are considered. The corresponding drift and noise components are reflected in section 4.2.3. As in equation (4.64), the approximated transformed particle displacement becomes, for i=1,2,3

$$\Delta \xi^{i^{(n)}} = \underbrace{\int_{t_n}^{t_{n+1}} \left(\tilde{h}_i\right)_U dt}_{\text{section 4.3.4}} + \underbrace{\left(\tilde{h}_i\right)_D^{(n)} \Delta t + \left(\tilde{h}_i\right)_C^{(n)} \Delta t + \sum_{j=1}^{3} \tilde{g}_{ij}^{(n)} \Delta W_j}_{\text{Euler method}} \tag{4.71}$$

The displacement given in (4.71) is dimensionless because the coordinates ξ^i are defined dimensionless. The corresponding physical displacement in the transformed space equals $\left(\sqrt{G_{\xi\xi}}\,\Delta\xi \;,\; \sqrt{G_{\eta\eta}}\,\Delta\eta \;,\; H\Delta\sigma\right)^T$. It is noted that the particle model, described here, is defined with respect to a predefined grid. This grid is introduced by the hydrodynamic model. Accurate simulations are expected only if the time step is restricted, such that each particle will contact one grid cell boundary at the most per time step. This restriction is similar to the stability condition of the Euler method in the deterministic case:

$$\|\Delta \underline{\xi}^{(n)}\| \leq 1 \tag{4.72}$$

Condition (4.72) is satisfied for sufficiently small time step. With respect to the vertical dimension JIN (1993) states that in shallow water flow problems the maximum length of a random displacement can be equated to the water depth to obtain acceptable concentration distributions that are required for engineering purposes.

Finally the expressions for the various terms in the right-hand side of (4.71) are given. All expressions make use of the flow and grid information obtained by the hydrodynamic model. Equation (4.40), (4.42), (4.44) and (4.45) are revised using the relations given in appendix B. For simulation purposes the expressions still need to be approximated numerically.

Influence of transformed local flow velocity

$$\begin{cases} \left(\tilde{h}_1\right)_U = \dfrac{1}{J}\left(H\sqrt{G_{\eta\eta}}\,u_\xi\right) = \dfrac{1}{\sqrt{G_{\xi\xi}}}\,u_\xi \\ \left(\tilde{h}_2\right)_U = \dfrac{1}{J}\left(H\sqrt{G_{\xi\xi}}\,u_\eta\right) = \dfrac{1}{\sqrt{G_{\eta\eta}}}\,u_\eta \\ \left(\tilde{h}_3\right)_U = \dfrac{1}{J}\sqrt{G}\,\omega = \dfrac{1}{H}\,\omega \end{cases} \tag{4.73}$$

The last expression at the right-hand side of the first two equations is valid in case of an orthogonal transformation in the horizontal plane.

Influence of transformed noise-induced drift

$$\left\{\begin{aligned}
\left(\tilde{h}_1\right)_D &= \frac{1}{G_{\xi\xi}}\frac{\partial D_H}{\partial \xi} - \frac{1}{HG_{\xi\xi}}\frac{\partial z}{\partial \xi}\frac{\partial D_H}{\partial \sigma}\\
\left(\tilde{h}_2\right)_D &= \frac{1}{G_{\eta\eta}}\frac{\partial D_H}{\partial \eta} - \frac{1}{HG_{\eta\eta}}\frac{\partial z}{\partial \eta}\frac{\partial D_H}{\partial \sigma}\\
\left(\tilde{h}_3\right)_D &= \frac{1}{H^2}\frac{\partial D_V}{\partial \sigma} + \frac{1}{H^2}\left[\frac{1}{G_{\xi\xi}}\left(\frac{\partial z}{\partial \xi}\right)^2 + \frac{1}{G_{\eta\eta}}\left(\frac{\partial z}{\partial \eta}\right)^2\right]\frac{\partial D_H}{\partial \sigma}\\
&\quad - \frac{1}{H}\left[\frac{1}{G_{\xi\xi}}\frac{\partial z}{\partial \xi}\frac{\partial D_H}{\partial \xi} + \frac{1}{G_{\eta\eta}}\frac{\partial z}{\partial \eta}\frac{\partial D_H}{\partial \eta}\right]
\end{aligned}\right. \tag{4.74}$$

Influence of transformed noise-induced curvature

$$\left\{\begin{aligned}
\left(\tilde{h}_1\right)_C &= \left[\frac{1}{\sqrt{G_{\xi\xi}}}\frac{\partial}{\partial \xi}\left(\frac{1}{\sqrt{G_{\xi\xi}}}\right) + \frac{1}{\sqrt{G}}\frac{1}{\sqrt{G_{\xi\xi}}}\frac{\partial}{\partial \xi}\left(\sqrt{G_{\eta\eta}}\right)\right] D_H\\
\left(\tilde{h}_2\right)_C &= \left[\frac{1}{\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial \eta}\left(\frac{1}{\sqrt{G_{\eta\eta}}}\right) + \frac{1}{\sqrt{G}}\frac{1}{\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial \eta}\left(\sqrt{G_{\xi\xi}}\right)\right] D_H\\
\left(\tilde{h}_3\right)_C &= \frac{1}{H^2}\frac{1}{\sqrt{G_{\xi\xi}}}\frac{\partial z}{\partial \xi}\left[\frac{D_H}{\sqrt{G_{\xi\xi}}}\frac{\partial H}{\partial \xi} - H\sqrt{G_{\xi\xi}}\left(\tilde{h}_1\right)_C\right]\\
&\quad + \frac{1}{H^2}\frac{1}{\sqrt{G_{\eta\eta}}}\frac{\partial z}{\partial \eta}\left[\frac{D_H}{\sqrt{G_{\eta\eta}}}\frac{\partial H}{\partial \eta} - H\sqrt{G_{\eta\eta}}\left(\tilde{h}_2\right)_C\right]\\
&\quad - \left[\frac{1}{G_{\xi\xi}}\frac{\partial}{\partial \xi}\left(\frac{1}{H}\frac{\partial z}{\partial \xi}\right) + \frac{1}{G_{\eta\eta}}\frac{\partial}{\partial \eta}\left(\frac{1}{H}\frac{\partial z}{\partial \eta}\right)\right] D_H
\end{aligned}\right. \tag{4.75}$$

and finally, the influence of the transformed stochastic forcing

$$\left\{\begin{aligned}
\sum_{j=1}^{3}\tilde{g}_{1j}\Delta W_j &= \frac{1}{\sqrt{G_{\xi\xi}}}\sqrt{6D_H\Delta t}\;R_1\\
\sum_{j=1}^{3}\tilde{g}_{2j}\Delta W_j &= \frac{1}{\sqrt{G_{\eta\eta}}}\sqrt{6D_H\Delta t}\;R_2\\
\sum_{j=1}^{3}\tilde{g}_{3j}\Delta W_j &= \frac{1}{H}\sqrt{6D_V\Delta t}\;R_3\\
&\quad - \frac{1}{H}\sqrt{6D_H\Delta t}\left[\frac{1}{\sqrt{G_{\xi\xi}}}\frac{\partial z}{\partial \xi}R_1 + \frac{1}{\sqrt{G_{\eta\eta}}}\frac{\partial z}{\partial \eta}R_2\right]
\end{aligned}\right. \tag{4.76}$$

with R_1, R_2 and R_3 uniform random numbers in $(-1,1)$.

4.3.3 Representation of the initial condition

As explained in the previous sections, the stochastic process is approximated by an ensemble average of sample paths. Each path represents an independent realization of the process. The start of a sample path is determined by the initial condition that corresponds with the initial condition of the advection-diffusion equation. Therefore, the initial concentration distribution is transformed into an initial condition of the stochastic process. Discharge problems are easy to handle. Then the starting point of each sample path is located at the discharge position. To account for a continuous concentration distribution particles are released inside the flow region satisfying the concept of the numerically defined distribution - see section 4.3.1. With respect to the concept of point spread functions it is noted that the width of the point spread function depends on the simulation time. At initial state the width equals zero. As a consequence, the point spread function approaches a delta-function. A meaningful defined curve is obtained if particles are released inside a predefined grid; that is, the numerically defined distribution is used.

The initial concentration distribution, C_I, is regarded as a sum of step functions

$$C_I(\underline{x},t_0) = \sum_{i=1}^{N^c} \overline{C}(\underline{x}_i,t_0) S_i(\underline{x}) \qquad \text{with} \qquad S_i(\underline{x}) = \begin{cases} 1 & \underline{x} \in I_i \\ 0 & \underline{x} \notin I_i \end{cases} \tag{4.77}$$

with N^c the number of grid cells, introduced by the hydrodynamic model. $\overline{C}$ denotes the cell-averaged concentration value derived from the initial concentration distribution C_I. $\underline{x}_i$ is located at the centre of grid cell I_i. Once the volume V_i of all grid cells is computed the mass that is present in each grid cell is easily obtained from (4.77). The following relation holds

$$\overline{C}(\underline{x}_i,t_0) = \frac{1}{V_i} \sum_{p=1}^{n_p^0} \mu_p^0 \, \chi_{\{\underline{X}^p(t_0) \in I_i\}} \tag{4.78}$$

χ denotes the indicator function as given in equation (4.16) of section 4.2.2. Initialization proceeds by either uniform injection of particles or by injection of particles at the centre of the i^{th} grid cell. Both procedures are consistent with the numerically defined distribution. Uniform injection gives a more smooth discrete mass density. Therefore, uniform injection is preferred. It is reminded that particle positions are mainly computed in the transformed space. Therefore uniform injection has to be realized in the transformed space. To inject

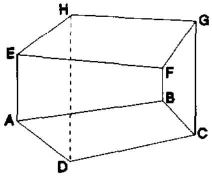


Figure (4.8): Hexahedral grid cell defined by eight corners (ABCD EFGH).

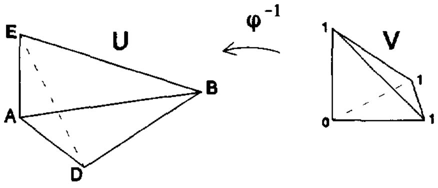


Figure (4.9): Linear mapping φ that maps arbitrary tetrahedron U into uniform tetrahedron V.

particles in the computational domain correctly each grid cell is regarded as the sum of 6 tetrahedra. The initial concentration distribution is considered as a sum of $6N^c$ step functions, similar as expressed in equation (4.77). The remainder of this section illustrates how to use this subdivision (i) to compute the grid cell volume, and (ii) to obtain a uniform mass density distribution in each part of the grid cell.

- Computation of the grid cell volume

To calculate the grid cell volume each grid cell is marked by eight corners (ABCD EFGH) - see figure (4.8). This gives rise to a decomposition into six tetrahedra (ABDE, EBDF, EDFH, CBDG, GBDF, GDFH). Since the coordinates of the corners are known, the edges of each tetrahedron are easily determined. The volume of an arbitrary tetrahedron U with corners ABDE is calculated as

$$\int_{\mathrm{U}} 1 \, d\underline{x} = V_{\mathrm{ABDE}} = \frac{1}{6}\left|\vec{AB}\cdot\left(\vec{AD}\times\vec{AE}\right)\right| \qquad (4.79)$$

$\left|\vec{AB}\cdot\left(\vec{AD}\times\vec{AE}\right)\right|$ denotes the Jacobian J of a linear transformation φ that maps ABDE into a uniform tetrahedron $\mathrm{V} = \{(\xi^1,\xi^2,\xi^3) \in \mathbf{R}^3 \mid 0<\xi^1<1,\ 0<\xi^2<1-\xi^1,\ 0<\xi^3<1-\xi^1-\xi^2\}$ - see figure (4.9). The volume of a grid cell is just the sum of the volumes of the individual tetrahedra.

- The transformed uniform mass distribution

Decomposition into 6 tetrahedra, as sketched above, provides that uniform injection becomes straightforward. If a random variable has a uniform mass density given by

$$p(\underline{x}) = \begin{cases} \dfrac{1}{\int_{U} 1\, d\underline{x}} & \underline{x} \in \mathrm{U} \\ 0 & \underline{x} \notin \mathrm{U} \end{cases} \tag{4.80}$$

then, by the change of variable technique, the mass density of the transformed random variable becomes

$$\tilde{p}(\underline{\xi}) = p(\underline{x}(\underline{\xi}))\,|J| = \begin{cases} \dfrac{1}{\int_{U} 1\, d\underline{x}}\,|J| = \dfrac{1}{\int_{V} 1\, d\underline{\xi}} & \underline{\xi} \in \mathrm{V} \\ 0 & \underline{\xi} \notin \mathrm{V} \end{cases} \tag{4.81}$$

Thus, as a result of the linear transformation φ a uniform distribution in the physical space implies a uniform distribution in the transformed space.

With respect to the numerical implementation uniform random numbers are selected to determine the initial position of particles in each tetrahedron. Figure (4.10) shows the result of uniform injection of 36 particles in a (two-dimensional) triangle.

Finally, four comments are made:

(i) The mass of each individual particle is determined by

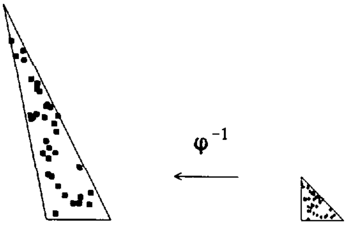


Figure (4.10): Uniform injection in the transformed space.

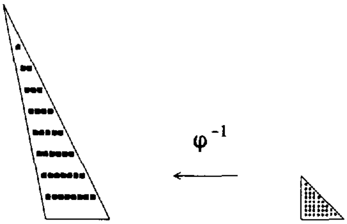


Figure (4.11): Equidistant injection in the transformed space (stretching phenomenon).

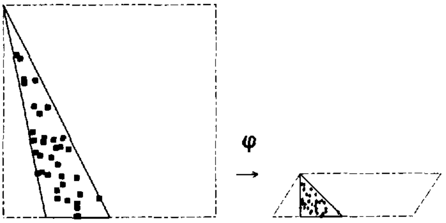


Figure (4.12): Uniform injection in envelope box in physical space.

$$\mu_p^0 = \frac{1}{N_{\mathrm{inj}}^{\mathrm{U}}} \left(\int_{\mathrm{U}} 1 dx \right) \left(\int_{\mathrm{U}} C_I(x) dx \right) \tag{4.82}$$

with $N_{\mathrm{inj}}^{\mathrm{U}}$ the number of injected particles in tetrahedron U. The total number of particles released at initial state equals

$$n_p^0 = \sum_{\mathrm{U}} N_{\mathrm{inj}}^{\mathrm{U}} \tag{4.83}$$

(ii) It is striking that an equidistant distribution in the transformed space is stretched in the physical space as shown in figure (4.11). This phenomenon seems to contradict the equivalence between the uniform density in the physical space and the uniform density in the transformed space. To account for this stretching phenomenon a square envelope box is introduced in the physical space. In three dimensions three random numbers are generated and scaled with the square box edge. If the so-obtained position belongs to the tetrahedron the particle is registered. Otherwise, the position is skipped. The procedure is illustrated for the triangle in figure (4.12).

(iii) The proposed decomposition is not unique. Therefore, the final result depends on the chosen decomposition.

(iv) In many practical applications the smoothing effect of the diffusion process provides that the error induced by the chosen injection procedure is negligible. Especially in cases where grid cells are approximately rectangular uniform injection can be applied directly in the entire grid cell. Then the simplest injection procedure is recommended.

4.3.4 The advection algorithm

The computation of particle displacements breaks up into two contributions: (i) the advective step, and (ii) the diffusive step - see equation (4.71) of section 4.3.2. This section focuses on an accurate approximation of the advective displacement. The transition from state $t = t_n$ to state $t = t_n + \Delta t$ is given by the solution of the following streamline equation, given the initial position of a particle at time $t = t_n$.

Cartesian coordinates	Transformed coordinates
$\begin{cases} \dfrac{dx}{dt} = u \\ \dfrac{dy}{dt} = v \\ \dfrac{dz}{dt} = w \end{cases} \qquad (4.84)$	$\begin{cases} \dfrac{d\xi}{dt} = \left(\tilde{h}_1\right)_U = U = \dfrac{1}{J}\left(H\sqrt{G_{\eta\eta}}\,u_\xi\right) \\ \dfrac{d\eta}{dt} = \left(\tilde{h}_2\right)_U = V = \dfrac{1}{J}\left(H\sqrt{G_{\xi\xi}}\,u_\eta\right) \\ \dfrac{d\sigma}{dt} = \left(\tilde{h}_3\right)_U = W = \dfrac{1}{J}\left(\sqrt{G}\,\omega\right) \end{cases} \qquad (4.85)$

The transformed case introduces the contravariant (velocity) components (U, V, W) - see appendix B. It is emphasized that equation (4.85) is also valid in case of a nonorthogonal horizontal transformation used in shallow water flow modelling.

To determine the magnitude of the advective displacement the (contravariant) velocity field needs to be integrated in time. This velocity field is usually not available at every point in space. As a result of a hydrodynamic simulation the flow information is given at discrete locations in a predefined grid. To solve equation (4.84) or (4.85) numerically several authors apply an interpolation procedure over the eight surrounding mesh points (trilinear interpolation) in combination with a time integration method (such as Euler or Runga Kutta methods). Examples are given in e.g., BUNING (1989) or HEEMINK (1990). In circulating flows the corresponding (numerically obtained) streamlines are not necessarily closed. These methods are able to cross streamlines, which is physically impossible. In SADARJOEN *et al.* (1994) it is stated that particle tracing algorithms in physical space generally perform better than algorithms in the computational space. Here a Lagrangian advection algorithm is considered for the approximation of (4.85) which violates this statement. Equation (4.84) is regarded as a special case. The algorithm is based upon the following assumptions:

(i) the flow field satisfies the continuity equation, and
(ii) the flow field is obtained by a conservative Eulerian method.

Assumption (i) is written in equational form as - see appendix A,

Cartesian coordinates	Transformed coordinates
$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (4.86)$	$\frac{1}{J}\left(\frac{\partial J}{\partial t} + \frac{\partial}{\partial \xi}(JU) + \frac{\partial}{\partial \eta}(JV) + \frac{\partial}{\partial \sigma}(JW)\right) = 0 \quad (4.87)$

The second assumption is provided by the discrete velocity field obtained by running a hydrodynamic model. This discrete velocity field satisfies the discretized continuity equation.

For these assumptions a numerical integration scheme is proposed for the solution of (4.85) fulfilling the condition that particles will always follow streamlines. Due to the strict condition of tracing a streamline computational problems, such as intersection of particle tracks with closed boundaries, will never occur.

The applicability of the algorithm comprises
- flow visualization (contour algorithm for constant values of the stream function),
- transport simulation (with respect to the advective displacement),
- determination of the advective contribution in the method of fractional steps - see YANENKO (1971). It is used to compute the substantial derivative D/Dt defined as the time rate of change along a streakline. The algorithm provides the position of the streakline.

The advection algorithm provides:
(i) the determination of the position of the streamline, and
(ii) an accurate approximation of the location of a particle travelling along the streamline for a period Δt.

The first item is sufficient for flow visualization purposes. In addition, the second item accounts for accurate particle tracking in transport simulation. With respect to the determination of the streamline the streamline equations will be written in parametric form. Subsequently, the introduced parameter is related to time such that the velocity along the streamline is accurately approximated. The three-dimensional treatment of the advection algorithm in shallow water applications will be preceded by the two-dimensional case considering Cartesian coordinates. The analogy with a contour algorithm is shown in the end of this section.

- Two-dimensional streamlines in Cartesian coordinates

In two dimensions the streamline equations will now be solved within one rectangular grid cell. Without loss of generality this grid cell covers $[0,\Delta x] \times [0,\Delta y]$. The equations are

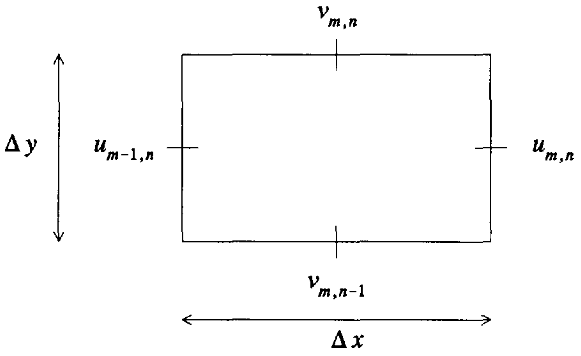


Figure (4.13): Two-dimensional staggered grid cell $[0,\Delta x] \times [0,\Delta y]$.

$$\frac{dx}{dt} = u \qquad\qquad \frac{dy}{dt} = v \tag{4.88}$$

In parametric form, introducing the parameter s, these equations become

$$\frac{dx}{ds} = \left(\frac{ds}{dt}\right)^{-1} u \qquad\qquad \frac{dy}{ds} = \left(\frac{ds}{dt}\right)^{-1} v \tag{4.89}$$

Similar to (4.86) the velocity field (u,v) satisfies the continuity equation (assumption (i))

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{4.90}$$

The discrete velocity field $(u_{m,n}, v_{m,n})$ satisfies the discretized continuity equation (assumption (ii)). A staggered grid is considered as shown in figure (4.13). A possible discretization may read

$$\frac{u_{m,n} - u_{m-1,n}}{\Delta x} + \frac{v_{m,n} - v_{m,n-1}}{\Delta y} = 0 \tag{4.91}$$

Subsequently, the streamline equations are written as

$$\frac{dx}{dt} = \frac{ds}{dt}\frac{dx}{ds} = \frac{ds}{dt}\left(\alpha_x x + \beta_x\right) \qquad \frac{dy}{dt} = \frac{ds}{dt}\frac{dy}{ds} = \frac{ds}{dt}\left(\alpha_y y + \beta_y\right) \tag{4.92}$$

Inserting $s(t) = t$ into equation (4.92) introduces a (continuous) pseudo velocity field $\left(\alpha_x x + \beta_x \,,\ \alpha_y y + \beta_y\right)$. This field preserves continuity if

$$\frac{\partial}{\partial x}\left(\alpha_x x + \beta_x\right) + \frac{\partial}{\partial y}\left(\alpha_y y + \beta_y\right) = \alpha_x + \alpha_y = 0 \tag{4.93}$$

Consequently, α_x and α_y are chosen such that equation (4.93) matches equation (4.91). For this special case the coefficients become:

$$\alpha_x = \frac{u_{m,n} - u_{m-1,n}}{\Delta x} \qquad \alpha_y = \frac{v_{m,n} - v_{m,n-1}}{\Delta y} \tag{4.94}$$

The choice of β_x and β_y is not essential. Their choice does not affect the property of continuity. However, a reasonable choice is obtained by matching the pseudo velocity field with the discrete velocity field given at the grid cell sides. Using $s(t) = t$:

$$\beta_x = u_{m-1,n} \qquad \beta_y = v_{m,n-1} \tag{4.95}$$

Doing so, the pseudo velocity field $\left(\alpha_x x + \beta_x \text{ , } \alpha_y y + \beta_y\right)$ is regarded as a continuous flow field. It satisfies the discrete continuity equation (zero-divergence) and resembles the discrete Eulerian flow field at the grid cell sides.

Resuming, the following equations remain to be solved within $[0,\Delta x] \times [0,\Delta y]$:

$$\begin{cases} \dfrac{dx}{ds} = \alpha_x x + \beta_x = \dfrac{u_{m,n} - u_{m-1,n}}{\Delta x}\, x + u_{m-1,n} \\[2ex] \dfrac{dy}{ds} = \alpha_y y + \beta_y = \dfrac{v_{m,n} - v_{m,n-1}}{\Delta y}\, y + v_{m,n-1} \end{cases} \tag{4.96}$$

This set of linear ordinary differential equations are solved analytically. The analytical solution becomes

$$x(s) = \begin{cases} \left(x_p^0 + \dfrac{\beta_x}{\alpha_x}\right)\exp\left(\alpha_x\, s\right) - \dfrac{\beta_x}{\alpha_x} & \alpha_x \neq 0 \\[2ex] \beta_x\, s + x_p^0 & \alpha_x = 0 \end{cases} \tag{4.97}$$

in which x_p^0 represents the x-coordinate of the initial position ($0 \leq x_p^0 \leq \Delta x$). A similar expression holds for $y(s)$. These expressions fix the position of the streamline. Since the constructed velocity field satisfies the continuity equation exactly at every point in space, the algorithm provides closed streamlines in (two-dimensional) circulating flows. The analytical solution of (4.96) provides that streamlines will inevitably bend away from a closed boundary with zero normal velocity component. The algorithm is demonstrated in section 4.4.1.

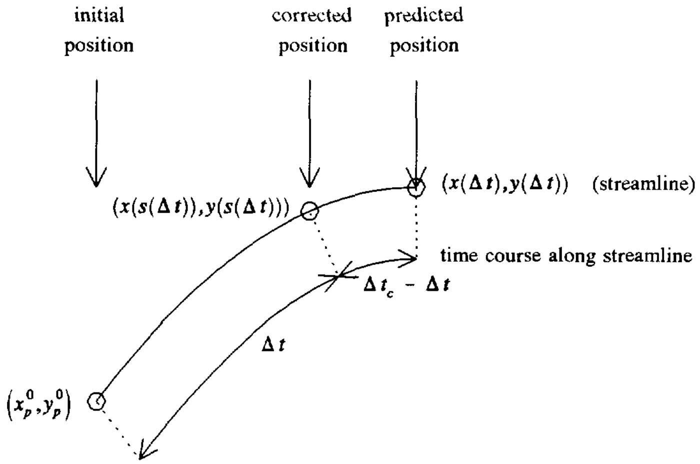

initial position
corrected position
predicted position
$(x(\Delta t),y(\Delta t))$ (streamline)
$(x(s(\Delta t)),y(s(\Delta t)))$
time course along streamline
$\Delta t_c - \Delta t$
Δt
(x_p^0,y_p^0)

Figure (4.14): Illustration of predictor-corrector Lagrangian advection solver. Δt_c denotes the corrected time increment needed to reach the predicted position, i.e., $s(\Delta t_c) = \Delta t$.

- Accurate particle tracking

If particle models are applied the velocity modulus along the streamline is of particular interest. Equation (4.96), with $s(t) = t$, gives only first order accurate approximations of the velocity components since lateral variations are disregarded. Higher order methods are constructed by considering suitable choices for the parameter function $s(t)$.

The algorithm is presented as a predictor-corrector method. In the predictor step the solution of (4.96) is considered setting $s(t) = t$. The particle traces the streamline travelling from (x_p^0,y_p^0) to its predicted position (x_p^{pred},y_p^{pred}). The predicted position is obtained from equation (4.97): $x_p^{pred} = x(\Delta t)$. Similarly, $y_p^{pred} = y(\Delta t)$. Setting $s = s(t)$ instead of$s = t$ the corrected end point is given by $(x(s(\Delta t)),y(s(\Delta t)))$. The duration Δt used in the predictor step is just an approximation of the required period needed to reach the predicted position. Due to inaccurate approximation of the velocity modulus along the streamline it is possible that the magnitude of the displacement is overestimated or underestimated. The corrector step gives rise to a corrected time increment to reach the predicted position. This corrected time increment will be denoted with Δt_c, where $s(\Delta t_c) = \Delta t$. The procedure is illustrated in figure (4.14). The remaining question involves the determination of an appropriate relation for $s(t)$.

An ordinary differential equation for $s(t)$ is derived from

$$\begin{cases} u = \dfrac{dx}{dt} = \dfrac{ds}{dt}\left(\alpha_x x + \beta_x\right) \\ v = \dfrac{dy}{dt} = \dfrac{ds}{dt}\left(\alpha_y y + \beta_y\right) \end{cases} \tag{4.98}$$

The left-hand side of equation (4.98) can be approximated with any consistent interpolation procedure, using the flow field that was obtained by a conservative Eulerian method. Such an approximation of the velocity modulus at $\left(x_p^0, y_p^0\right)$ and $(x(\Delta t), y(\Delta t))$ is denoted with $\sqrt{u^2 + v^2}\big|_{init}$ and $\sqrt{u^2 + v^2}\big|_{pred}$ respectively. Using equation (4.98), ds/dt is approximated at initial and predicted position:

$$\left.\frac{ds}{dt}\right|_{init} = \frac{\sqrt{u^2 + v^2}\big|_{init}}{\left(\left(\alpha_x x_p^0 + \beta_x\right)^2 + \left(\alpha_y y_p^0 + \beta_y\right)^2\right)^{1/2}} \tag{4.99}$$

and

$$\left.\frac{ds}{dt}\right|_{pred} = \frac{\sqrt{u^2 + v^2}\big|_{pred}}{\left(\left(\alpha_x x(\Delta t) + \beta_x\right)^2 + \left(\alpha_y y(\Delta t) + \beta_y\right)^2\right)^{1/2}} \tag{4.100}$$

The desired differential equation for $s(t)$ is obtained from an assumption about the time rate of change of the velocity modulus while moving along the streamline. Various procedures are possible. The simplest choice is possibly inspired by the mid-point rule:

$$\begin{cases} \dfrac{ds}{dt} = \tfrac{1}{2}\left(\left.\dfrac{ds}{dt}\right|_{init} + \left.\dfrac{ds}{dt}\right|_{pred}\right) \\ s(0) = 0 \end{cases} \tag{4.101}$$

If a linear time change in the velocity modulus is assumed the following equation seems natural (first two terms of Taylor series expansion)

$$\begin{cases} \dfrac{ds}{dt} = \left.\dfrac{ds}{dt}\right|_{init} + t\,\dfrac{\left.\dfrac{ds}{dt}\right|_{pred} - \left.\dfrac{ds}{dt}\right|_{init}}{\Delta t_c} \\ s(0) = 0 \end{cases} \tag{4.102}$$

The procedure is summarized as follows:

Predictor step:

determination of track

(1) Construct a pseudo velocity field and corresponding streamline equations similar to (4.96).

(2) Solve the streamline equations analytically (equation (4.97)) to determine the predicted particle location $(x(\Delta t), y(\Delta t))$.

Corrector step:

determination of position along track

(3) Approximate the velocity modulus at initial and predicted location.

(4) Compute relation (4.99) and relation (4.100).

(5) Construct a suitable differential equation for $s(t)$ as in (4.101) or (4.102), and solve this equation analytically. Finally, the corrected position of the end point of a particle tracing the streamline for a period Δt is given by $(x(s(\Delta t)), y(s(\Delta t)))$.

For de sake of completeness, the solutions of (4.101) and (4.102) are reflected,

$$\text{for (4.101):}\quad s(t) = \tfrac{1}{2}\left(\left.\frac{ds}{dt}\right|_{init} + \left.\frac{ds}{dt}\right|_{pred} \right) t \tag{4.103a}$$

$$\text{for (4.102):}\quad \begin{cases} s(t) = \left.\dfrac{ds}{dt}\right|_{init} t + \tfrac{1}{2}\left(\left.\dfrac{ds}{dt}\right|_{pred} - \left.\dfrac{ds}{dt}\right|_{init} \right) \dfrac{t^2}{\Delta t_c} \\ \Delta t_c = 2\Delta t \left(\left.\dfrac{ds}{dt}\right|_{pred} + \left.\dfrac{ds}{dt}\right|_{init} \right)^{-1} \end{cases} \tag{4.103b}$$

- Extension to the transformed case

The extension to a streamline algorithm in transformed coordinates runs similar as in Cartesian coordinates. The predictor step now accounts for the streamline equations given in (4.85). A continuous pseudo contravariant velocity field is constructed such that its divergence resembles the discretized transformed continuity equation. The corrector step becomes more complex since the velocity modulus at initial and predicted location must be approximated using nonorthogonal contravariant velocity components. Nonorthogonality restricts immediate applicability of the Euclidean norm. Other interpolation techniques that account for the direction of the covariant base vectors are necessary. As soon as$ds/dt|_{init}$ and $ds/dt|_{pred}$ are accurately estimated, the algorithm is again in accordance with the Cartesian version sketched above. Looking at equation (4.85), the Jacobian of the transformation arises as J^{-1}. This observation gives rise to an alternative description of the differential equation for $s(t)$. For instance, in nearly uniform flow it is expected that the numerator (interrelated with the flow rate), arising in the right-hand side of (4.85), hardly changes while moving along the streamline. In extreme nonorthogonal grids it is reasonable to assume that the Jacobian changes linearly in time while moving along the streamline. This is envisaged in the alternative choice

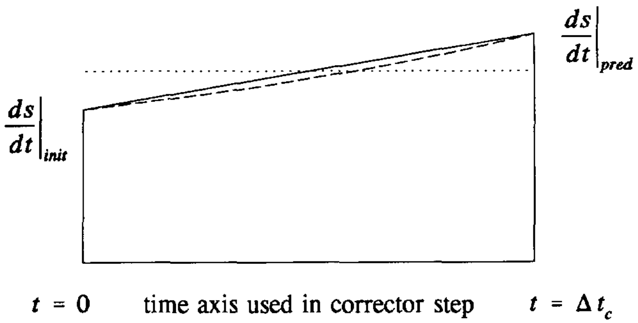

$\left.\frac{ds}{dt}\right|_{pred}$
$\left.\frac{ds}{dt}\right|_{init}$
$t = 0$
time axis used in corrector step
$t = \Delta t_c$

Figure (4.15): Ratio between (i) accurate approximation of velocity modulus and (ii) approximation of velocity modulus used in predictor step sketched as a function of time.

$$\frac{ds}{dt}\begin{cases}\text{dotted line:} & \text{equation (4.101),}\\ \text{straight line:} & \text{equation (4.102),}\\ \text{broken line:} & \text{equation (4.104).}\end{cases}$$

$$\begin{cases} \dfrac{ds}{dt} = \dfrac{\Delta t_c}{(\Delta t_c - t)\left.\dfrac{ds}{dt}\right|_{init}^{-1} + t\left.\dfrac{ds}{dt}\right|_{pred}^{-1}} \\ s(0) = 0 \end{cases} \tag{4.104}$$

Here $ds/dt|_{init}$ and $ds/dt|_{pred}$ account for accurate approximations of the Jacobian at the initial and predicted location respectively. If $ds/dt|_{init} = ds/dt|_{pred}$ then the solution of (4.104) resembles the solution of (4.101). If $ds/dt|_{init} \neq ds/dt|_{pred}$ the solution of (4.104) becomes

$$\begin{cases} s(t) = \dfrac{\Delta t_c}{\left.\dfrac{ds}{dt}\right|_{pred}^{-1} - \left.\dfrac{ds}{dt}\right|_{init}^{-1}} \ln\left[1 + \left.\dfrac{ds}{dt}\right|_{init}\left(\left.\dfrac{ds}{dt}\right|_{pred}^{-1} - \left.\dfrac{ds}{dt}\right|_{init}^{-1}\right)\dfrac{t}{\Delta t_c}\right] \\ \Delta t_c = \left(\left.\dfrac{ds}{dt}\right|_{pred}^{-1} - \left.\dfrac{ds}{dt}\right|_{init}^{-1}\right)\left(\ln\left[\left.\dfrac{ds}{dt}\right|_{init}\right] - \ln\left[\left.\dfrac{ds}{dt}\right|_{pred}\right]\right)^{-1}\Delta t \end{cases} \tag{4.105}$$

Although it is observed in figure (4.15) that the choice of ds/dt does not really lead to extreme differences it is recommended to apply (4.102) in approximate orthogonal grids, while (4.104) is useful in irregular grids with space-varying Jacobian.

- The Lagrangian advection solver for three-dimensional shallow water flow

The Lagrangian advection solver will now be applied considering a nonstationary Eulerian flow field obtained with the hydrodynamic model of section 2.5. In nonstationary flows the advection solver and the hydrodynamic model are applied simultaneously with equal time increment Δt. Since the hydrodynamic model uses a time splitting scheme, the advection algorithm will also be presented as a time splitting scheme. In the first step (marching from t to $t+\frac{1}{2}\Delta t$) as well as in the second step (marching from $t+\frac{1}{2}\Delta t$ to $t+\Delta t$) the complete predictor-corrector advection scheme is applied. In time interval $(t, t+\frac{1}{2}\Delta t)$ the advection solver accounts for the discretized continuity equation as given in equation (2.29) of section 2.5. During $(t+\frac{1}{2}\Delta t, t+\Delta t)$ equation (2.30) is considered. For convenience, the procedure is illustrated within a grid cell (m,n,k) with dimensions $[0,1] \times [0,1] \times \left[\sigma_k^{\text{bot}}, \sigma_k^{\text{top}}\right]$. σ_k^{top} denotes the σ - coordinate at the $\omega_{m,n,k-\frac{1}{2}}$ - location and $\sigma_k^{\text{bot}} = \sigma_k^{\text{top}} - \Delta\sigma_k$. The equations that need to be solved are given by (compare with equation (4.92))

$$\left\{\begin{aligned} \frac{d\xi}{dt} &= \frac{ds}{dt}\frac{d\xi}{ds} = \frac{ds}{dt}U_p = \frac{ds}{dt}\frac{Q_\xi}{\bar{J}} \\ \frac{d\eta}{dt} &= \frac{ds}{dt}\frac{d\eta}{ds} = \frac{ds}{dt}V_p = \frac{ds}{dt}\frac{Q_\eta}{\bar{J}} \\ \frac{d\sigma}{dt} &= \frac{ds}{dt}\frac{d\sigma}{ds} = \frac{ds}{dt}W_p = \frac{ds}{dt}\frac{Q_\sigma}{\bar{J}} \end{aligned}\right. \tag{4.106}$$

(U_p, V_p, W_p) is regarded as a continuous approximation of the discrete contravariant velocity field (providing $s(t) = t$). The Jacobian $\bar{J}$ and the flow rates Q_ξ, Q_η and Q_σ are introduced to meet the expressions given in equation (4.85). The corresponding divergence equals

$$\begin{aligned} &\frac{1}{\bar{J}}\left(\frac{\partial\bar{J}}{\partial t} + \frac{\partial}{\partial\xi}\left(\bar{J}\frac{ds}{dt}U_p\right) + \frac{\partial}{\partial\eta}\left(\bar{J}\frac{ds}{dt}V_p\right) + \frac{\partial}{\partial\sigma}\left(\bar{J}\frac{ds}{dt}W_p\right)\right) \\ &= \frac{ds}{dt}\frac{1}{\bar{J}}\left(\frac{\partial\bar{J}}{\partial s} + \frac{\partial}{\partial\xi}(Q_\xi) + \frac{\partial}{\partial\eta}(Q_\eta) + \frac{\partial}{\partial\sigma}(Q_\sigma)\right) \end{aligned} \tag{4.107}$$

If

$$\bar{J}(s) = \left(H^l_{m,n} + \left(H^*_{m,n} - H^l_{m,n}\right)\frac{s}{\frac{1}{2}\Delta t}\right)\sqrt{G}\Big|_{m,n} \tag{4.108a}$$

$$Q_\xi(\xi) = \left(Q_\xi^E - Q_\xi^W\right)\xi + Q_\xi^W \tag{4.108b}$$

$$Q_\xi^E = \left(d\sqrt{G_{\eta\eta}}u_\xi\right)^*_{m+\frac{1}{2},n,k} + \zeta^*_{m,n}\left(\sqrt{G_{\eta\eta}}u_\xi\right)^*_{m+\frac{1}{2},n,k} + \left(\sqrt{G_{\eta\eta}}u_\xi\right)^*_{m,n,k}\zeta^*_{m+\frac{1}{2},n,k}$$

$$Q_\xi^W = \left(d\sqrt{G_{\eta\eta}}u_\xi\right)^*_{m-\frac{1}{2},n,k} + \zeta^*_{m,n}\left(\sqrt{G_{\eta\eta}}u_\xi\right)^*_{m-\frac{1}{2},n,k} + \left(\sqrt{G_{\eta\eta}}u_\xi\right)^*_{m,n,k}\zeta^*_{m-\frac{1}{2},n,k}$$

$$Q_\eta(\eta) = \left[\left(H\sqrt{G_{\xi\xi}}u_\eta\right)^l_{m,n+½,k} - \left(H\sqrt{G_{\xi\xi}}u_\eta\right)^l_{m,n-½,k}\right]\eta + \left(H\sqrt{G_{\xi\xi}}u_\eta\right)^l_{m,n-½,k} \tag{4.108c}$$

$$Q_\sigma(\sigma) = \sqrt{G}\Big|_{m,n} \frac{\left(\overset{*}{\omega}_{m,n,k-½} - \overset{*}{\omega}_{m,n,k+½}\right)\sigma + \overset{*}{\omega}_{m,n,k+½}\sigma_k^{\text{top}} - \overset{*}{\omega}_{m,n,k-½}\sigma_k^{\text{bot}}}{\Delta\sigma_k} \tag{4.108d}$$

then the discretized continuity equation (2.29) provides that the constructed pseudo velocity field, introduced in (4.106), also preserves continuity. This is observed by substitution of (4.108) into (4.107). The result equals the continuity equation (2.29) multiplied by $(ds/dt)\sqrt{G}\big|_{m,n}/\bar{J}$. It is noted that the flow rate contributions Q_ξ, Q_η and Q_σ are chosen such that they resemble the flow rate contributions as computed by the hydrodynamic model at grid cell sides. For instance, $Q_\sigma\left(\sigma_k^{\text{top}}\right) = \sqrt{G}\big|_{m,n}\overset{*}{\omega}_{m,n,k-½}$.

Substitution of (4.108) into (4.106) yields a set of linear ordinary differential equations. Setting $s(t) = t$, this set can be solved analytically. The general form of the arising equations becomes, with $\tau = ½\Delta t$,

$$\begin{cases} \dfrac{d\xi}{ds} = \dfrac{\alpha_\xi\xi + \beta_\xi}{J_0 + \left(J_1 - J_0\right)\dfrac{s}{\tau}} \\ \xi(0) = \xi_p^0 \end{cases} \tag{4.109}$$

The corresponding solution equals

$$\xi(s) = \begin{cases} \xi_p^0 + \dfrac{\beta_\xi}{J_0}s & J_0 = J_1 \quad \alpha_\xi = 0 \\ \left(\xi_p^0 + \dfrac{\beta_\xi}{\alpha_\xi}\right)\exp\left[\dfrac{\alpha_\xi}{J_0}s\right] - \dfrac{\beta_\xi}{\alpha_\xi} & J_0 = J_1 \quad \alpha_\xi \neq 0 \\ \xi_p^0 + \dfrac{\beta_\xi\tau}{J_1 - J_0}\ln\left[1 + \dfrac{J_1 - J_0}{J_0}\dfrac{s}{\tau}\right] & J_0 \neq J_1 \quad \alpha_\xi = 0 \\ \left(\xi_p^0 + \dfrac{\beta_\xi}{\alpha_\xi}\right)\left[1 + \dfrac{J_1 - J_0}{J_0}\dfrac{s}{\tau}\right]^{\frac{\alpha_\xi\tau}{J_1 - J_0}} - \dfrac{\beta_\xi}{\alpha_\xi} & J_0 \neq J_1 \quad \alpha_\xi \neq 0 \end{cases} \tag{4.110}$$

Setting $s(t) = t$, the predicted location becomes $\left(\xi^{pred},\eta^{pred},\sigma^{pred}\right) = \left(\xi(\tau),\eta(\tau),\sigma(\tau)\right)$. With this the predictor step is completed. A particle has suffered an advective displacement in the period $(t, t+\tau)$. This displacement is computed in correspondence with the numerical treatment of the hydrodynamic model that solves the nonstationary shallow water flow. The corrector step is useful only if the velocity moduli at initial and predicted position can be approximated accurately. For that reason it is stated that for general engineering purposes the predictor step is sufficient (providing $s(t) = t$). However, application of the corrector step is still feasible. Then, similar as in (4.99) and (4.100), $ds/dt|_{init}$ and $ds/dt|_{pred}$ must be approximated and inserted into, for instance, equation (4.102). The corrected position becomes $(\xi(s(\tau)),\eta(s(\tau)),\sigma(s(\tau)))$. With this the first step of the time splitting procedure is finished. The advection solver for the second step (marching from $t+\tau$ to $t+\Delta t$) is constructed similarly.

- Two-dimensional contour algorithm for constant values of the stream function

It will now be shown that the predictor (-corrector) algorithm is also applicable as a contour algorithm. A streamline written in parametric form $(x(s),y(s))$ is determined by the relation of constant values of the stream function:

$$\psi(x(s),y(s)) = \text{constant} \tag{4.111}$$

Again it is assumed that only discrete flow information is available - see figure (4.16). To solve $(x(s),y(s))$ form (4.111), $\psi(x,y)$ can be approximated using a bilinear interpolation. With $\Delta x = \Delta y = 1$,

$$\psi(x,y) = xy\psi_4 + x(1-y)\psi_2 + (1-x)y\psi_3 + (1-x)(1-y)\psi_1 \tag{4.112}$$

Differentiation of equation (4.111) with respect to s yields

$$\frac{\partial\psi}{\partial x}\frac{dx}{ds} + \frac{\partial\psi}{\partial y}\frac{dy}{ds} = 0 \tag{4.113}$$

In discrete form equation (4.113) is satisfied if

$$\left\{\begin{aligned} \frac{dx}{ds} &= \frac{\partial\psi}{\partial y} = x(\psi_4-\psi_2) + (1-x)(\psi_3-\psi_1) \\ \frac{dy}{ds} &= -\frac{\partial\psi}{\partial x} = -[y(\psi_4-\psi_3) + (1-y)(\psi_2-\psi_1)] \end{aligned}\right. \tag{4.114}$$

With this the analogy with the earlier derived advection algorithm is achieved by:

(i) equation (4.114) ↔ equation (4.96), and
(ii) equation (4.113) ↔ continuity equation.

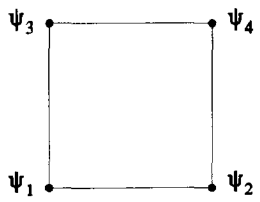


Figure (4.16): Grid cell with discrete information of stream function ($\Delta x = \Delta y = 1$).

The Lagrangian advection algorithm will be illustrated by means of numerical experiments. These experiments are moved to section 4.4.1.

4.3.5 Implementation of exponential decay and source terms

In view of the backward approach discussed in section 4.2.2, the concentration value is approximated by contributions of sample paths staying inside the flow region and by contributions of sample paths reaching the boundary. In the forward approach the origin of each particle track is determined by either the initial concentration distribution, the source terms, or the boundary conditions. The boundary treatment will be discussed in the next section. Realization of the initial concentration distribution is already examined in section 4.3.1. The implementation of source terms evolves similarly. For each time interval(t_n, t_{n+1}) the source term S is evaluated. The corresponding amount of mass to be injected at time $t = t_n$ in cell I_i equals

$$F_S^n(I_i) = \int_{t_n}^{t_{n+1}} \int_{I_i} S(\underline{x}, s)\, d\underline{x} ds \tag{4.115}$$

Equation (4.115) is approximated with any consistent numerical integration procedure. The number of particles that are injected at time $t = t_n$ in cell I_i simply equals $n_p^{inj} = \left[F_S^n(I_i)/\mu_p^0\right] + 1$, with $[\cdot]$ the truncated integer of the argument. Each injected particle carries a mass $\mu_p^n = F_S^n(I_i)/n_p^{inj}$.

Exponential decay is recognized by observing

$$\frac{\partial C}{\partial t} = -\gamma C \tag{4.116}$$

which yields for $t > t_n$

$$C(\underline{x},t) = C(\underline{x},t_n)\exp\left[-\int_{t_n}^{t} \gamma(\underline{x},s)\, ds\right] \tag{4.117}$$

The (exponential) scaling factor is also observed in the backward approach - see equation (4.15) of section 4.2.2. The decay represents a property of the element the particle is carrying. It does not affect the particle trajectory. So, in addition to the particle displacement the evolution of the mass of a particle is included. As a consequence of (4.117), the mass of a particle at time $t = t_{n+1}$, μ_p^{n+1}, becomes

$$\mu_p^{n+1} = \mu_p^n \exp\left[-\int_{t_n}^{t_{n+1}} \gamma(\underline{X}^p(s),s)\, ds\right] \tag{4.118}$$

A first order approximation is obtained by

$$\mu_p^{n+1} = \left(1 - \gamma^n\left(\underline{X}^p(t_n)\right)\Delta t\right)\mu_p^n \tag{4.119}$$

It is emphasized that the loss of mass depends on the particle trajectories.

4.3.6 Implementation of boundary conditions

In section 4.2.2 the treatment of boundary conditions is discussed for particle models. The implementation of this procedure will be given in detail applied to the so-called Robbin boundary condition. This type of boundary condition includes the Dirichlet boundary condition which prescribes a fixed concentration at the boundary. Also the Neumann boundary condition is covered. In most existing particle models the treatment of particles near the boundary is controlled by an absorbtion/reflection procedure. It appears that total absorbtion is consistent with the zero concentration boundary condition, while total reflection is consistent with a zero flux boundary condition. This section presents some comments about alternative reflection approaches focusing on anisotropic media.

To illustrate the Robbin boundary condition the one-dimensional case is considered

$$R_1(z,t)\frac{\partial C}{\partial z}\bigg|_{z\in\partial\Omega} + R_2(z,t)C\big|_{z\in\partial\Omega} = R_3(z,t)\big|_{z\in\partial\Omega} \tag{4.120}$$

with z the vertical coordinate and $\partial\Omega$ the boundary of the (computational) flow region. This type of boundary condition occurs, for instance, near the bottom in sediment transport models. It is sometimes argued that the concentration near the bottom is determined by the local flow conditions introducing an equilibrium bed concentration C_e - see WANG (1989). This is expressed by the Dirichlet boundary condition

$$C|_{z\in\text{bottom}} = C_e \tag{4.121}$$

Another example is to prescribe the upward sediment flux in terms of the local conditions

$$D_z \frac{\partial C}{\partial z}\bigg|_{z\in\text{bottom}} = -w_s C_e \tag{4.122}$$

with D_z the vertical diffusion coefficient and w_s the settling velocity of the sediment. (4.121) and (4.122) are special cases of condition (4.120).

- Particle injection controlled by the estimated mass flux through the boundary.

The number of particles to be injected at the boundary is approximated by estimation of the mass flux per time step through the boundary. This estimation will be used to correct the amount of mass that floats out of the domain as a consequence of the transport process. For discrete times the procedure becomes:

(i) Estimate the mass flux per time step through the grid cell boundaries that contact the physical boundary. The expression given in (4.18) of section 4.2.2 represents the sum of all these contributions. Each contribution is denoted with $F_{\partial\Omega}^{n+1}(I_i)$, in which I_i represents the i^{th} - grid cell that contacts the boundary ($\partial\Omega \cap \partial I_i \neq \emptyset$),

$$F_{\partial\Omega}^{n+1}(I_i) = \tfrac{1}{2}\Delta t\left(\underline{q}^n\cdot\underline{n}\;\Delta A^n + \underline{q}^{n+1}\cdot\underline{n}\;\Delta A^{n+1}\right)\Big|_{\underline{x}=\underline{r}} \tag{4.123a}$$

$\underline{r}$ represents the centre point of $\partial\Omega \cap \partial I_i$. ΔA^n denotes the area of $\partial\Omega \cap \partial I_i$ at time $t = t_n$. In the absence of decay $\underline{q}^{n+1}$ is set equal to $\underline{q}^n$. A first order explicit time integration procedure results in

$$F_{\partial\Omega}^{n+1}(I_i) = \Delta t\left(\underline{q}^n\cdot\underline{n}\right)\Big|_{\underline{x}=\underline{r}}\;\Delta A^n \tag{4.123b}$$

$\underline{q}^n$ at the boundary is obtained using the prescribed boundary condition and the simulated concentration distribution at time $t = t_n$ - see equation (4.19) of section 4.2.2. So, the local flow velocity, the diffusion coefficients and a (continuous) representation of the concentration distribution should be available at every time step in the vicinity of the boundary. Here it is assumed that u_i and D_{ij} are determined by running a hydrodynamic model which solves this flow field. The concentration and its derivative must be estimated using the information about the particle positions and the prescribed boundary condition. So, in case of a Neumann boundary condition the concentration derivative is prescribed and only the concentration at the boundary must be estimated. In case of a Dirichlet boundary condition the concentration derivative must be estimated. Other type of boundary conditions are written in a form such that either the concentration or its derivative is eliminated and the remaining quantity needs to be estimated. For instance, the Robbin boundary condition can be written such that the concentration derivative at the boundary is a function of concentration. The obtained expression is then substituted in (4.123).

For the approximation of q^{n+1} in the presence of decay equation (4.117) of section 4.2.5 is taken into account. The numerical experiments of section 4.4 will give a further illustration.

(ii) Record the amount of mass, carried by the particles, that cross the boundary $\partial\Omega$. These particles are absorbed and therefore excluded for further computations. The total absorbed mass during time interval (t_n, t_{n+1}) is denoted with $\overline{F}_{\partial\Omega}^{n+1}(I_i)$. It is obtained by

$$\overline{F}_{\partial\Omega}^{n+1}(I_i) = \sum_{p=1}^{n_p} \left[\left(1 - \delta^p\right) \mu_p^n + \delta^p \mu_p^{n+1} \right] \chi_{\{t_{n+1} \geq \tau \,\wedge\, X^p(\tau) \,\in\, \partial\Omega \,\cap\, \partial I_i\}} \qquad (4.124)$$

with χ the indicator function defined in equation (4.16) of section 4.2.2. δ^p equals

$$\delta^p = \frac{\mathrm{d}\left(\underline{X}^p(t_n), \underline{X}^p(\tau)\right)}{\mathrm{d}\left(\underline{X}^p(t_n), \underline{X}^p(t_{n+1})\right)} \qquad (4.125)$$

$\mathrm{d}(\cdot,\cdot)$ defines the distance between both arguments measured along the particle track.

(iii) Inject a finite number of particles in cell I_i with total mass equal to $\overline{F}_{\partial\Omega}^{n+1}(I_i) - F_{\partial\Omega}^{n+1}(I_i)$. Thus the boundary is regarded as a point source. The intensity of this source is controlled by the difference in the estimated ($F_{\partial\Omega}$) and simulated ($\overline{F}_{\partial\Omega}$) mass outflow.

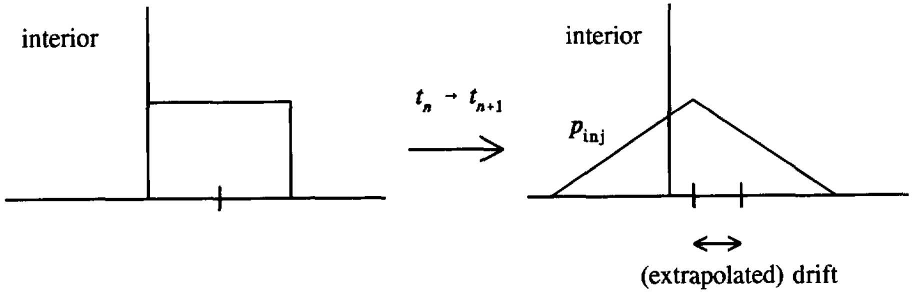


Figure (4.17): Probability density function of injected particle position at an open boundary.

DURBIN (1983) argues that particles should be returned into the interior where its position should be determined by a random variable chosen from a distribution, which is the solution of the Fokker-Planck equation with linearized coefficients. It is based on the assumption that the Fokker-Planck equation is also valid near the boundary. This idea is used to derive a simple distribution that determines the position of the injected particle. Suppose that a mass M has to be injected at the boundary. $n_p^{\text{inj}} = \left[M/\mu_p^0\right] + 1$ particles are released each carrying a mass $\mu_p^{\text{inj}} = M/n_p^{\text{inj}}$, with $[\ \cdot\ ]$ the truncated integer of the argument. The position of the injected particle in the interior is determined by a distribution p_{inj}. The distribution p_{inj} is obtained by assuming a uniform mass distribution boarding the boundary just outside the interior domain. Furthermore, it is assumed that particle displacements of particles outside the computational domain can be computed by extrapolation using flow data of the interior. In a single time step a uniform distribution deforms into a triangular distribution as shown in figure (4.17). Since a uniform distribution is chosen for transport simulation - see equation (4.61) of section 4.3.2, the part of the triangular distribution that overlaps the interior is used for p_{inj}. With this injection algorithm the estimated mass flux procedure is completed.

During simulations it is possible that the estimated ($F_{\partial\Omega}$) and simulated ($\bar{F}_{\partial\Omega}$) mass flux per time step through the boundary is such that M becomes negative. It expresses that in the vicinity of the boundary at the previous time step the particle density was too low to satisfy the correct mass outflow condition. So, the original problem is either ill-posed or the estimated mass flux, given in equation (4.123), is overestimated due to inaccurate simulation. All the executed numerical experiments in which M became negative showed that this was caused by inaccurate approximation of the concentration derivative at the boundary. A correct approximation of $F_{\partial\Omega}$ can only be achieved if a sufficient number of particles is used in the simulation.

Special types of boundary conditions can also be implemented by the appropriate rule of absorbtion or reflection of particles. Subsequently, an alternative treatment of special boundary conditions involving total absorbtion and total reflection will be discussed.

- Particle absorbtion

Many particle transport simulation models apply a far field concentration assumption at open boundaries. This assumption implies that the open boundaries are positioned far away from the initial cloud of particles. If a particle reaches the open boundary the particle is absorbed (i.e., removed from further computations). The prescription of an absorbtion of particles at the open boundary is consistent with a zero concentration boundary condition. This can be proved in various ways - see e.g., TAIRA (1988), DURBIN (1983), CSANADY (1973) or CHANDRASEKHAR (1943). For this type of boundary condition the absorbtion procedure is far more accurate than the estimated mass flux procedure described above.

- Particle reflection

At closed boundaries it is often assumed that transfer of mass through such a boundary is physically impossible. This assumption is consistent with the zero flux boundary condition. Due to the discrete numerical procedure, particles may contact closed boundaries. To guarantee zero flux all particles that reach the boundary are reflected. Numerical implementation needs the prescription of the exact reflection procedure. From a physical point of view one would expect perpendicular reflection against a closed boundary (in correspondence with Snellius' law). For complicated closed boundaries the computation of the position of the reflected image point becomes rather awkward. However, the use of an orthogonal transformed computational grid simplifies the computation considerably. In nonorthogonal grids, where the covariant base vectors vary throughout the grid cell, the computation of the image point remains cumbersome.

To avoid a complex reflection procedure alternative methods have been studied by FLIERT (1988). Three methods were compared for one-dimensional applications. To determine the end point of the random displacement random numbers are selected. If the stochastic displacement is such that the particle track crosses the closed boundary then either

(i) the particle will suffer (perpendicular) reflection (bounces back),
(ii) the new position of the particle is determined by a repeated trial providing that the particle stays inside the domain, or
(iii) the particle remains at its original position.

The corresponding probability density functions are sketched in figure (4.18). A well

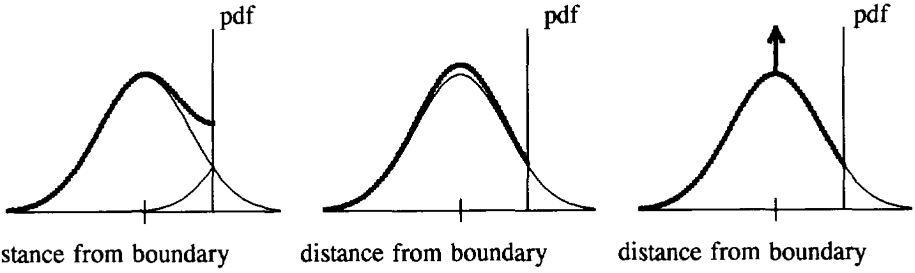


Figure (4.18): Probability density function (pdf) of particle position after reflection.

LEFT:	perpendicular reflection - procedure (i),
CENTRE:	repeated trial - procedure (ii),
RIGHT:	frozen position - procedure (iii).

comparison between the investigated procedures is established by studying the asymptotic behaviour of a stationary concentration distribution. For diffusion processes in the absence of advection the stationary concentration distribution is uniform. It was concluded that procedure (i) yields the correct asymptotic behaviour. Procedure (ii) results in a stationary solution which is not uniform near the closed boundary. The width of this nonuniform region depends on the chosen time step. Procedure (iii) gives the correct asymptotic behaviour, although it reduces the intensity of the diffusion coefficient near the boundary. Still, the first reflection procedure remains preferable.

In three dimensions perpendicular reflection is invariant for orthogonal coordinate transformations. In nonorthogonal applications the numerical result of perpendicular reflection in the physical space does not always coincide with perpendicular reflection in the transformed space - see figure (4.19). From a computational point of view reflection in the transformed space is preferable. With respect to the assumptions made in the hydrodynamic model both methods are doubtful. This will be illustrated by considering anisotropic media.

In shallow water flow problems different horizontal and vertical diffusion coefficients are distinguished. To illustrate the deficiency of perpendicular reflection procedures in an anisotropic medium the ultimate case is assumed: the absence of vertical diffusion. The principal axes of diffusion are assumed to coincide with the Cartesian coordinate axes. The off-diagonal elements of the diffusion tensor equal zero. Then, particles are restricted to move in a horizontal plane. A particle that hits an oblique bottom will experience a vertical displacement in case of perpendicular reflection - see figure (4.19). To overcome this inconsistency an alternative reflection principle is proposed which deals with the orientation of the principal axes of diffusion. If $\underline{X}(\Delta t)$ reaches outside the flow region the particle must

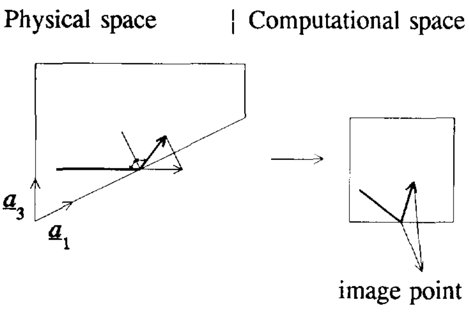


Figure (4.19a): Perpendicular reflection in physical space.

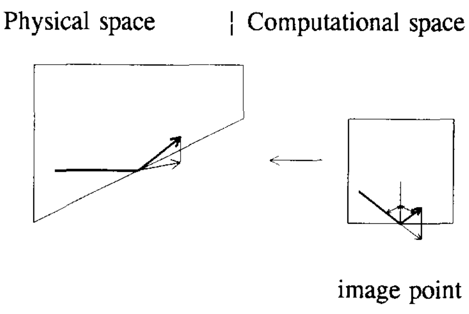


Figure (4.19b): Perpendicular reflection in computational space.

suffer reflection. Traditional perpendicular reflection deals with the normal vector $\underline{n}$ to the closed boundary. Given $\underline{n}$ and line segment $(\underline{X}(0), \underline{X}(\Delta t))$ the image point is determined using classical algebra - see figure (4.20). The alternative reflection principle uses $D^T\underline{n}$ instead of $\underline{n}$. A two-dimensional illustration is given in figure (4.21a) where $D = \mathbf{diag}\,(D_H, D_V)$ with $D_V = 0$, i.e., the absence of vertical diffusion. Consequently, a particle moving in the horizontal plane will remain in this horizontal plane even after reflection. More details about this reflection procedure are described in appendix E. A disadvantage of this procedure is that the track of a particle that suffers reflection is

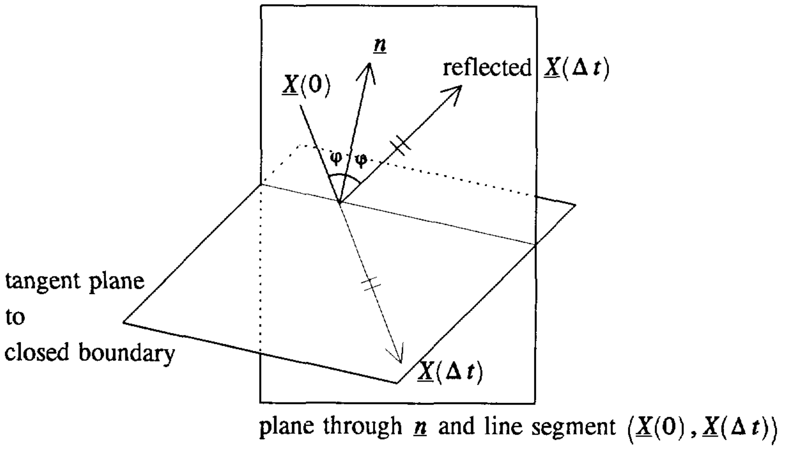

$\underline{n}$
$\underline{X}(0)$
reflected $\underline{X}(\Delta t)$
φ
φ
tangent plane to closed boundary
$\underline{X}(\Delta t)$
plane through $\underline{n}$ and line segment $(\underline{X}(0), \underline{X}(\Delta t))$

Figure (4.20): Determination of image point in case of perpendicular reflection in a three-dimensional space.

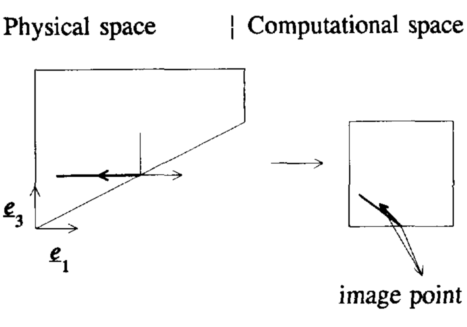


Figure (4.21a): Reflection in physical space regarding the principal axes of diffusion. In the absence of vertical diffusion a particle travelling in a horizontal plane will stay in this horizontal plane, even after reflection.

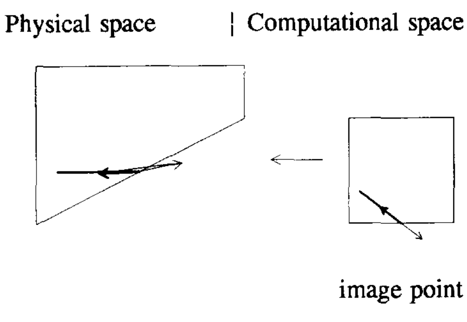


Figure (4.21b): Reflection in computational space regarding the principal axes of diffusion.

computed in the physical space. A similar reflection principle performed in the computational space is demonstrated in figure (4.21b). It approximates reflection according the procedure sketched in figure (4.21a).

The impact of these reflection alternatives are discussed in section 4.4.2 by means of numerical experiments. It is already suggested that, due to programming reasons, alternative (iii) - see figure (4.18), might be reconsidered. Although the diffusion coefficient near the boundary is reduced (computational artifact), it is noted that in practical applications the physical justification about the correct diffusion coefficient near closed boundaries, produced by the hydrodynamic model, already provokes discussion.

- Particle treatment near boundaries

It is concluded that various types of boundary conditions can be implemented in particle models. Reflection algorithms can be constructed such that the assumptions made in the hydrodynamic model are not violated. To summarize, three algorithms are distinguished for different boundary conditions:

1) Zero concentration boundaries are realized by a removal of particles (absorbtion).

2) A zero flux boundary is obtained if a particle suffers reflection. The position of a particle after reflection is ruled by the assumption about the diffusion tensor and by the normal vector to the boundary - see appendix E.

3) In principle all possible well-posed boundary conditions for multidimensional diffusion processes can be treated with the estimated mass flux procedure. The algorithm has been illustrated for boundary conditions written as a Robbin boundary condition. Then the spatial derivative of the concentration is expressed in terms of (i) the boundary condition and (ii) an estimation of the concentration value at the boundary. It is used in the approximation of the mass flux through the boundary. This estimated mass flux determines the intensity of a point source positioned at the boundary. The numerical experiments of section 4.4 will show that the method yields adequate results only if the time step and the estimation procedure for the concentration value at the boundary are properly chosen. This method is not recommended for general engineering purposes. The efficiency and robustness of the method can not compete with Eulerian analogues.

4.4 Numerical experiments

During the development of the particle model many numerical experiments were executed. These experiments are separated into two classes. The first class covers simple test cases to study numerical aspects of the particle model individually. The second class concerns the application of the particle model in physical problems.

The experiments of the first class go into the numerical treatment of (i) the advection step - section 4.4.1, (ii) the diffusive step - section 4.4.2, and (iii) boundary conditions - section 4.4.3. The main interest is the evaluation of the advantageous and disadvantageous aspects of the computation of particle displacements relative to an Eulerian grid. Furthermore the impact of reflection procedures at boundaries is studied.

The second class considers heat exchange in a square harbour boarding a river, and salt intrusion in a tidal inlet of an estuary - section 4.4.4. Usually, particles models are applied to discharge applications. Such an example is discussed in section 4.4.2 and in chapter 5.

4.4.1 The advection step

To illustrate the advection algorithm of section 4.3.4 four numerical experiments are executed:

(I) circulating flow in a two-dimensional rectangular grid (Molenkamp test),
(II) uniform flow in a two-dimensional irregular grid,
(III) circulating flow in a three-dimensional rectangular grid, and
(IV) wind-driven flow in a three-dimensional sigma grid.

For convenience stationary cases are considered. In all applications particles trace streamlines. The time step Δt defines the time needed to reach the end point. The defined streamline equations are valid within one grid cell. So the used time step τ is defined as the minimum of (i) Δt, and (ii) the minimal positive time needed to reach one of the grid cell sides. If τ is less than Δt the particle meets the cell boundary in time τ. The procedure is continued for a period $(\Delta t - \tau)$ tracing the streamline in the adjacent grid cell.

In the first experiment the grid under consideration is rectangular and uniform. In the predictor step closed streamlines are generated. The corrector step accounts for accurate approximations of the velocity modulus along the streamline. The ratio between this accurate velocity modulus and the approximated velocity modulus used in the predictor step(ds/dt) is modelled as in equation (4.102) of section 4.3.4.

The second experiment shows the influence of the accuracy in the approximation of the Jacobian in an irregular grid. The predictor-corrector method of section 4.3.4 will correct inaccurate approximations of the Jacobian. ds/dt is defined as in equation (4.104).

The third and fourth experiment illustrate the streamline algorithm in a three-dimensional velocity field. Only the predictor step is applied with $s(t) = t$.

Experiment I: Circulating flow in a two-dimensional rectangular grid

This experiment is known as the Molenkamp test as described in MOLENKAMP (1967). It tests the advection of a gaussian hill in a rigidly rotating flow with its centre within the flow domain. To guarantee that the pseudo velocity field satisfies the continuity equation it suffices to show that one single particle follows a closed streamline. The influence of the predictor-corrector method is demonstrated by tracing a circle (i.e., contour line of the initial gaussian hill). Many numerical algorithms exist to minimize the numerical problems in advection dominated flows. Such algorithms are reported in, for instance, VREUGDENHIL and KOREN (1993). Usually, numerical advection algorithms spread the gaussian hill, which is recognized by a deviation of the initial circle. Here it will be shown that for the predictor-corrector method the artificial spreading is acceptable. Together with the property of mass conservation the predictor-corrector method appears to be a powerful advection solver.

The tests are subjected to a coarse and a fine dimensionless staggered grid $[-1,1] \times [-1,1]$. Both grids with corresponding flow field are depicted in figure (4.22) and (4.24). The flow is constructed such that the continuity equation is satisfied as expressed by equation (4.91) of section 4.3.4.

The numerical data are given by:

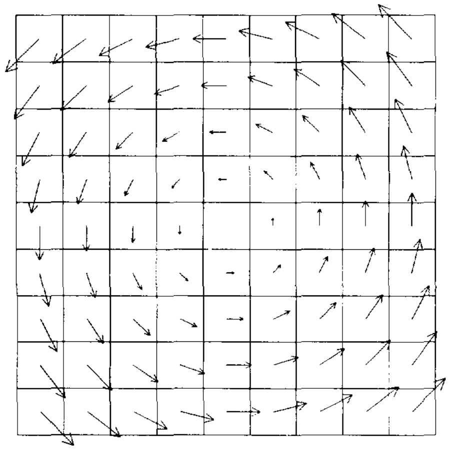


Figure (4.22): Vortex flow in rectangular coarse grid.

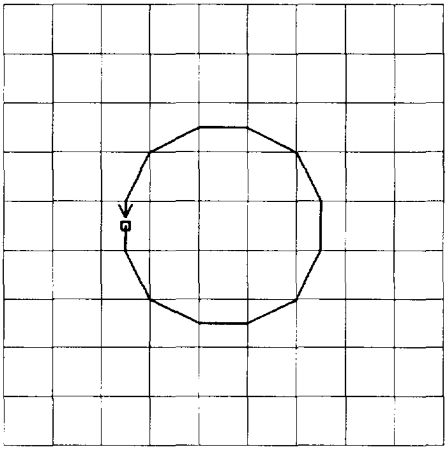


Figure (4.23): Closed streamline (1 round, T = 1).

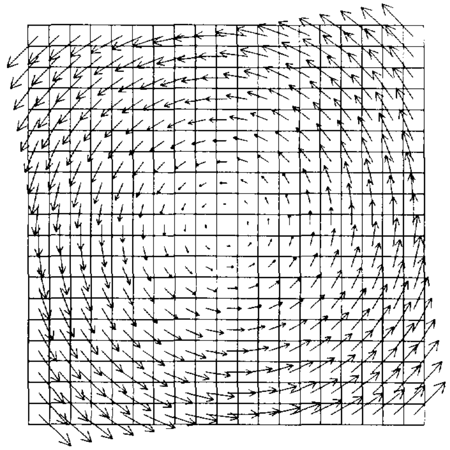


Figure (4.24): Vortex flow in rectangular fine grid.

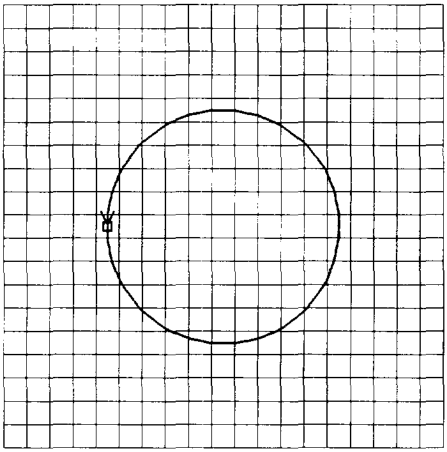


Figure (4.25): Closed streamline (1 round, T = 1).

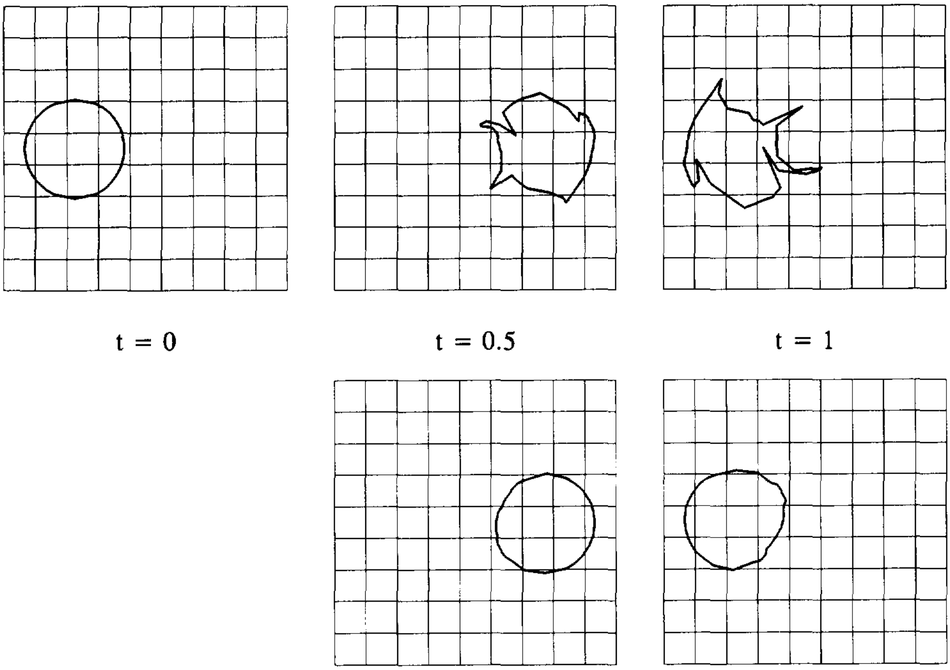


Figure (4.26): Tracing a circle of particles in a vortex flow with respect to a coarse grid (9 × 9).

4.26a - 4.26b - 4.26c (TOP): predictor step with $s(t) = t$,

4.26d - 4.26e (BOTTOM): predictor-corrector method with $s(t)$ as in equation (4.102) of section 4.3.4.

along a circular streamline with radius r_p, the velocity modulus remains constant $\sqrt{u^2 + v^2}\big|_{\underline{x}^p} = 2\pi r_p / T$, with $T = 1$ (dimensionless turn around time),
$\Delta t = 1/320$ (one complete rotation takes place in 320 time units),
the coarse grid consists of 9×9 equidistant grid cells,
the fine grid consists of 19×19 equidistant grid cells.

The choice of Δt is not essential. The algorithm solves the streamline equations within one grid cell, which implies that $u\tau/\Delta x \leq 1$ and $v\tau/\Delta y \leq 1$ are always satisfied.

The result of the streamline algorithm of section 4.3.4 is shown in figure (4.23) and figure (4.25). Since the method uses discrete velocity information, the produced closed streamline

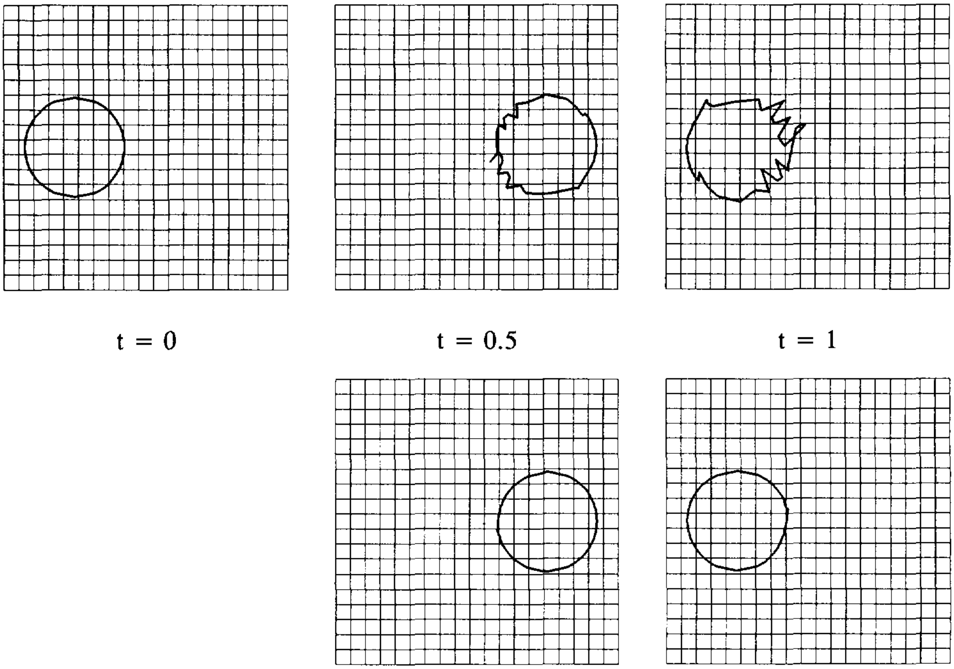


Figure (4.27): Tracing a circle of particles in a vortex flow with respect to a fine grid (19 × 19).

4.27a - 4.27b - 4.27c (TOP): predictor step with $s(t) = t$,

4.27d - 4.27e (BOTTOM): predictor-corrector method with $s(t)$ as in equation (4.102) of section 4.3.4.

does not exactly coincide with the circle. The length of the numerically obtained streamline is longer than the circumference of the circle. As a consequence, in coarse grids the end point of the tracer after time $T = 1$ has not yet met the starting point. This inaccuracy is larger in coarse grids.

Figure (4.26) and (4.27) show the influence of the predictor-corrector method with respect to a coarse and fine grid respectively. If the streamline equations are solved without applying the corrector step and setting $s(t) = t$ poor results are obtained especially near the centre of rotation - see figure (4.26b) and (4.26c). This is due to a discontinuous approximation of the velocity modulus along the streamline at grid cell sides. The corrector step improves the result considerably - see figure (4.26d) and (4.26e). The velocity modulus $\sqrt{u^2 + v^2}\big|_{\underline{x}^p}$ as it arises in the differential equation for $s(t)$ (equation (4.102) of section 4.3.4) is determined by the analytical solution of the flow field, i.e., $\sqrt{u^2 + v^2}\big|_{\underline{x}^p} = 2\pi r_p / T$. It is noted that ds/dt as in equation (4.104) yields almost the same results, since the velocity modulus along each streamline is constant $(ds/dt|_{init} \approx ds/dt|_{pred})$. The approximated position of the streamline does not coincide with the closed circle. Therefore the simulation is not able to produce exact solutions. Grid refinement results in a more accurate approximation of the position of the streamline (preserving mass conservation) as shown in figure (4.25). As a result, tracing a contour line on a fine grid gives adequate results - see figure (4.27).

Experiment II: Uniform flow in a two-dimensional irregular grid

The horizontal grid with corresponding uniform velocity field is depicted in figure (4.28a). The information of this velocity field is provided half way the grid cell sides (staggered grid). For one cell the vectors $u_\xi\ \underline{a}^1/\|\underline{a}^1\|$ and $u_\eta\ \underline{a}^2/\|\underline{a}^2\|$ are plotted in figure (4.28b). For convenience, set $\Delta\xi_m = \Delta\eta_n = 1 \quad \forall\, m,n$. It is emphasized that the given velocity components satisfy the continuity equation. It is given by

$$\left[\left(\sqrt{G_{\eta\eta}}\,u_\xi\right)_{m+\frac{1}{2},n} - \left(\sqrt{G_{\eta\eta}}\,u_\xi\right)_{m-\frac{1}{2},n}\right] + \left[\left(\sqrt{G_{\xi\xi}}\,u_\eta\right)_{m,n+\frac{1}{2}} - \left(\sqrt{G_{\xi\xi}}\,u_\eta\right)_{m,n-\frac{1}{2}}\right] = 0 \qquad (4.126)$$

The numerical experiments distinguish by the approximation of the Jacobian and the usage of the predictor-corrector method. The influence of the computation of the Jacobian is examined in experiment (i) until (iii). In this two-dimensional case the Jacobian is denoted with $\sqrt{G}$ - see equation (B.5) of appendix B.

(i) The Jacobian is approximated with

$$\begin{aligned}\sqrt{G} = &\left(\xi_p \sqrt{G_{\eta\eta}}\,\Big|_{m+\frac{1}{2},n} + (1-\xi_p)\sqrt{G_{\eta\eta}}\,\Big|_{m-\frac{1}{2},n}\right) \\ &\times \left(\eta_p \sqrt{G_{\xi\xi}}\,\Big|_{m,n+\frac{1}{2}} + (1-\eta_p)\sqrt{G_{\xi\xi}}\,\Big|_{m,n-\frac{1}{2}}\right)\end{aligned} \qquad (4.127)$$

(ii) The Jacobian is approximated with

$$\sqrt{G} = \frac{1}{4}\left(\sqrt{G_{\xi\xi}}\Big|_{m,n+\frac{1}{2}} + \sqrt{G_{\xi\xi}}\Big|_{m,n-\frac{1}{2}}\right)\left(\sqrt{G_{\eta\eta}}\Big|_{m+\frac{1}{2},n} + \sqrt{G_{\eta\eta}}\Big|_{m-\frac{1}{2},n}\right) \qquad (4.128)$$

(iii) The Jacobian is related to the exact grid cell area.

The predictor-corrector method is applied in experiment (iv):

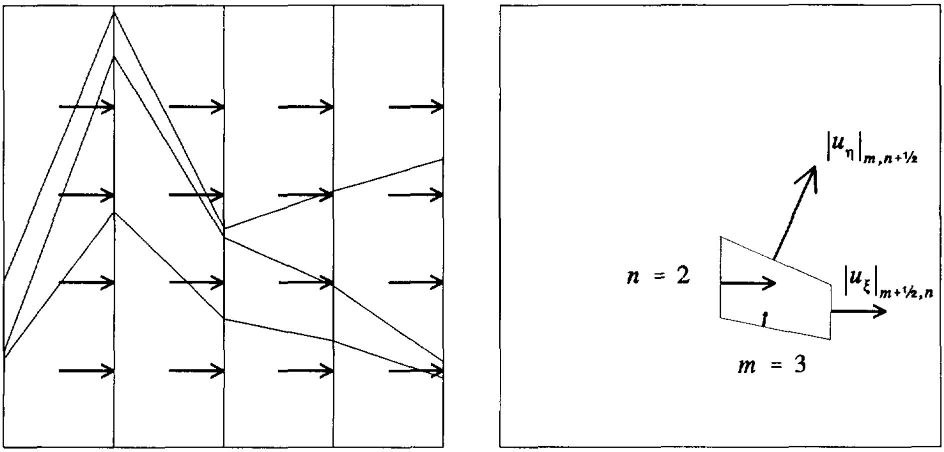


Figure (4.28): 4.28a (LEFT): uniform flow in a fixed irregular grid,
4.28b (RIGHT): available velocity information $u_\xi\, \underline{a}^1/\|\underline{a}^1\|$ and $u_\eta\, \underline{a}^2/\|\underline{a}^2\|$.

(iv) In the predictor step the Jacobian is approximated as in experiment (iii). It is noted that this choice is not essential. The corrector step will correct the predicted location irrespective of the approximation of the Jacobian used in the predictor step. The differential equation for $s(t)$ is chosen in accordance with equation (4.104) of section 4.3.4.

All depicted results - figure (4.29a) until figure (4.29d) - are compared with a first order Euler method. This method uses the same linear interpolation technique as it was used to construct the approximated streamline equations. The Euler method uses the numerically obtained value of the velocity components at the particle's location. Multiplication of these components with the chosen time step determines the displacement of the tracer. This displacement fixes the streamline.

The numerical data are given by:

$$\begin{array}{llll}
\sqrt{G_{\xi\xi}} & = 10 \text{ m} & \text{mean } \sqrt{G_{\eta\eta}} & = 10 \text{ m} \\
u & = 1 \text{ m/s} & T & = 25 \text{ s} \\
v & = 0 \text{ m/s} & \Delta t & = 1.0 \text{ s}
\end{array}$$

Figure (4.29) shows that the Lagrangian advection scheme accounts for the continuity equation (4.126), since straight streamlines are obtained. The result of the first order Euler method is affected by the grid.

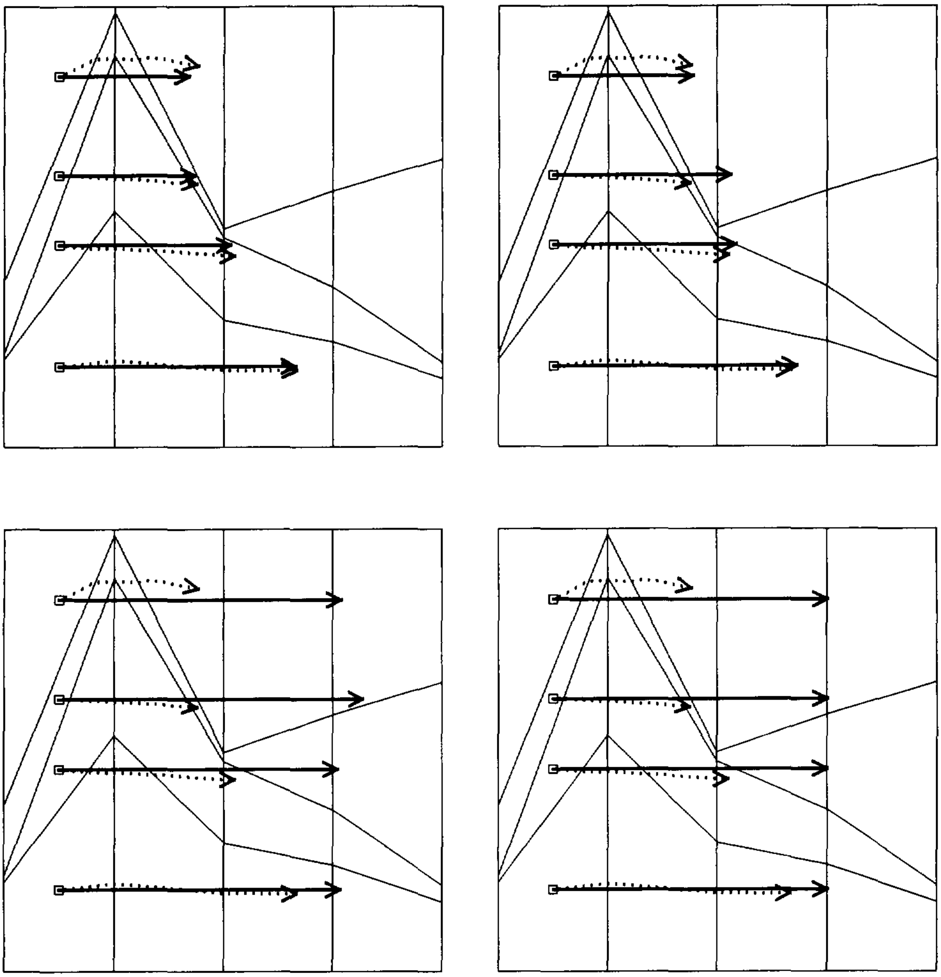


Figure (4.29): Particle tracks tracing streamlines in uniform flow.

dotted line: Euler method,

straight line: Lagrangian advection algorithm,

4.29a (TOP-LEFT): Jacobian approximated with (4.127), $s(t) = t$,

4.29b (TOP-RIGHT): Jacobian approximated with (4.128), $s(t) = t$,

4.29c (BOTTOM-LEFT): Jacobian approximated with exact grid cell area, $s(t) = t$,

4.29d (BOTTOM-RIGHT): predictor-corrector method, $s(t)$ as in equation (4.104) of section 4.3.4 (yielding exact solutions).

With respect to the computed end point of the track figure (4.29a) and (4.29b) show that the approximation of $\sqrt{G}$, as in (4.127) and (4.128) respectively, is as accurate as a first order Euler method. Striking better results are obtained if the Jacobian of the transformation is related to the exact grid cell area - see figure (4.29c). This observation was already reflected in section 3.3. Since the flow is considered uniform, exact results are obtained using the predictor-corrector method - see figure (4.29d). It is concluded that the accuracy of the approximation of the Jacobian is important in finding the correct end point of the track.

Experiment III: Circulating flow in a three-dimensional rectangular grid

The conditions that the divergence of the velocity field equals zero and that the normal component of the velocity vector at each closed boundary is zero lead to the conclusion that streamlines must close. Here a velocity field is constructed such that closed streamlines are obtained after one rotation. The depth-averaged flow field resembles the flow field of experiment I. The (dimensionless) computational domain is set to $[-1,1]\times[-1,1]\times[-1,0]$. The dimensionless analytical expression for the flow field is

$$\begin{cases} u = y + 3(z + ½)(x^2 + y^2 - r^2) \\ v = -x + 3(z + ½)(x^2 + y^2 - r^2) \\ w = -3z(z + 1)(x + y) \end{cases} \tag{4.129}$$

with $r = 0.9$. Tracks starting inside the cylinder $x^2 + y^2 = r^2$ remain captured. Since $w(0) = w(-1) = 0$, streamlines in $x^2 + y^2 < r^2$, $-1 < z < 0$ must be closed. Figure (4.30) shows two numerically obtained closed streamlines in a $9\times9\times5$ rectangular staggered grid. The algorithm uses velocity components given at the centre of grid cell sides, similar to the grid definition of section 2.5. The position of a streamline within one grid cell is given analytically as shown in section 4.3.4. This procedure solves the question of closure of streamlines.

Experiment IV: Wind-driven flow in a three-dimensional sigma grid

In complicated three-dimensional flows a streamline through a given point may pass near that point many times before actual closure - see TRUESDELL (1954). Many numerical tracking procedures are not capable of the accuracy required to investigate the question of closure of the streamlines - see MALLINSON and de VAHL DAVIS (1977). The accuracy of the analytical approach of section 4.3.4 is satisfactory in producing a definite answer. This is illustrated in the experiments described above. In flows in which a streamline makes several traverses, rounding-off errors may affect the accuracy required. An illustration will now be given for a wind-driven three-dimensional flow field.

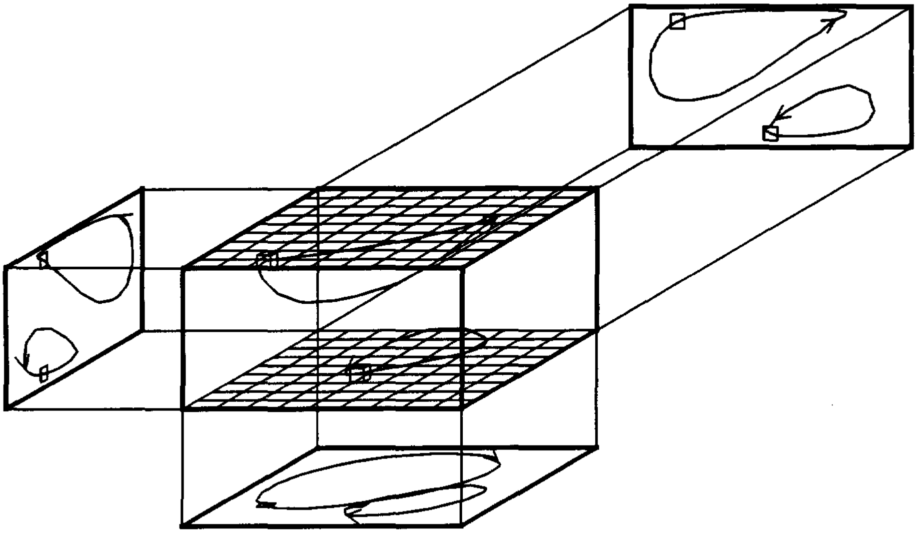


Figure (4.30): Two streamlines in a three-dimensional flow field. Plot contains projections on horizontal and vertical planes. Box represents $[-1,1] \times [-1,1] \times [-1,0]$.

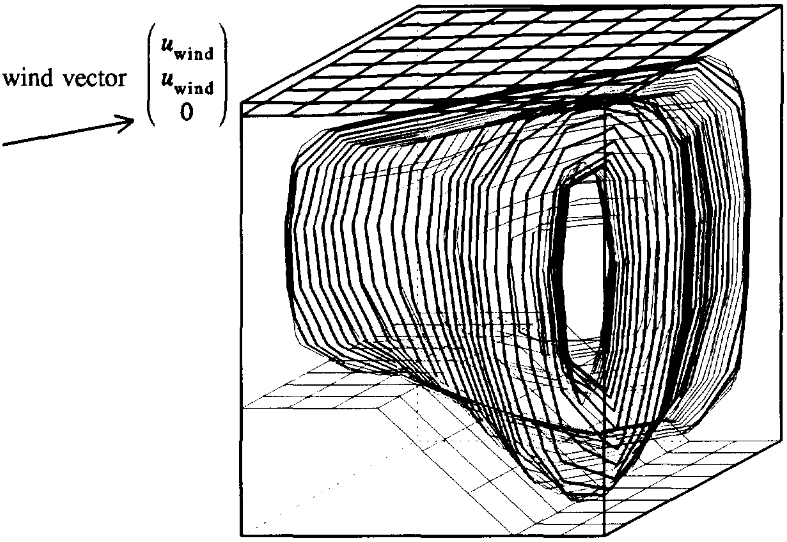

wind vector
$\begin{pmatrix} u_{\text{wind}} \\ u_{\text{wind}} \\ 0 \end{pmatrix}$

Figure (4.31): Single streamline in wind-driven flow field. The vertical dimension is distorted. Maximum depth is 1m, the horizontal dimensions are 80m × 80m.

A square lake (80m × 80m) is considered with space-varying depth (0.7m-1.0m). The hydrodynamic model TRISULA - see section 2.5, is used to generate stationary flow data in case of a constant south-west wind (wind direction equals 225°). The driving force of the fluid flow is given by a constant wind shear stress vector, $\underline{W}_{\mathbf{wind}}$, exerted on the free water surface. It is implemented as a free surface boundary condition: $1/\rho_{\mathbf{air}}\ \|\underline{W}_{\mathbf{wind}}\| = 2.5\mathrm{m}^2/\mathrm{s}^2$, with $\rho_{\mathbf{air}}$ the air density. This numerical value corresponds with a constant wind speed of approximately 30 m/s. Spatial increments used in the numerical model are $\Delta x = \Delta y = 10\mathrm{m}$, $\Delta\sigma = 0.2$ (5 layers). The computed stationary flow field satisfies the discretized continuity equation with an accuracy of $0.5\ 10^{-7}\ \mathrm{s}^{-1}$.

Figure (4.31) shows a particle track (streamline) that is traced for 24 hours. The numerically obtained track is not yet closed. It is reasonable to expect that it lies on a two-dimensional surface. As required, the streamline does not intersect with itself. If the computation of the track is continued the question of closure is still not solved, bearing in mind that over 10000 time steps (travelling from cell boundary to cell boundary) are involved in computing the streamline. The cumulative rounding-off errors may be responsible for not producing a closed streamline. A difference was recognized between single precision and double precision calculations. Another possible inaccuracy is introduced by the numerically obtained velocity field. The used flow field only approximates the condition of zero divergence. It is also noted that if the flow is not fully stationary the conclusion about the presence of closed streamlines is not justified.

It is concluded that the analytical approach in producing streamlines is accurate for simple flows and adequate for complicated flows.

4.4.2 The diffusive step

Diffusion problems applied to regular or irregular grids can be solved with finite differencing methods. Discretizations errors affect the produced solution. These errors propagate along the coordinate lines of the introduced grid. Therefore, such approximations are always grid dependent. It is emphasized that also rectangular grids introduce a similar grid dependency. This section will go into grid dependency aspects of the particle model, involving diffusion.

Two numerical experiments are discussed:

(I) anistropic diffusion in a nonorthogonal sigma grid, and
(II) isotropic diffusion in an orthogonal grid with space-varying aspect ratio.

The first test deals with nonorthogonality aspects in the computation of the noise-induced

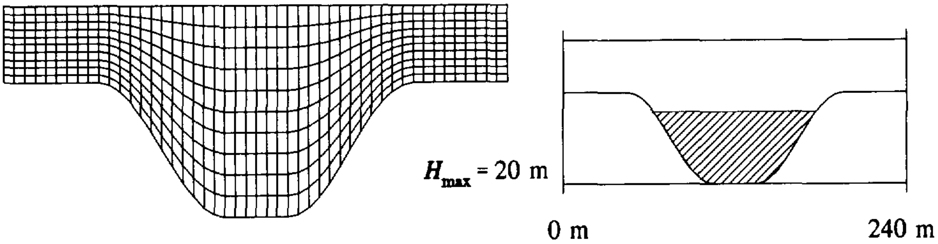


Figure (4.32): LEFT: nonorthogonal sigma grid,
RIGHT: geometrical description of trench and uniform polluted area (shaded).

curvature ($\tilde{\underline{h}}_C$). In addition the impact of reflection principles is studied. The second test deals with the influence of the noise-induced curvature term in an orthogonal application. Since the influence of the noise-induced drift is well-known, the test problems will be simplified considering a constant diffusivity ($\tilde{\underline{h}}_D = \underline{0}$).

Experiment I: Anisotropy in a sigma grid

This experiment considers a cross section of a vertical trench. The geometrical description of the trench is given in figure (4.32). It is assumed that for instance during slack tide, a uniform concentration is present at the bottom of the trench. It is known that a finite difference method may introduce an artificial vertical diffusion due to inaccurate numerical implementation of the transformed diffusion problem - see appendix C. This phenomenon is well illustrated if the ultimate case is considered, i.e., the absence of vertical diffusion. The assumed initial uniform concentration distribution at the bottom of the trench has to remain uniform when time evolves. This somewhat academic test is very useful to study (i) the asymptotic behaviour of a stationary uniform concentration distribution, (ii) the influence of the reflection principles introduced in section 4.3.6, (iii) the influence of the noise-induced curvature term, and (iv) the influence of the numerical treatment of particle displacements. These aspect are studied by means of numerical experiments listed in table (4.4.2.1).

It is emphasized that in the absence of vertical diffusion particles should be restricted to move in a horizontal plane. This is only achieved in experiment (ii). All other particle model experiments deal with approximated particle displacements in the computational space that will introduce a vertical displacement.

The asymptotic behaviour is quantified by a chi-square test. Let $f_{n^c}^{(n)}$ denote the number of particles in cell n^c at time $n\Delta t$. To test whether the computed distribution can still be considered uniform the following statistic is introduced

	simulation method	physical / computational space	D_H	$\underline{\tilde{h}}_C$	reflection principle as in
i	Finite differencing	computational	-	-	-
ii	Particle	physical	analytical	analytical	figure (4.21a)
iii	Particle	computational	analytical	analytical	figure (4.21b)
iv	Particle	computational	analytical	analytical	figure (4.19b)
v	Particle	computational	analytical	ignored	figure (4.21b)
vi	Particle	computational	numerical	numerical	figure (4.21b)

Table (4.4.2.1): Numerical experiments (anisotropy in a sigma grid).

$$Q^{(n)}_{N_0^c-1} = \sum_{n^c=1\,,\,f^{(0)}_{n^c}\neq 0}^{N^c} \frac{\left[f^{(n)}_{n^c} - f^{(0)}_{n^c}\right]^2}{f^{(0)}_{n^c}} \tag{4.130}$$

It can be proved that $Q^{(n)}_{N_0^c-1}$ has a limiting distribution that is $\chi^2\left(N_0^c - 1\right)$, for $n_p^0 \to \infty$ (approximated chi-square distribution). A similar chi-square value is also defined for the finite difference test

$$Q^{(n)}_{N_0^c-1} = \frac{1}{\mu_p^0} \sum_{i=1\,,\,C_i^{(0)}\neq 0}^{N^c} V_i \frac{\left[C_i^{(n)} - C_i^{(0)}\right]^2}{C_i^{(0)}} \tag{4.131}$$

In the experiments the number of cells with $f^{(n)}_{n^c} \neq 0$ equals $N_0^c = 100$, which gives a critical value $c = 134.64$ for a significance level that is approximately equal to 0.01 - see ROHLF and SOKAL (1981).

Other numerical and physical data are:

$\sqrt{G_{\xi\xi}} = 5$ m , $\Delta\sigma_k = 0.1$ ($k = 1,..,10$ layers)
$\Delta t = 0.1$ s , $T = 900$ s
$D_H = 2$ m²/s , $D_V = 0$ m²/s
$H_{\max} = 20$ m , $N_0^c = 100$
$n_p^0 = 57992$ (that is, for constant μ_p the concentration values are approximated with an absolute error less than 0.001)
initial particle injection is performed below $z = -10$ m ($= -\frac{1}{2}H_{\max}$)

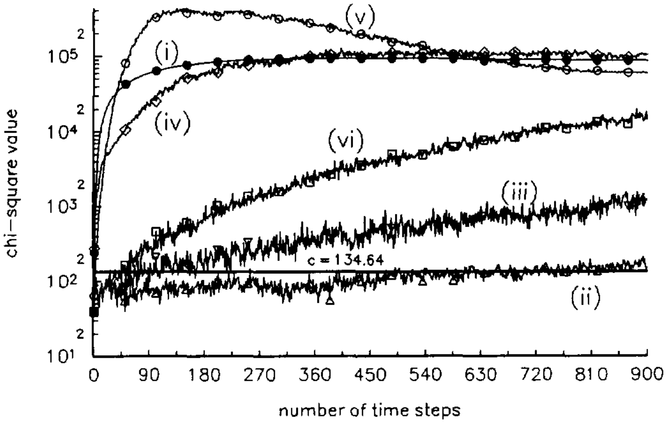


Figure (4.33): Chi-square values for experiment (i) until (vi).

For experiment (vi) it is assumed that the flow and geometry information is known at discrete grid points similar as in section 2.5. $\tilde{\underline{h}}_c$ is approximated with first order finite differences.

The test results are summarized in figure (4.33) and figure (4.35). Figure (4.33) gives the chi-square values as a function of time steps, while figure (4.35) sketches the concentration distribution within the trench. Figure (4.34) presents the visualization of the initial concentration distribution. It is noted that although $C = 1$ kg/m^2 per unit length for $z \leq -\frac{1}{2}H_{\max}$ and $C = 0$ for $z > -\frac{1}{2}H_{\max}$, the interpolation procedure used in the plotting routine does not show this sharp interface.

The experiments lead to the following conclusions:

1. The diffusive term in finite difference methods must be handled with care to avoid an artificial vertical diffusion - see experiment (i) and KESTER and UITTENBOGAARD (1990). In experiment (i) a first order upwind scheme is used to emphasize the phenomenon of artificial vertical diffusion. It is however noted that finite difference schemes do exist such that artificial diffusion is minimal - see section 3.3.
2. The computation of particle displacements in the computational space is as accurate as computation in the physical space as long as curvature terms are included. The introduced vertical displacement is negligible - see experiment (ii) and (iii).
3. Reflection against bottom has to be realized as demonstrated in figure (4.21). It yields the correct asymptotic behaviour - see experiment (iii). Perpendicular reflection results in a monotonically increasing error - see experiment (iv).

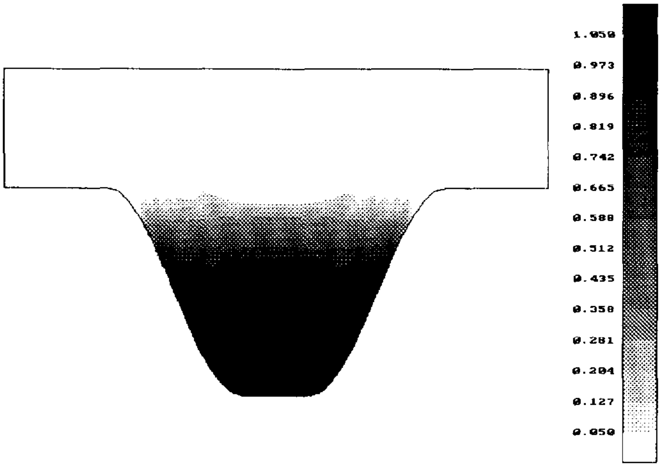


Figure (4.34): Visualization of initial uniform concentration distribution at bottom of trench. The grey-colour table defines the concentration values. This table also refers to figure (4.35).

4. The noise-induced curvature term has to be taken into account - see experiment (v).
5. The asymptotic solution obtained with first order approximations of $\underline{\tilde{h}}_C$ deviates from the uniform solution - see experiment (vi). The Euler method is sufficient with respect to time integration, while higher order methods are preferred to approximate $\underline{\tilde{h}}_C$.
6. The chi-square test is too strict to be applied for engineering purposes. Asymptotic solutions seem to be approximately uniform, while the chi-square test rejects this hypothesis - see figure (4.33) with respect to experiment (iii).

Experiment II: Grid dependency in the simulation of a discharge problem

A discharge of an initial slug of mass is released in a two-dimensional horizontal unbounded domain in the absence of a current. If this dispersion problem is approximated with a finite differencing method which solves the diffusion equation in a stretched grid the polluted region becomes oval shaped stressed by the steep concentration gradient at initial state. In fact, the solver should produce a circle-shaped polluted region. In this experiment particle displacements are computed in the transformed space to test if the produced solution is also affected by the grid. Since the exact solution is known the particle model will be compared with the analytical solution.

To support the interpretation of computational results contour plots of the concentration field are generated. However, if a contour plot of constant concentration values shows a grid

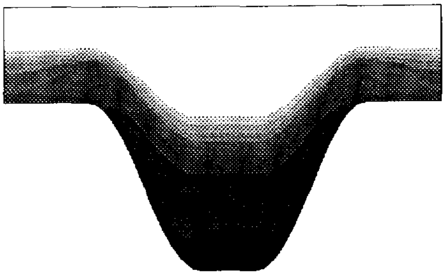


Figure (4.35a): Concentration distribution at bottom of trench produced by experiment (i).

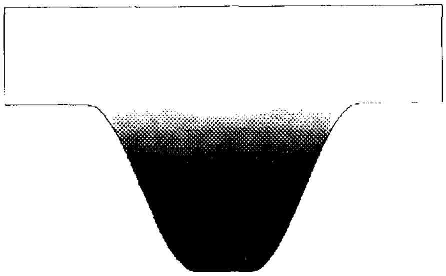


Figure (4.35b): Concentration distribution at bottom of trench produced by experiment (ii).

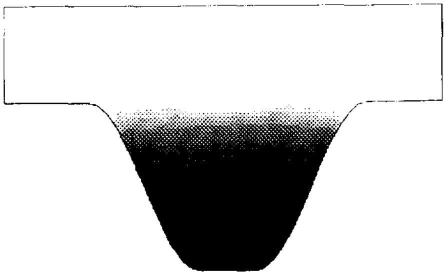


Figure (4.35c): Concentration distribution at bottom of trench produced by experiment (iii).

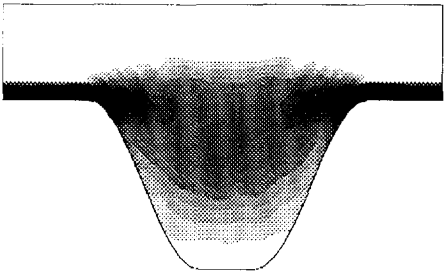


Figure (4.35d): Concentration distribution at bottom of trench produced by experiment (iv).

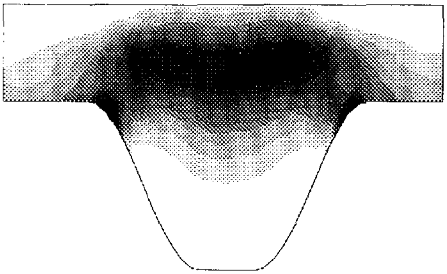


Figure (4.35e): Concentration distribution at bottom of trench produced by experiment (v).

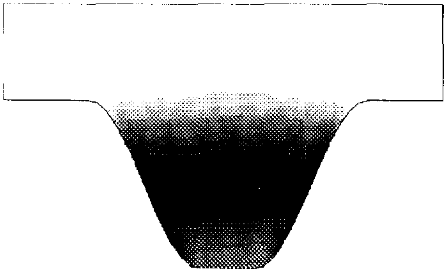


Figure (4.35f): Concentration distribution at bottom of trench produced by experiment (vi).

dependency it is sometimes not immediately clear that this is due to either the plotting routine or the numerical method that solved the concentration field. In general, numerical methods produce a discrete concentration field specified at grid nodes. The plotting routine uses this information and applies a linear interpolation technique within, for instance, a quadrangle. In a coarse grid, the obtained contour line results in a slightly misleading plot. Therefore, the concentration distribution is expressed with the point spread function algorithm in two dimensions, as described in section 4.3.1, to obtain a smooth distribution that is defined everywhere in space. Then the plotting routine will be applied to a much finer grid without repeating the particle simulation.

The analytical solution of a (point-) discharge problem, with no mean velocity and constant diffusivity D_H is given by - see FISHER *et al.* (1979),

$$C(x_1,x_2,t) = \frac{M/H}{4\pi t D_H} \exp\left(-\frac{(x_1-x_1^c)^2}{4D_H t} - \frac{(x_2-x_2^c)^2}{4D_H t}\right) \tag{4.132}$$

with M the initial mass of the conservative pollutant, H the constant water depth and $\left(x_1^c,x_2^c\right)$ the injection point at $t = 0$. The initial concentration distribution is written as a Dirac-delta function $\delta(\underline{x})$

$$C(x_1,x_2,0) = \frac{M}{H}\,\delta(x_1-x_1^c,x_2-x_2^c) \tag{4.133}$$

The particle method simulates the discharge by injecting $\mathrm{n_p}^0$ particles at the exact discharge location $\left(x_1^c,x_2^c\right)$. The physical data are given by

H = 5m, M = 500kg,
D_H = 1.0m²/s, and T = 100s.

The numerical grid is depicted in figure (4.36). Near the discharge location $\left(x_1^c,x_2^c\right)$ the grid cell dimensions equal $\Delta x_1 \times \Delta x_2$ with $\Delta x_1 = 1.0$m and $\Delta x_2 = 10.0$m. In many hydrodynamic applications a grid is introduced that is refined in the area of interest. Usually this is done in both x_1- and x_2- direction. In this numerical test problem the grid will only be stretched in x_1- direction to obtain a well illustration of the influence of $\underline{\tilde{h}}_C$ while using particle models. Away from $\left(x_1^c,x_2^c\right)$ the grid is stretched with a factor $\alpha > 1$, i.e., the i^{th} grid cell left and right of $\left(x_1^c,x_2^c\right)$ has dimensions $\alpha^{i-1}\Delta x_1 \times \Delta x_2$. As a consequence, setting $x_1^c = 0$ such that $\xi = 0$ corresponds with $x_1 = x_1^c$,

$\sqrt{G_{\eta\eta}} = \Delta x_2 =$ 10.0m (constant throughout the grid), and

$$\sqrt{G_{\xi\xi}} = -\log\alpha \, \frac{\Delta x_1}{1-\alpha} \, \alpha^{|\xi|}$$

These global grid transformation relations arise in the expression for the particle displacement - see e.g., equation (4.75) of section 4.3.2. Three experiments are executed with $\alpha = 1.5$:

(i) particle model executed in the physical space (irrespective of the transformed grid):

$$\begin{cases} X_1^{(n+1)} = X_1^{(n)} + \sqrt{6 D_H \Delta t} \, R_1 \\ X_2^{(n+1)} = X_2^{(n)} + \sqrt{6 D_H \Delta t} \, R_2 \end{cases} \tag{4.134}$$

(ii) particle model executed in the transformed space (particle positions are related to the grid) in the absence of the curvature term:

$$\begin{cases} \xi^{(n+1)} = \xi^{(n)} + \dfrac{1}{\sqrt{G_{\xi\xi}}} \sqrt{6 D_H \Delta t} \, R_1 \\ \eta^{(n+1)} = \eta^{(n)} + \dfrac{1}{\sqrt{G_{\eta\eta}}} \sqrt{6 D_H \Delta t} \, R_2 \end{cases} \tag{4.135}$$

(iii) particle model executed in the transformed space including the curvature term:

$$\begin{cases} \xi^{(n+1)} = \xi^{(n)} + \dfrac{1}{\sqrt{G_{\xi\xi}}} \sqrt{6 D_H \Delta t} \, R_1 + \dfrac{1}{\sqrt{G_{\xi\xi}}} \dfrac{\partial}{\partial \xi} \left(\dfrac{1}{\sqrt{G_{\xi\xi}}} \right) D_H \Delta t \\ \eta^{(n+1)} = \eta^{(n)} + \dfrac{1}{\sqrt{G_{\eta\eta}}} \sqrt{6 D_H \Delta t} \, R_2 \end{cases} \tag{4.136}$$

with R_1 and R_2 uniform random numbers in $(-1,1)$. The numerical data equal: $\Delta t = 0.2$s (500 time steps) and $n_p^0 = 10000$.

The results are presented in figure (4.36), where iso-concentration contour lines are plotted. In experiment (i) and (iii) the contour lines correspond with approximate circles as required. Experiment (ii) shows an artificial drift due to the ignorance of $\underline{\tilde{h}}_C$. Particles are drifted away from the injection point. This drift is the strongest in the vicinity of the injection point. As a consequence, the polluted region shows two local maxima which is physically incorrect. It is concluded that the influence of $\underline{\tilde{h}}_C$ is noticeable and may not be ignored. In general hydrodynamic applications $\sqrt{G_{\xi\xi}}$ is given at discrete points in space related to the grid cell dimension in ξ - direction (local grid transformation). However, an accurate numerical

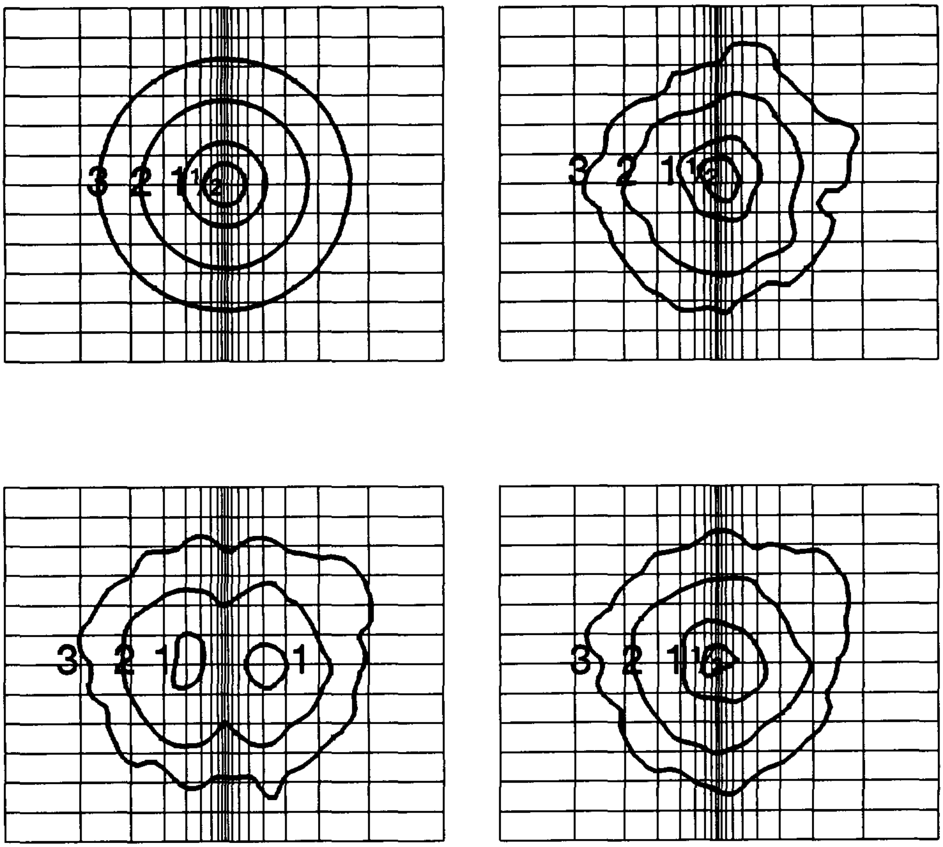


Figure (4.36): Iso-concentration contour lines of discharge in a two-dimensional domain (150m × 120m) which is covered with a stretched grid:

$C_i = C_{max} \exp\left(-i^2/2\right)$ for $i = ½, 1, 2, 3$ with $C_{max} = \dfrac{M/H}{4\pi D_H T}$

4.36a (TOP-LEFT):	analytical solution,	
4.36b (TOP-RIGHT):	experiment (i)	physical space,
4.36c (BOTTOM-LEFT):	experiment (ii)	computational space (curvature excluded),
4.36d (BOTTOM-RIGHT):	experiment (iii)	computational space (curvature included).

approximation of $\tilde{h}_C$ is necessary to obtain reliable results. Therefore, in diffusion dominated flows it is *not* recommended to apply the particle model in transformed coordinates. If the hydrodynamic model introduces such a grid, best results are obtained if the particle model is executed in the physical space irrespective of a predefined grid. Only for uniform grids the particle model in transformed coordinates is as accurate as the particle model in Cartesian coordinates.

4.4.3 The estimated mass flux procedure at boundaries

In section 4.3.6 the numerical treatment of boundary conditions in particle models is described. Three procedures have been reviewed: (i) absorbtion, (ii) reflection, and (iii) injection using an estimation of the mass flux through boundaries. The absorbtion procedure is straightforward and does not run up against numerical implementation problems. The reflection procedure is already demonstrated in the previous section. This section will demonstrate the third procedure.

A one-dimensional (vertical) settling tube is considered. The vertical local flow velocity is ignored and the advection of each particle is determined by a constant falling velocity w_s. A positive falling velocity implies that sediment falls towards the bed due to gravity. The dispersion is given by the diffusion coefficient D_V. The transport process is governed by the following one-dimensional advection-diffusion equation

$$\frac{\partial C}{\partial t} - \frac{\partial (w_s C)}{\partial z} = \frac{\partial}{\partial z}\left(D_V \frac{\partial C}{\partial z}\right) \tag{4.137a}$$

in which C represents the concentration in kg/m per unit area. The vertical coordinate is denoted with $z \in [-H, 0]$. The initial condition at $t = 0$ is given by

$$C(z,0) = C_I(z) \tag{4.137b}$$

At the top of the settling tube transfer of mass is excluded

$$w_s C\Big|_{z=0} + D_V \frac{\partial C}{\partial z}\Big|_{z=0} = 0 \tag{4.137c}$$

In view of section 4.3.6 the particle method will be applied to two different bottom boundary conditions in order to illustrate the estimated mass flux procedure. The first experiment defines

$$C\big|_{z=-H} = C_e \tag{4.137d-i}$$

while the second experiment sets

$$D_V \frac{\partial C}{\partial z}\bigg|_{z=-H} = -w_s C_e \tag{4.137d-ii}$$

with C_e an equilibrium bed concentration. For convenience a constant diffusivity is applied. Subsequently, the particle method is summarized:

1) K cells of height $\Delta H = H/K$ are introduced. Distribute n_p^0 particles according to the initial concentration C_I. Each particle carries a mass μ_p^0.

2) At each discrete time step $t = t_n = n\Delta t$, the following procedure is applied:

   i) Estimate the mass outflow per time step $F_{\partial\Omega}^{n+1}(-H)$. A first order approximation is obtained from

$$F_{\partial\Omega}^{n+1}(-H) = \Delta t \left(\underbrace{w_s C^n \big|_{z=-H}} + \underbrace{D_V \frac{\partial C^n}{\partial z}\bigg|_{z=-H}} \right) \tag{4.138}$$

In experiment (i):	prescribed boundary condition (4.137d-i)	needs to be estimated
In experiment (ii):	needs to be estimated	prescribed boundary condition (4.137d-ii)

   ii) Simulate the process for each particle $Z = Z^p$,

$$Z^{n+1} = Z^n - w_s \Delta t + \sqrt{6 D_V \Delta t}\, R \tag{4.139}$$

   R represents a uniform random number in $(-1,1)$. In the absence of decay the evolution process of the particle mass is ignored. Particles that reach the top of the settling tube are reflected. Particles that cross the bottom $z = -H$ are absorbed. The amount of mass that is absorbed in time Δt is recorded and stored in $\overline{F}_{\partial\Omega}^{n+1}(-H)$ - see equation (4.124) of section 4.3.6.

   iii) Inject a finite number of particles near the bottom according to the procedure sketched in section 4.3.6. The total mass to be injected per time step equals $\overline{F}_{\partial\Omega}^{n+1}(-H) - F_{\partial\Omega}^{n+1}(-H)$.

iv) Evaluate the mean concentration per unit area in interval I_k

$$\overline{C}_k^n = \frac{1}{\Delta H} \sum_{p=1}^{n_p^n} \mu_p^n \chi_{\{Z^p(t_n) \in I_k\}} \tag{4.140}$$

The procedure presented above requires the estimation of either $C|_{z=-H}$ or $\partial C/\partial z|_{z=-H}$. In general particle applications the estimation of these quantities may be cumbersome. If the estimation is inaccurate the procedure fails in producing reliable concentration distributions. For a large number of particles and a small spatial step the procedure is executed without severe trouble. The mean concentration approximates the concentration at the centre of the grid cell. Simply approximations are

for experiment (i): $$\left.\frac{\partial C^n}{\partial z}\right|_{z=-H} = \frac{1}{\Delta H}\left(\delta_{-1} C^n|_{z=-H} + \sum_{j=0}^{J-1} \delta_j \overline{C}_{K-j}^n\right) \tag{4.141}$$

and

for experiment (ii): $$C^n|_{z=-H} = \sum_{j=0}^{J-1} \beta_j \overline{C}_{K-j}^n - \tfrac{1}{2}\Delta H \beta_{-1} \left.\frac{\partial C^n}{\partial z}\right|_{z=-H} \tag{4.142}$$

The coefficients δ_j and β_j, $j = -1,..,J-1$, are determined by consistency relations. In applications with large grid cell dimensions regression techniques must be used to obtain reasonable approximations.

The experiments are executed using the following physical and numerical data:

$H = 8\text{m}$	$K = 20$
$w_s = 0.1\text{m/s}$	$n_p^0 = 5000$
$D_V = 0.1\text{m}^2/\text{s}$	$J = 1$
$C_e = 1\text{kg/m}$ per unit area	$\Delta t = 0.1\text{s}$

$T = 300\text{s}$, time in which stationary state has been reached.

For this choice of physical data the stationary state solutions of experiment (i) and (ii) are identical. The initial condition C_I is chosen such that $C_I|_{z=-H} = C_e$ and $D_V\, \partial C_I/\partial z\,|_{z=-H} = -w_s C_e$.

Additional tests were executed studying the influence of the number of particles, the number of grid cells, the injection procedure, the choice of the approximate order J as it arises in equation (4.141) and equation (4.142), and the prescribed initial condition. The results of the experiments executed with the above described data are summarized in figure (4.37) and (4.38). The following aspects are distinguished: (i) the stationary concentration distribution, (ii) the breakthrough curve, defined as the amount of mass floating out through the bottom of the settling tube as a function of time, and (iii) the estimated $C|_{z=-H}$ and $\partial C/\partial z|_{z=-H}$ as a function of time. With respect to the Dirichlet case (4.137d-i) the residence (flushing) time is computed. This residence time, t_{res}, results from the integration of the concentration distribution - see DIMOU and ADAMS (1989), i.e.,

$$t_{res}(t) = \frac{\int_{-H}^{0} C(z,t)dz}{C_e w_s} \tag{4.143}$$

This quantity is associated with the breakthrough curve. The stationary results are compared with the analytical solution. The time-varying results are compared with a finite difference solution. This finite difference solution is obtained using central differences and a Crank-Nicolson time integration scheme. At the boundary third order finite differences are used. It has been observed that at stationary state the absolute error of the finite difference model divided by the absolute error of the particle model is of order 1.

The experiments give rise to the following considerations:

- The number of intervals K is based on judgement. The optimal choice is obtained considering: (1) a large value of K is only effective if $\mathrm{n_p}^0$ is also large, (2) a small value of K gives an inadequate approximation of the estimated mass flux resulting in meaningless solutions, (3) in boundary layer experiments K must be chosen such that several grid cells cover the present boundary layer.
- An increase of $\mathrm{n_p}^0$ reduces the scatter in the approximation of $C|_{z=-H}$ and $\partial C/\partial z|_{z=-H}$. In general, the concentration is approximated more accurately than the concentration derivative - see figure (4.37c) and (4.38c). As a consequence, the simulation of Neumann boundary conditions is more robust than the simulation of Dirichlet boundary conditions.
- The procedure fails if the maximum particle displacement per time step exceeds the grid cell dimensions (here: ΔH). Then, the simulation process conflicts with the applied injection procedure - see section 4.3.6.
- In equation (4.141) and (4.142), the mean concentration value $\overline{C}_k^n$ is a first order approximation of $C(z_k)$ with z_k positioned at the centre of cell I_k. It is sufficient to apply $J = 1$. The coefficients become $\delta_{-1} = -2$, $\delta_0 = 2$, $\beta_{-1} = 1$ and $\beta_0 = 1$.
- At stationary state the influence of the initial condition has vanished. An initial condition only affects the breakthrough curve.

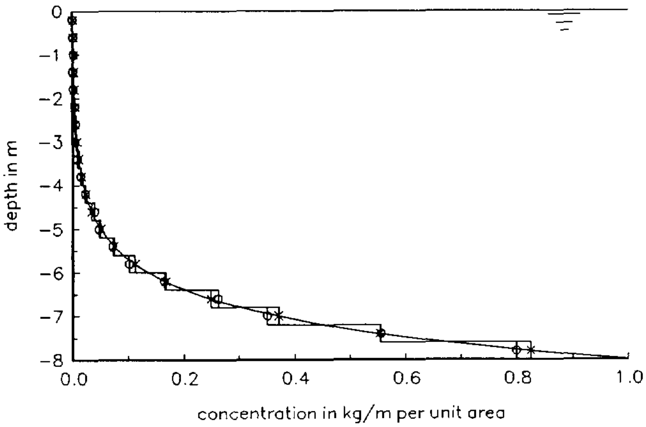


Figure (4.37a): Stationary mean concentration solution of settling tube experiment (i), with prescribed concentration value at the bottom (Dirichlet boundary condition).

circle: particle model, cross: analytical solution.

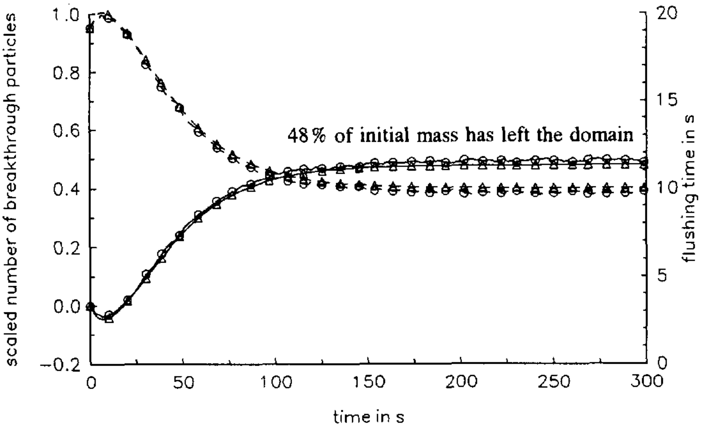


Figure (4.37b): Breakthrough curve and flushing times of settling tube experiment (i), with prescribed concentration value at the bottom (Dirichlet boundary condition). Results are scaled with n_p^0.

circle:	particle model,	triangle up:	finite difference solution,
straight line:	breakthrough particles,	dashed line:	flushing times (t_{res} → 10s).

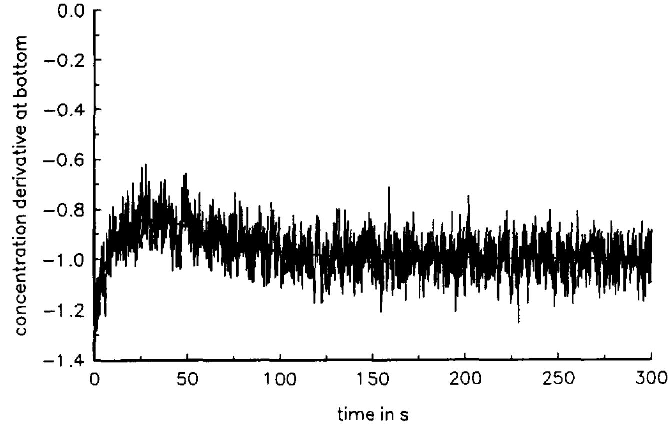


Figure (4.37c): Estimated concentration derivative at the bottom of settling tube experiment (i), with prescribed concentration value at the bottom (Dirichlet boundary condition).
scattered line: particle model, straight line: finite difference solution.

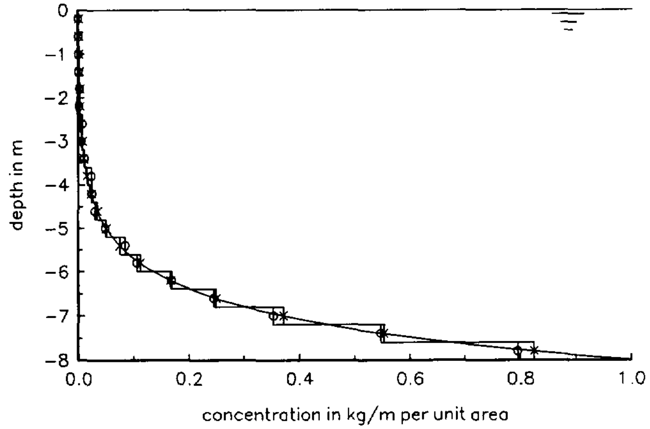


Figure (4.38a): Stationary mean concentration solution of settling tube experiment (ii), with prescribed concentration derivative at the bottom (Neumann boundary condition).
circle: particle model, cross: analytical solution.

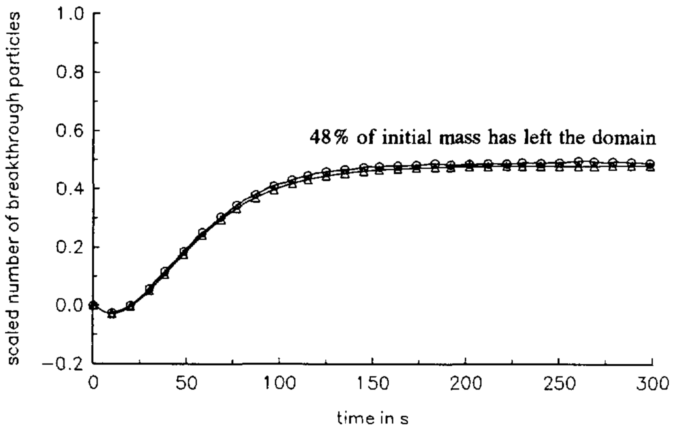


Figure (4.38b): Breakthrough curve of settling tube experiment (ii), with prescribed concentration derivative at the bottom (Neumann boundary condition). Results are scaled with n_p^0.
circle: particle model, triangle up: finite difference solution.

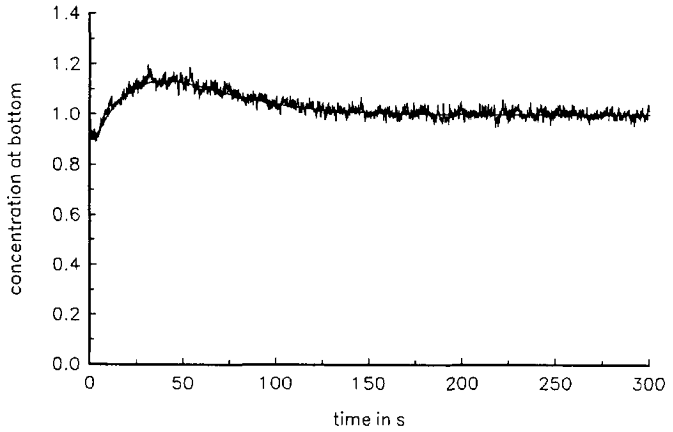


Figure (4.38c): Estimated concentration at the bottom of settling tube experiment (ii), with prescribed concentration derivative at the bottom (Neumann boundary condition).
scattered line: particle model, straight line: finite difference solution.

4.4.4 Physical experiments

This section considers two physical experiments. The first experiment deals with heat exchange in a scale model of a square harbour boarding a river. The second experiment deals with salt intrusion in the estuary of a tidal inlet. These applications are complicated and illustrative demonstrations of the described particle model.

Experiment I: Heat exchange in a square harbour

The knowledge of the flow pattern in a harbour and its entrance is very important for the prediction of the distribution of pollutants inside a harbour. For steady flow conditions the exchange of matter through the harbour entrance is mainly caused by lateral turbulent fluctuations of velocity - see LANGENDOEN (1992). The flow pattern is frequently calculated numerically applying the eddy viscosity concept. Booij stated in BOOIJ (1991) that turbulence is mainly generated in the mixing layer in the harbour entrance and convected through the harbour. The depth-averaged computations, summarized in BOOIJ (1991), executed by the software package PHOENICS with a $k-\epsilon$ turbulence model, yield velocity components and eddy viscosities. Flow and grid information from this computation is used to simulate the heat exchange process.

At initial state the experiment deals with a homogeneous mixed (warm) heat field in the harbour and a cold river-water flow outside the harbour. Cold water packets and warm water packets will be entrained into a mixing layer. This exchange process is conceived as a (2DH, depth-averaged) diffusion process by introducing eddy diffusivities. It is noted that two-dimensional simulations do not take into account directly three-dimensional effects. These effects concern e.g., density effects because of temperature variations and depth-dependent velocities. The experiments in a scale model of the square harbour showed that these effects are of negligible importance - see BOOIJ (1994). The comparison between numerical experiments and physical experiments concentrate on qualitative aspects rather than quantitative aspects. Moreover, the initial homogeneous (warm) heat field of the numerical experiment compared with the actual initial condition of the physical experiment shows only qualitative agreement. Even qualitative reliable approximations must show a negligible numerical diffusion. Therefore, accepting the advection-diffusion equation as a description of the heat exchange process the particle model seems to be appropriate.

As time elapses the temperature inside the harbour is reduced because of the heat exchange through the harbour entrance. The simulated rate of this temperature reduction is very sensitive with respect to the modelled diffusivities. For instance, a larger eddy diffusivity expresses a larger turbulent exchange of heat. As a consequence, the rate of temperature reduction also increases. This rate of temperature reduction appears to be suitable in a comparison study between numerical and physical experiments. It is quantified by the relaxation time, that is, the time in which the temperature is reduced from a reference temperature to $1/e$ times the same reference temperature. Relaxation times are obtained at 13 locations from temperature measurements in a scale model of the square harbour. The associated relaxation times are used to evaluate the model results. The numerical experiments are executed with the particle model and with the finite difference model of section 3.3. The latter alternative is applied to study the influence of numerical diffusion manifested in the computed mean relaxation time.

Resuming, the heat exchange in a harbour is measured in a scale model, and is compared with particle model solutions and finite difference approximations. The laboratory experiments are taken as a reference. The relaxation time derived from these experiments is regarded as the correct value. Given the outcome of the physical experiment numerical models are being compared. Attention is paid to:

- formulation of eddy diffusivity and the presence of numerical diffusion,
- computation of relaxation times, and
- numerical aspects of the particle model, such as
  (i) the computation of a complete particle displacement, and
  (ii) a reinitialization procedure.

- Model description

The scale model of the square harbour consists of a (1m × 1m) harbour along a river of 1m width. The computational model consists of a (1m × 1m) harbour along a river of 0.3m width. This width reduction hardly affects the computed velocity distribution near the harbour entrance. Doing so, the number of grid points is reduced which benefits storage and computational efficiency. The water depth equals approximately 0.1m. At the inflow boundary a uniform velocity profile is prescribed with $\|\underline{u}\| = 0.5\text{m/s}$. In the computations the harbour is covered by $40 \times 40 = 1600$ equal grid cells with dimension $0.025\text{m} \times 0.025\text{m}$. The geometrical description of the square harbour and the locations of the 13 stations are sketched in figure (4.38). The computed flow field is visualized in figure (4.39). Streamlines of the stationary flow field are obtained with the advection algorithm of the particle model - see section 4.3.4. The duration of a circulation in the harbour is approximate 25s.

- Eddy diffusivity

In BOOIJ (1991) it is argued that the computation of the depth-averaged flow field can be

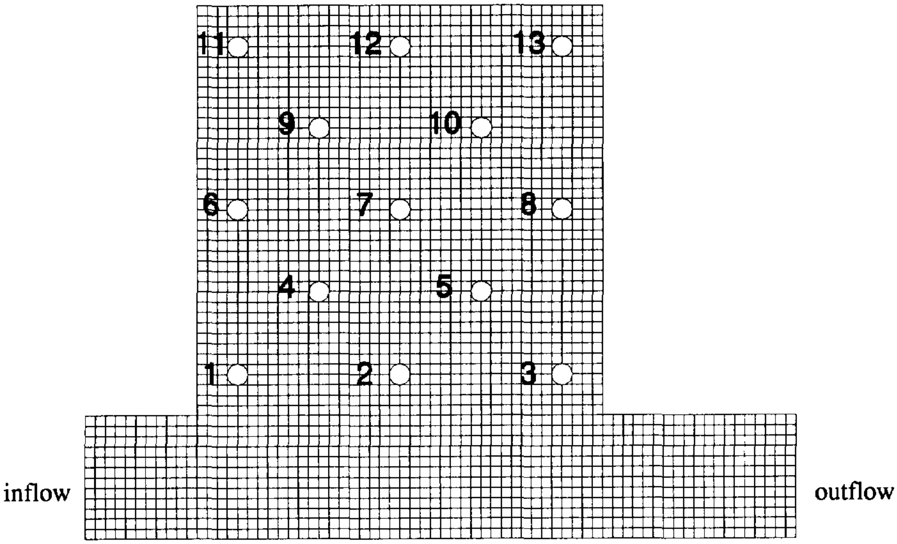


Figure (4.38): Geometrical description of square harbour (1m × 1m), including 13 stations. Width of river = 0.3m, depth = 0.1m.

executed with a constant eddy viscosity, which yields a reasonable resemblance with the observations. The eddy viscosity is related to the eddy diffusivity by the turbulent Prandtl (or Schmidt) number σ_t - see RODI (1980), such that

$$D = \frac{\nu_t}{\sigma_t} \tag{4.144}$$

with D the turbulent diffusivity of heat and ν_t the turbulent viscosity. A value for the latter quantity was obtained from the $k-\epsilon$ model computations (PHOENICS - see BOOIJ (1991)). σ_t reflects the nature of the mixing process. Equation (4.144) expresses the analogy between turbulent heat transport and momentum transport. The used value of σ_t applies for a mixing layer as present at the harbour entrance. It is also possible to regard σ_t as a model parameter. By repeated execution of a numerical model the optimal σ_t is computed with respect to the observations of the physical model (trial- and error calculations). This method can be justified because the value of this turbulent Prandtl number includes physical effects that are not taken into account by the numerical model. If σ_t is estimated with a numerical model, this model parameter can also contain numerical deficiencies. For instance, the presence of numerical diffusion causes an underestimate of the value of σ_t. The experiments discussed below distinguish computations using a constant eddy diffusivity and computations

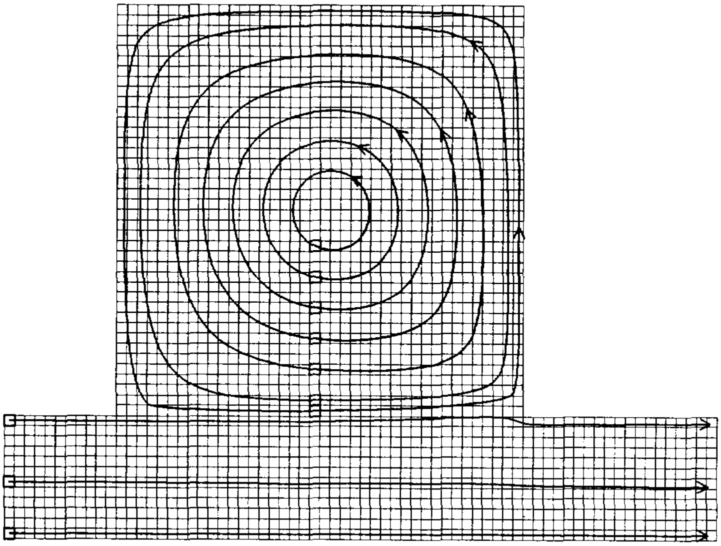


Figure (4.39a): Streamlines in harbour, obtained with predictor scheme of section 4.3.4, T = 30s.

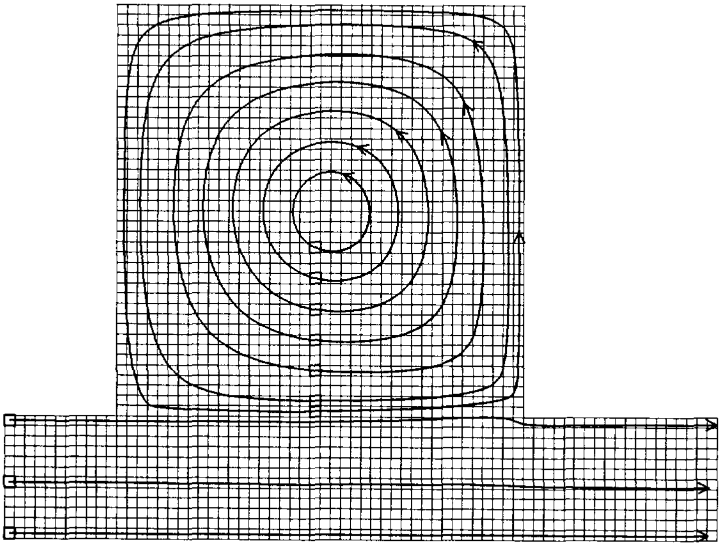


Figure (4.39b): Streamlines in harbour, obtained with predictor-corrector scheme of section 4.3.4, T = 30s.

using a space-varying eddy diffusivity. The case of constant eddy diffusivity is considered to exclude additional numerical inaccuracy in the computation of particle displacements due to linear interpolation of eddy diffusivities and the approximation of spatial derivatives of eddy diffusivities. σ_t is chosen such that the computed relaxation time of accurate finite difference approximations matches the relaxation time associated with the measurements.

- Computation of the relaxation time

In a square harbour containing a single eddy it is reasonable to assume that in an effective time range the temperature decay will follow an exponential decay law. Initially the exchange by diffusion is less important than, for instance, exchange by convection. Therefore, an initial period is excluded from the effective time range. It is also taken into consideration that in numerical experiments it is easy to impose a uniform heat distribution. However, regarding the measurements, a uniform initial temperature distribution is difficult to realize. This also motivates the exclusion of the starting period from the effective time range. Subsequently, for each individual station an effective time range is obtained from finite difference approximations. To reduce the arbitrariness of the considered time range nine ranges are evaluated. It results in nine approximations of the relaxation time. The final approximation of the relaxation time, accepted here as the actual relaxation time at a particular station, is found by averaging those 9 values. This procedure will be illustrated later - see figure (4.42). Relaxation times associated with a single station of the measurements show a noticeable dependency on the chosen time range (3%). So, the assumption of an exponential decay law is of limited use. Consequently, it is not justified to compare results of individual stations. The comparison between computational results and measurements will be pinned down to a relaxation time averaged over all stations inside the harbour.

Once an effective time range is selected the maximum entropy method of section 4.3.1 is applied with one moment constraint. The corresponding procedure yields two Lagrange multipliers λ_0 and λ_1. Thus, the computed heat distribution is fitted in a specific range $\left(t_a, t_b\right)$ with the exponential distribution

$$T(t) = T_0 \exp\left[\lambda_0 + \lambda_1 t\right] \tag{4.145}$$

for $t \in \left(t_a, t_b\right)$ with $T_0 = 1°$. λ_0 is dimensionless, while the dimension of λ_1 is one over time (s^{-1}). The relaxation time, τ, is then defined as

$$\tau = -1/\lambda_1 \tag{4.146}$$

- Numerical aspects of the finite difference model

The finite difference model that solves the heat distribution uses a rectangular equidistant staggered grid in the horizontal plane. For time integration a two step A.D.I.-scheme is applied. To show the influence of numerical diffusion two approximations for the advection flux are considered. The advection flux is defined in equation (3.7) of section 3.3.

(i) A second order upwind advection flux in one direction and a second order central advection flux in the other direction. In addition a Forester filter is applied to remove negative contributions.
(ii) First order upwind advection fluxes in both directions. This method produces positive contributions, such that filtering is not necessary.

It is known that the second approach will introduce numerical diffusion. This will be displayed by a smaller relaxation time.

- Numerical aspects of the particle model

The particle model computes individual particle tracks. The displacement per time step is expressed by,

$$\begin{cases} \Delta X^n = \displaystyle\int_{t_n}^{t_n+\Delta t} u\,dt + \frac{\partial D^n}{\partial x}\Delta t + \sqrt{6D^n\Delta t}\,R_1 \\ \Delta Y^n = \displaystyle\int_{t_n}^{t_n+\Delta t} v\,dt + \frac{\partial D^n}{\partial y}\Delta t + \sqrt{6D^n\Delta t}\,R_2 \end{cases} \tag{4.147}$$

with R_i $(i = 1,2)$ a uniform random number in $(-1,1)$. The transformed coordinates ξ and η are defined as $\xi = x/\Delta x$ and $\eta = y/\Delta y$.

(i) advection followed by diffusion versus diffusion followed by advection

Two alternatives are considered in the approximation of (4.147):

(1) A particle starting at $\underline{X} = \underline{X}_0$ traces the pathline through $\underline{X} = \underline{X}_0$ for a period Δt. In addition the stochastic forcing is computed at $\underline{X} = \underline{X}_0$ and added to the deterministic displacement - see figure (4.40a).
(2) A particle starting at $\underline{X} = \underline{X}_0$ suffers a stochastic displacement. This determines an intermediate position of the particle. Then, the particle traces the pathline through this intermediate position for a period Δt (deterministic displacement) - see figure (4.40b).

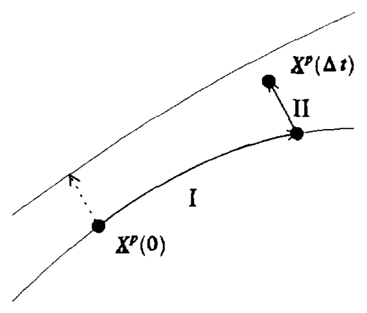


Figure (4.40a): Advection (I) followed by diffusion (II).

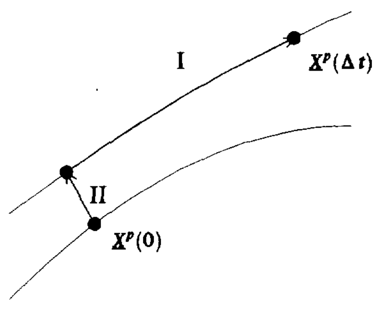


Figure (4.40b): Diffusion (II) followed by advection (I).

The second alternative is inconsistent with equation (4.147) since the point of impact of the influence of the local flow velocity does not coincide with $\underline{X}_0$. This inconsistency is quantified by comparison between observed and computed relaxation times.

(ii) the predictor-corrector method for the advection step

The computation of the advection step is executed with the predictor method as well as the predictor-corrector method - see section 4.3.4. In the latter case the local flow velocity is approximated with a bilinear interpolation scheme.

(iii) approximation of space-varying eddy diffusivities and spatial derivatives

The particle model uses the grid information of the hydrodynamic model. Since particle positions are not restricted to grid nodes, each physical quantity must be approximated for every point in space. Viscosities are defined at the centre of each grid cell - see section 2.5. Since spatial derivatives of eddy diffusivities have to be computed, viscosity information, provided by the hydrodynamic model, is extended to u_ξ - and u_η - locations (centre of grid cell sides). Then, providing constant grid spacing and isotropic diffusion,

$$D_{m+\frac{1}{2},n} = \tfrac{1}{2}\left((\nu_t)_{m,n} + (\nu_t)_{m+1,n}\right) / \sigma_t \tag{4.148a}$$

$$D_{m,n+\frac{1}{2}} = \tfrac{1}{2}\left((\nu_t)_{m,n} + (\nu_t)_{m,n+1}\right) / \sigma_t \tag{4.148b}$$

Approximations used to compute particle displacements in ξ - direction become

$$D(\xi^p,\eta^p) = \xi^p D_{m+\frac{1}{2},n} + (1 - \xi^p) D_{m-\frac{1}{2},n} \qquad \text{(disregarding lateral variations)} \tag{4.149}$$

$$\frac{\partial D}{\partial \xi}(\xi^p,\eta^p) = D_{m+½,n} - D_{m-½,n} \tag{4.150}$$

with similar expressions for the η - direction.

(iv) the reinitialization procedure

A uniform heat distribution at initial state requires a large amount of particles. During computation many particles will leave the computational flow region at the outflow boundary. As a consequence, the resolution decreases as time evolves. This can be overcome with a reinitialization of particles during computation. If a certain fraction α of the initial number of particles, n_p^0, has left the domain, say at time $t = t_r$, the particle load is reinitialized. Two reinitialization procedures are considered:

(1) A heat distribution is computed at time $t = t_r$ using the concepts of section 4.3.1. This distribution is regarded as an initial condition at $t = t_r$ and n_p^0 particles are newly injected as described in section 4.3.3. As a consequence, the weight of the heat contribution of each particle, μ_p, will be less than its contribution at time $t < t_r$. It is clear that such a reinitialization process will introduce artificial spreading, since the reinitialized particles are uniformly injected inside each cell. It is expected that it will affect the results in applications with large grid cell dimensions.

(2) All particles present in the flow region at time $t = t_r$ are split in two particles. The number of particles inside the computational flow region increases with a factor two, and each individual heat contribution is halved. This procedure does not introduce artificial spreading. It just increases the resolution.

- Boundary conditions

The initial heat distribution is given in nondimensional form. For that purpose the temperature inside the harbour is set to 10, while the temperature outside the harbour is set to zero.

At the inflow boundary a zero temperature is prescribed. Since the inflow boundary is chosen far away from the harbour entrance it is not expected that particles will reach the inflow boundary. It is noted that for the finite difference model this $T = 0$ boundary is a necessity. Due to filtering a $\partial T/\partial n = 0$ boundary may result in a noticeable heat inflow as time evolves. The mean relaxation time in the harbour would increase.

At the outflow boundary particles will leave the computational domain. The easiest treatment of particles that reach the outflow boundary is the total absorbtion procedure. As discussed in section 4.3.6, this procedure is consistent with a zero temperature boundary. Application of such an outflow boundary condition introduces a small boundary layer. It is found by running the finite difference model that the results of applying a $T = 0$ boundary, instead of the more obvious $\partial T/\partial n = 0$ boundary, hardly influences the computed mean relaxation time. Therefore, the $T = 0$ boundary condition, i.e., a total absorbtion of particles is applied at the outflow boundary.

- Model results

The physical experiments are reported in BOOIJ (1994). This source motivates an averaged relaxation time of approximately 95s. This value is accepted as the actual relaxation time. The turbulent Prandtl number is estimated from finite difference approximations. Recommended values are $\sigma_t = 0.5$ for mixing layers, $0.5 < \sigma_t < 0.8$ for most free flows and $\sigma_t = 0.9$ for flow near walls and recirculating flows - see LAUNDER (1991). The fourth order time splitting scheme gave rise to $\sigma_t = 0.47$. Accepting $\sigma_t = 0.47$ the finite difference computations are repeated to estimate the value of a constant eddy viscosity/diffusivity. Again, by comparison between the computed mean relaxation time with the experimental value, it was deduced that $\nu_t \approx 6.2\ 10^{-4}\mathrm{m}^2/\mathrm{s}$.

- Finite difference approximations

The relaxation times associated with each individual station are averaged over nine time intervals $[175,340+20i]$, $i = 0,..,8$. The so-obtained mean relaxation times are averaged over the 13 stations in the harbour. The heat distribution obtained with the fourth order time splitting scheme with space-varying ν_t is given in figure (4.41). The temperature evolution that corresponds with station 7 together with the exponential decay distribution, given in equation (4.145), is plotted in figure (4.42). The results of various numerical experiments are listed in table (4.4.4.1).

For a time step $\Delta t = 0.02$s the maximum Courant number is approximately one. The computed mean relaxation time associated with the first order upwind scheme is smaller than the wished 95s. This is explained by the presence of numerical diffusion. It is reminded that σ_t is chosen such that the results of the fourth order time splitting scheme, for $\Delta t \downarrow 0$, yield $\tau \approx 95$s. So, the first order upwind simulation does not yield predictive ability.

- Particle model approximations

The experiments and mean relaxation times obtained by running the particle model with simulation time $T = 500$s are summarized in table (4.4.4.2). The reinitialization procedure that duplicates each particle is activated if the number of computational particles is less than

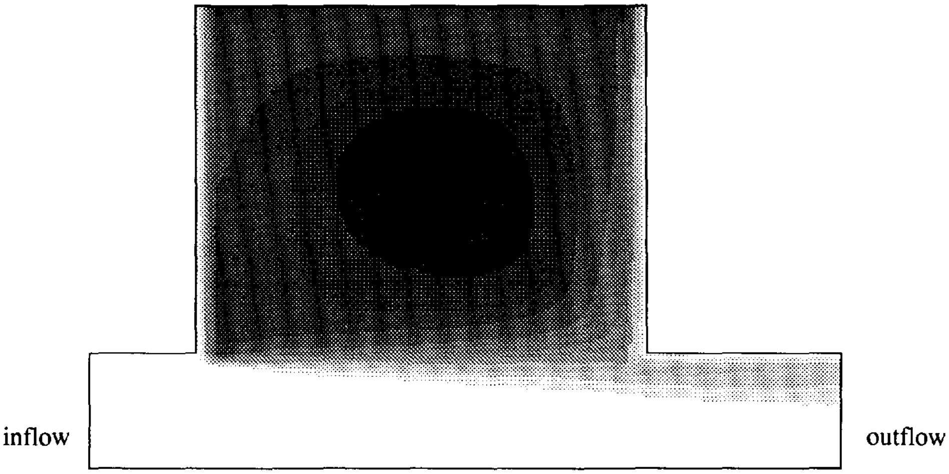


Figure (4.41): Heat distribution at T = 500s obtained by finite difference simulation (space-varying ν_t, fourth order time splitting, Δt = 0.004s). The grey shading denotes temperature ranges. The maximum temperature in centre of harbour approximates 0.0665 (black), while at the inflow boundary the temperature remains 0 (white).

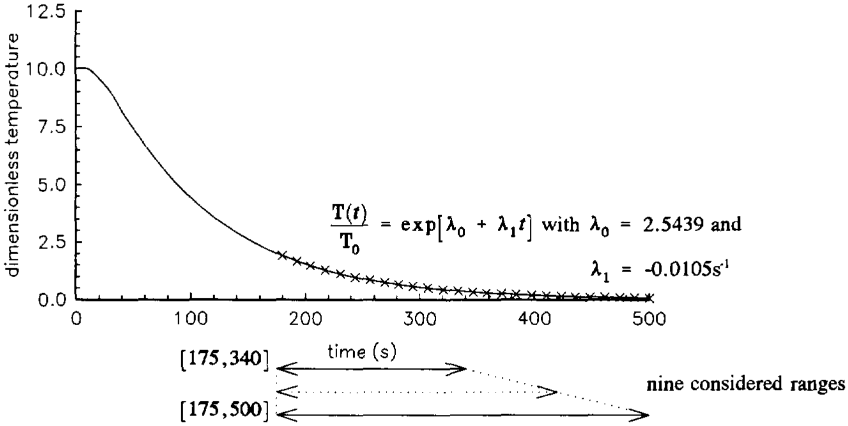


Figure (4.42): Heat decay in centre of harbour (station 7), simulated with finite differencing. Crosses indicate exponential decay distribution T(t), t > 175s, with relaxation time τ = 94.98s.

space-varying D	fourth order time splitting	Δt = 0.004s	τ = 94.98s (0.004)
		Δt = 0.02s	τ = 95.04s (0.012)
		Δt = 0.1s	τ = 96.92s (0.038)
		Δt = 0.2s	τ = 101.35s (0.031)
space-varying D	first order upwind	Δt = 0.02s	τ = 89.21 (0.001)
constant D (with ν_t = 6.2 $10^{-4}\text{m}^2/\text{s}$)	fourth order time splitting	Δt = 0.02s	τ = 94.75s (0.005)

Table (4.4.4.1): Finite difference experiments and computed mean relaxation time τ in harbour. The standard deviation of the computed mean relaxation time is included in parenthesis.

αn_p^0 with $\alpha = 0.75$.

Experiments with constant eddy diffusivity

Temperature profiles that correspond with experiment (1) are given in figure (4.43). This figure also contains the expected behaviour at large times (exponential decay). Although the computed distributions contain a significant scatter the computed relaxation times give rise to an exponential decay distribution that resembles the finite difference approximation. It has been tried to reduce the scatter in two ways. In the first way the initial number of particles has been increased with a factor 10, i.e., $n_p^0 = 160000$ - experiment (4), while in the second way experiment (1) has been repeated six times with different seeds of the random number generator. This second approach averages the so-obtained six realizations of the transport process. Figure (4.44) contains the result of experiment (4). Figure (4.45) sketches the averaged distributions of the second approach. Both approaches reduce the scatter. Still a mean relaxation time of approximate $\tau = 95$s is computed.

All experiments result in an approximation of the mean relaxation time that is found within a 2%-range of the expected value. Due to the observed scatter it is reasonable to expect that the error in the particle approximations is dominated by the number of particles. This error affects the computed mean relaxation time. With $\tau \in (95\text{s},97\text{s})$ confidence is gained that $\sigma_t \approx 0.47$ is appropriate.

Experiments with space-varying eddy diffusivity

Particle simulations yield $\tau \approx 97$s. It may be concluded that finite difference approximations

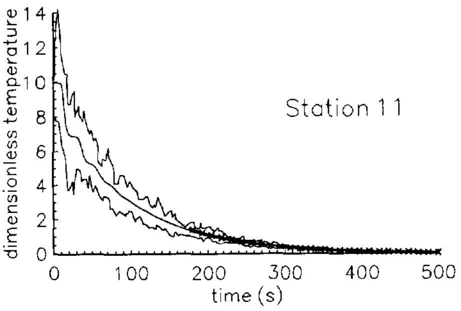
	n_p^0	Δt	advection step	reinitialization	displacement	τ (constant D)	τ (space-varying D)
(1)	16000	0.2	predictor/ corrector	n_p^n is doubled	advection → diffusion	95.07 (1.348)	97.41 (0.816)
(2)	16000	0.2	predictor/ corrector	particle load is newly injected	advection → diffusion	96.15 (1.720)	98.35 (1.387)
(3)	16000	0.2	predictor/ corrector	no reinitialization	advection → diffusion	96.42 (3.608)	98.70 (3.263)
(4)	160000	0.2	predictor/ corrector	n_p^n is doubled	advection → diffusion	94.90 (0.424)	97.36 (0.234)
(5)	160000	0.2	predictor/ corrector	no reinitialization	advection → diffusion	95.30 (1.288)	96.75 (1.069)
(6)	16000	0.02	predictor/ corrector	n_p^n is doubled	advection → diffusion	97.07 (1.144)	98.14 (0.992)
(7)	16000	0.2	predictor	n_p^n is doubled	advection → diffusion	94.77 (1.250)	96.19 (0.956)
(8)	16000	0.2	predictor/ corrector	n_p^n is doubled	diffusion → advection	95.08 (1.155)	

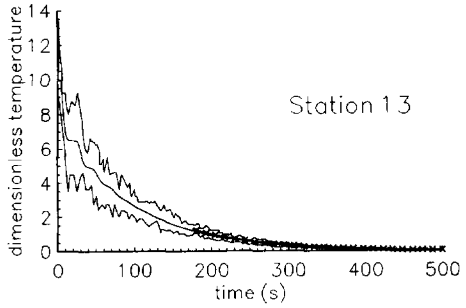
Table (4.4.4.2): Particle tracking experiments and computed mean relaxation time τ in harbour. The standard deviation of the computed mean relaxation time is included in parenthesis.

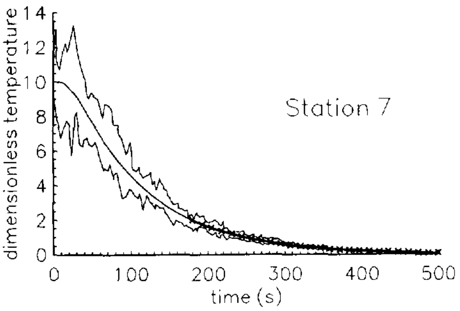
(with space-varying eddy diffusivity) are affected by numerical diffusion. However, it also makes sense to conclude that approximations of $D(\xi,\eta)$ and its spatial derivatives used in the particle model are less accurate than the overall accuracy of the finite difference simulation. It is noted that the approximations given in equation (4.149) and (4.150) are only first order accurate, while the finite difference model is at least second order accurate.

- Concluding remarks

- The computed mean relaxation time results from (i) averaging nine approximations of relaxation times associated with a single station and (ii) averaging all individual station results. Finite difference computations yield small standard deviations (< 0.04s). The particle approximations give larger standard deviations (with order of magnitude 1.5s for n_p^0 = 16000). This is mainly due to the stochastic approach. The standard deviation diminishes with increasing number of particles. It is observed in figure (4.45) that a more smooth distribution is also obtained by evaluation of various realizations. It results in the same effect as can be observed by examining one realization with increased number of particles.







Particle simulation - experiment(1):

mean τ in harbour: τ = 95.07s

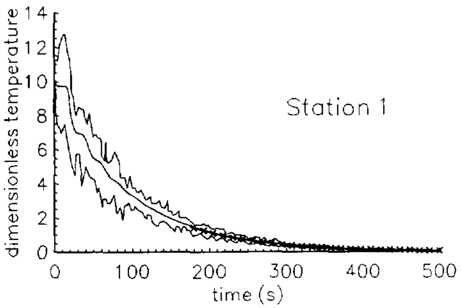
τ at station 1: τ = 95.41s

τ at station 3: τ = 94.76s

τ at station 7: τ = 97.07s

τ at station 11: τ = 92.53s

τ at station 13: τ = 92.70s



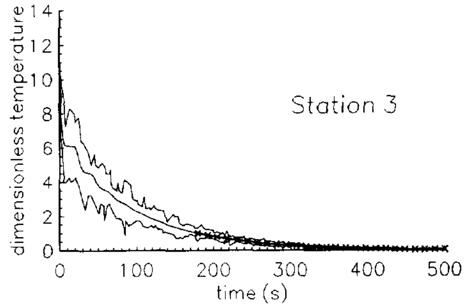
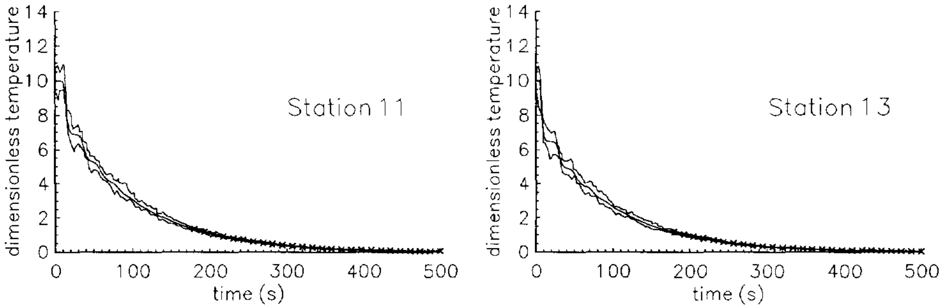
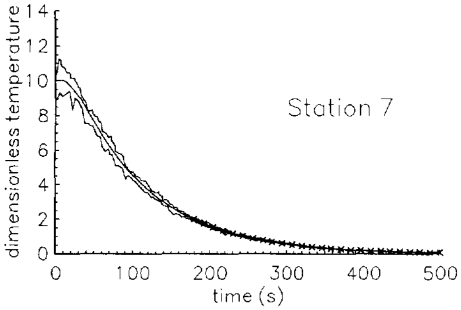


Figure (4.43): Heat decay at 5 stations in harbour simulated with constant eddy diffusivity.

The individual plots contain:

- envelope of particle distribution obtained with experiment (1),
- (smooth) finite difference approximation, and
- exponential decay distribution, derived from particle experiment, for t > 175s (crosses).





Particle simulation - experiment(4):

mean τ in harbour: τ = 94.90s

τ at station 1: τ = 94.29s

τ at station 3: τ = 95.90s

τ at station 7: τ = 95.25s

τ at station 11: τ = 95.22s

τ at station 13: τ = 94.35s

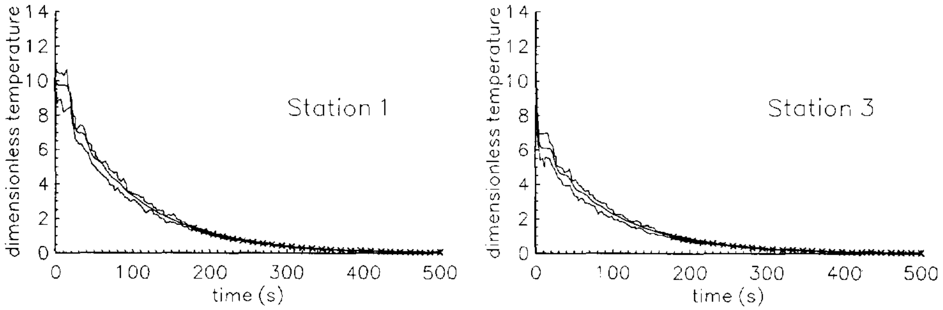
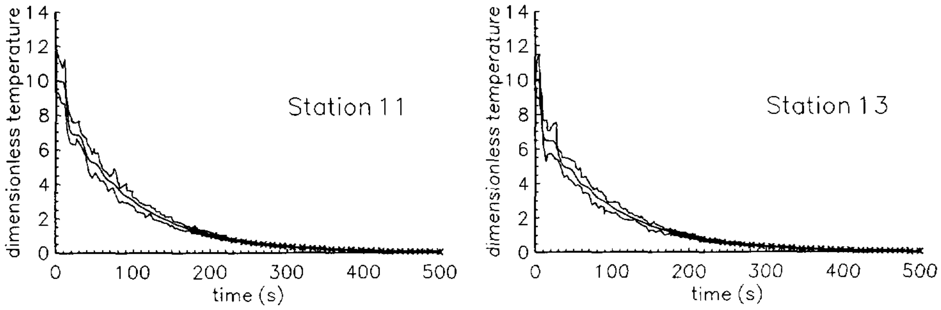
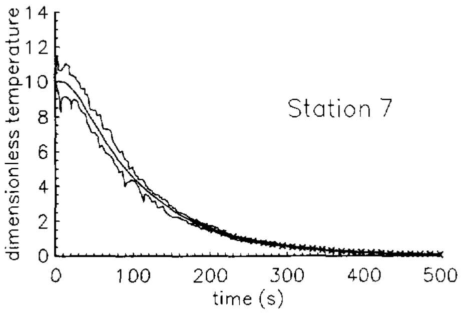


Figure (4.44): Heat decay at 5 stations in harbour simulated with constant eddy diffusivity.

The individual plots contain:

- envelope of particle distribution obtained with experiment (4),
- (smooth) finite difference approximation, and
- exponential decay distribution, derived from particle experiment, for $t > 175$s (crosses).





Averaging six realizations of experiment (1):

mean τ in harbour: τ = 94.36s (0.450)

τ at station 1: τ = 94.17s

τ at station 3: τ = 94.77s

τ at station 7: τ = 94.24s

τ at station 11: τ = 93.68s

τ at station 13: τ = 94.01s

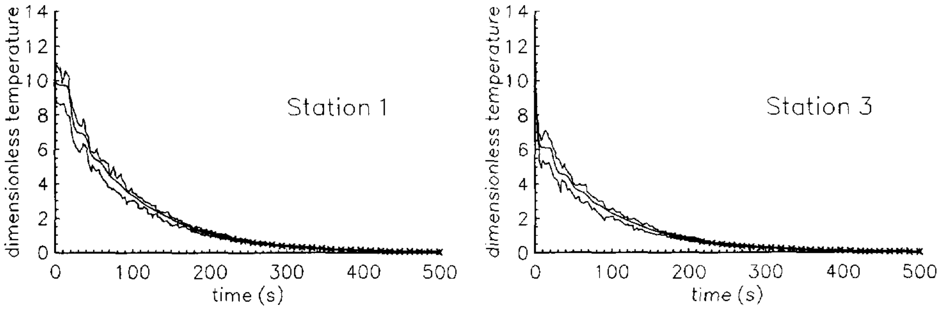


Figure (4.45): Heat decay at 5 stations in harbour simulated with constant eddy diffusivity.

The individual plots contain:

- envelope of averaged particle distribution deduced from 6 realizations of experiment (1),
- (smooth) finite difference approximation, and
- exponential decay distribution, derived from particle experiments, for t > 175s (crosses).

- Accuracy is mainly determined by n_p^0 and Δt. The time step should be chosen such that each particle displacement per time step is limited by the grid cell dimensions. Time step reduction is only effective if the overall error is not dominated by n_p^0. It is noted that time step limitations are only recommended in the presence of diffusion. The algorithm of the advection step is constructed such that in stationary applications the error is independent of the chosen time step (computations are performed within one grid cell). It has been observed that the predictor advection algorithm is sufficient. It is recommended to apply the predictor-corrector procedure in coarse grids.
- It is sometimes argued that particle models lack numerical diffusion. However, particle displacements are computed numerically using discrete flow information. Also the time integration is performed with a first order Euler scheme. So it is not obvious that the particle model lacks numerical diffusion altogether. Since the particle approximations yield $\tau > 95$s in case of space-varying diffusivity, it is reasonable to conclude that the numerical diffusion present in the particle model is less than the numerical diffusion present in the finite difference model. It is recommended to compute physical quantities at particle locations with a higher order interpolation scheme.
- The reinitialization procedure may introduce numerical diffusion. It is however observed that this is hardly quantified in the experiments - see experiment (1), (2) and (3). It is stated that the grid cell dimensions are small enough to suppress artificial spreading. The reinitialization procedure that duplicates the computational particles at reinitialization occurrences is sensitive for rounding-off errors. It has been observed that numerically

$$\sum_{p=1}^{\alpha n_p^0} \mu_p \neq \sum_{p=1}^{2\alpha n_p^0} ½\mu_p \qquad (4.151)$$

  Consequently, the reinitialization procedure does not conserve heat. The relaxation time increases if heat is artificially generated (i.e., $\sum_{p=1}^{\alpha n_p^0} \mu_p < \sum_{p=1}^{2\alpha n_p^0} ½\mu_p$). It is recommended to consider double precision calculations if the number of particles times the machine-dependent rounding-off error exceeds a value of 0.05. For instance, experiment (4) with constant diffusivity executed with double precision yields $\tau = 94.84$s (0.309).
- Although the procedure "diffusion followed by advection" is incorrect it is hardly expressed in the computed mean relaxation time.

- Finite differencing versus particle tracking

The comparison between finite difference approximations and particle approximations involves accuracy and effectiveness. It is concluded that finite difference approximations are often obscured by numerical diffusion. However, the fourth order time splitting (advection) scheme contains minimal numerical diffusion. This is expressed by a relaxation time that is slightly less than the relaxation time produced by particle tracking. The particle approximations of relaxation times are as reliable as finite difference approximations with minimal numerical diffusion. The heat distribution computed with particle tracking shows a considerable scatter, while the finite difference approximations remain smooth. Particle approximations become smooth if a large amount of particles is released. This is realized by (i) a single experiment with large n_p^0 or (ii) a repeated experiment with small n_p^0 after which the obtained realizations are averaged. It is reminded that the choice of representation of the discrete distribution also affects the smoothness of the distribution - see section 4.3.1. Smooth distributions are expected if the representation is performed with e.g. point spread functions.

Another aspect involves computation time. A large number of particles gives rise to an excessive computation time. The fourth order finite difference scheme with Δt = 0.02s requires approximately 1 hour of computation time on a HP9000/720 work station with a 32-bits PA-RISC processor. Execution of particle experiment (1) lasts 4½ hours, while experiment (4) needs 44 hours of computation time. Only experiment (3) is comparable with the finite difference computations (50 minutes). It is expected that competitiveness of particle models with respect to computation time is far more improved if vector/parallel computers are used.

Experiment II: Salt intrusion in the Mekong estuary

Because of the relatively rapid mixing between estuary water and the open sea, enhanced by tidal action, the hydrodynamics of the water movement in regions where rivers and sea water meet and interact, is of great interest. Physical processes acting in estuaries are described in DYER (1977).

By quantifying the mixing processes the estuarine characteristics are predicted. For that purpose, the equation of salt continuity is derived taking into account the salt (as well as water) flux caused by the mean flow (advection), and the salt flux caused by short period eddies (eddy diffusivity). The resulting equation is again a three-dimensional advection-diffusion equation. This equation is ready to be simulated as soon as the velocity components and eddy diffusivities are known. Unfortunately, the eddy diffusion terms are all unknown. Besides the problem of eddy diffusivities the boundary condition at the estuary entrance in tide-driven flows is not obvious. Over a certain period, called the return period, the salt concentration at the estuary entrance will be affected by the concentration during outflow. This memory effect will be modelled by the so-called Thatcher-Harleman boundary condition - see THATCHER and HARLEMAN (1972). Doing so, during high water the salt concentration at the estuary entrance is set to a fixed value. During ebb tide the outflow is modelled with a weak boundary condition. During the change from outflow to inflow the concentration at the estuary entrance is gradually adjusted to the concentration at high water according a cosine profile. The time needed to adjust to the high water concentration is called the constituent return time τ_c. Strong mixing at sea implies a short τ_c, while little salt refreshment gives rise to a larger τ_c.

The objective of this experiment is to demonstrate the boundary procedure of section 4.3.6. It will be shown that particle models are feasible in producing solutions that satisfy complicated boundary conditions. It is not claimed that particle models are preferred to simulate this type of physical problems, since the computational effort is large and less robust in comparison with a traditional finite difference model.

The application deals with the Mekong estuary in the south-east of Vietnam. The estuary ranges from Cho Lach to Vam Kenh, where the estuary entrance is found. The length of the estuary is approximately 67100 m. Time-varying field data (salt concentrations) are available at four locations (My Tho, Xuan Hoa, Gia Hoa, Vam Going).

To obtain an appropriate formulation of the particle model the estuary is assumed sectionally homogeneous. So the model becomes one-dimensional. The estuary is schematized by 28 nodes, i.e., 27 intervals of varying length. The schematization is depicted in figure (4.46). Field data are available at node 6, 10, 12 and 18. The interval lengths, $\Delta x = O(2500m)$, and the cross sections of the estuary, $A(x,t) = O(10000m^2)$, are available at grid nodes (field measurements). A significant initial salt concentration has been measured - see figure (4.47a). The velocity components, $u(x,t)$, are computed at the centre of each grid cell and at the (open) boundary locations, with the hydrodynamic model WENDY, using a one-dimensional framework. Computational results are summarized in figure (4.48a). The eddy diffusion coefficients are obtained by running an optimal control off-line program called OBZEDT. Both software packages were made available by Delft Hydraulics. The optimal control program treats the coefficients as parameters by taking into account the observed salt concentration data. For that purpose the space-varying eddy diffusivity is written in parametric form (with $x=0$ at node 1, estuary exit),

$$D(x,t) = \tilde{\alpha}_1 \exp\left[\alpha_2 x\right] \tag{4.152}$$

The optimal control program yields the following estimates: $\tilde{\alpha}_1 = 111.53\mathrm{m}^2/\mathrm{s}$ and $\alpha_2 = 3.4216\ 10^{-6}\mathrm{m}^{-1}$ - see figure (4.47b). It is noted that large values of D already demonstrate the crude schematization of the estuary.

- Description of particle model; consistency with salt balance

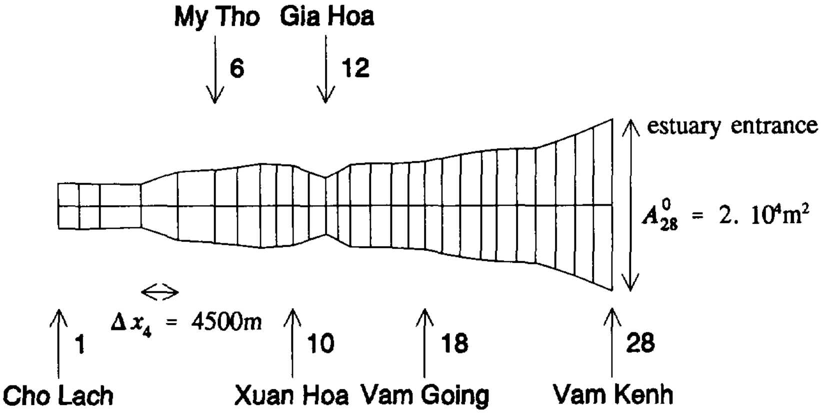


Figure (4.46): Interval lengths versus cross sections of the Mekong estuary at initial state. Node numbers and corresponding place-names are included. Measured concentration profiles are available at node 6, 10, 12 and 18. The estuary entrance is located at node 28.

For a sectionally homogeneous estuary the following advection-diffusion equation is valid

$$\frac{\partial}{\partial t}[A(x,t)C(x,t)] + \frac{\partial}{\partial x}[Q(x,t)C(x,t)] = \frac{\partial}{\partial x}\left[D(x,t)A(x,t)\frac{\partial}{\partial x}C(x,t)\right] \tag{4.153}$$

with C the salt concentration, Q the discharge through cross section A, and D the eddy diffusivity. The local flow velocity is defined as

$$u(x,t) = \frac{Q(x,t)}{A(x,t)} \tag{4.154}$$

The one-dimensional analogue of the Fokker-Planck equation (4.6) of section 4.2.1, is

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}(A_1 p) + \frac{\partial^2}{\partial x^2}(B_{11} p) \tag{4.155}$$

which corresponds with the stochastic differential equation

$$dX(t) = h(X,t)\, dt + g(X,t)\, dW(t) \tag{4.156}$$

The simulation of the transport model is performed by matching equations (4.153) and (4.155). It results in the following expression for the drift, h, and noise, g - see BOOGAARD *et al.* (1993):

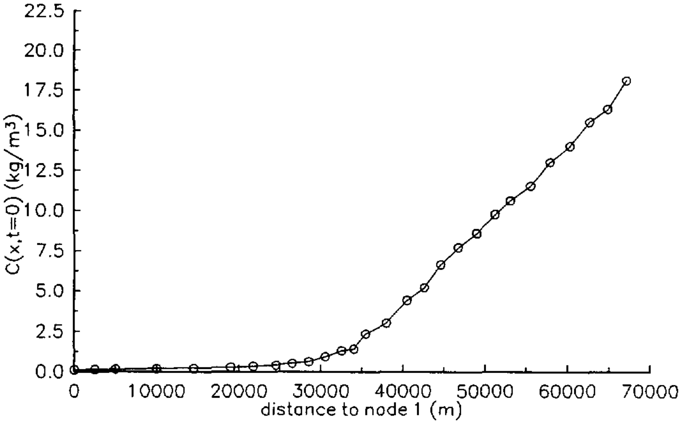


Figure (4.47a): Initial concentration in Mekong estuary, measured at 28 nodes.

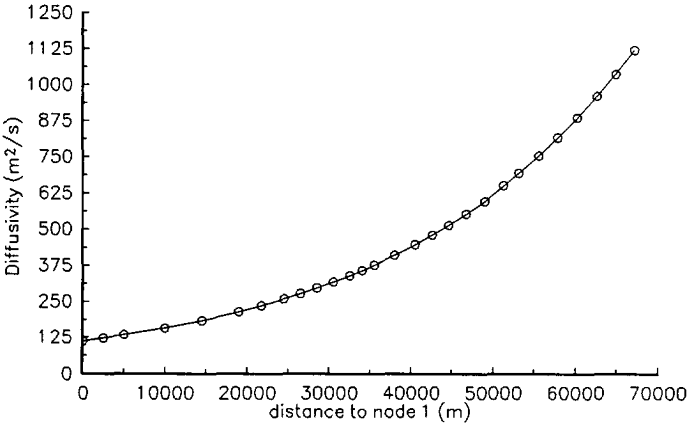


Figure (4.47b): Space-varying eddy diffusivity obtained with the optimal control program OBZEDT.

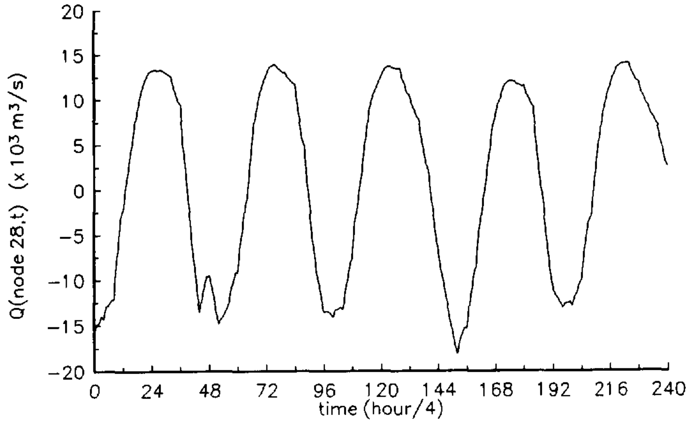


Figure (4.48a): Discharge through cross section at node 28.

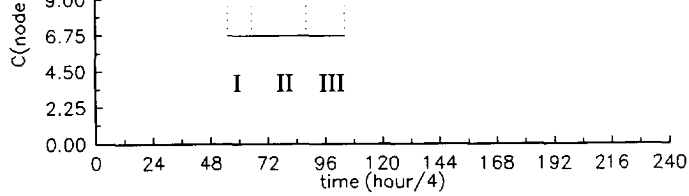


Figure (4.48b): Concentration as a function of time at node 28 (Thatcher-Harleman boundary condition).

I high water,

II ebb tide,

III outflow → inflow (duration determined by constituent return time τ_c).

$$\begin{cases} h(x,t) = \dfrac{Q(x,t)}{A(x,t)} + \dfrac{\partial D(x,t)}{\partial x} + D(x,t)\dfrac{\dfrac{\partial A(x,t)}{\partial x}}{A(x,t)} \\ g(x,t) = \sqrt{2D(x,t)} \end{cases} \tag{4.157}$$

The drift consists of a local flow velocity contribution, with in addition the noise-induced drift and a geometry correction term. If the one-dimensional geometry is transformed to a geometry with equidistant grid cells, complex curvature terms must also be taken into account, as explained in section 4.2.3. Numerical approximation of such curvature terms is inaccurate and does not serve the feasibility of the particle model. Therefore the simulation is totally performed in the physical space.

- Simulation of boundary conditions

At the estuary exit, node 1, a Dirichlet boundary condition is prescribed: $C_1(t) = 0.1\text{kg/m}^3$. At the estuary entrance, node 28, the Thatcher-Harleman boundary condition is prescribed. Three situations are distinguished:

high water $C_{28}(t) = C_{28}^{\text{hw}} = 21\text{kg/m}^3$

ebb tide $\dfrac{\partial^2 C}{\partial x^2} = 0$

outflow → inflow $C_{28}(t) = C_{28}^{\text{lw}} + \left(C_{28}^{\text{hw}} - C_{28}^{\text{lw}}\right)\cos\left[\tfrac{1}{2}\pi - \tfrac{1}{2}\pi(t-t^{\text{lw}})/\tau_c\right]$

The concentration value C_{28}^{lw} equals the concentration at the end of the ebb tide period. The constituent return time τ_c is set to 225 min. The so-defined boundary condition is sketched in figure (4.48b) simulated with the finite difference model.

Particle simulation, including boundary conditions, is performed similar as demonstrated in section 4.4.3. At node 1 the estimated mass outflow equals

$$F_1^{n+1} = -\tfrac{1}{2}\Delta t \left[\left(Q^n C^n + Q^{n+1} C^{n+1}\right) - \left(A^n D^n \frac{\partial C^n}{\partial x} + A^{n+1} D^{n+1} \frac{\partial C^{n+1}}{\partial x}\right)\right]_{\text{node 1}} \tag{4.158}$$

In equation (4.158) $\partial C/\partial x$ at node 1 needs to be estimated. All other quantities are known. F_1^{n+1} determines the intensity of the source at the estuary exit.

At node 28 the estimated mass outflow is written as

$$
\begin{aligned}
F_{28}^{n+1} = \tfrac{1}{2}\Delta t \Bigg[& A^{n}\left(u^{n}C^{n} - D^{n}\left(\frac{R_3^{n} - R_2^{n}C^{n}}{R_1^{n}}\right)\right) \\
& + A^{n+1}\left(u^{n+1}C^{n} - D^{n+1}\left(\frac{R_3^{n+1} - R_2^{n+1}C^{n}}{R_1^{n+1}}\right)\right)\Bigg]_{\text{node28}}
\end{aligned}
\tag{4.159}
$$

in which $\left(R_3 - R_2 C\right)/R_1$ denotes an approximation of $\partial C/\partial x$. For instance, during ebb tide the advection-diffusion equation is written as

$$
\underbrace{\left(Q - \frac{\partial}{\partial x}(DA)\right)\frac{\partial C}{\partial x}}_{R_1} + \underbrace{\frac{\partial Q}{\partial x}}_{R_2}\ C = \underbrace{-\frac{\partial}{\partial t}(AC)}_{R_3} + DA\frac{\partial^2 C}{\partial x^2}
\tag{4.160}
$$

Substitution of the prescribed boundary condition into equation (4.160) yields expressions for R_1, R_2 and R_3 respectively. Finally, estimation of these coefficients determines the intensity of the source at the estuary entrance.

If necessary, the estimation of the concentration derivative at the boundary is performed using a regression technique. It is noted that the expression suggested in equation (4.141) of section 4.4.3 is not appropriate, since the grid cell dimensions are too large. The regression procedure runs as follows. (1) Divide the grid cell touching the boundary into N_R equidistant cells. This number is based on judgement. On account of numerical experiments this number is selected such that each (sub) cell covers 500 particle positions ($N_R = 1$ is the minimum). (2) Within these N_R cells convert the mass distribution into a (cell averaged) concentration distribution. (3) Determine the regression line. (4) Determine the concentration and the concentration derivative using this regression line. If the estimated concentration value reaches outside an acceptable range specified by the prescribed (Dirichlet) concentration value the procedure is repeated with a smaller N_R. A decrease of N_R results in a smoother distribution.

A regression technique has also been used to determine C_{28}^{lw} at time $t = t_n$ as it arises in the Thatcher-Harleman boundary condition

$$
\begin{aligned}
C_{28}^{\text{lw}} = \overline{C}^{\text{reg}} + \tfrac{1}{2}\Big(& -1.5\left(C_{28}^{n-4} - \overline{C}^{\text{reg}}\right) - 0.5\left(C_{28}^{n-3} - \overline{C}^{\text{reg}}\right) \\
& + 0.5\left(C_{28}^{n-2} - \overline{C}^{\text{reg}}\right) + 1.5\left(C_{28}^{n-1} - \overline{C}^{\text{reg}}\right)\Big)
\end{aligned}
\tag{4.161}
$$

with

$$\overline{C}^{reg} = \frac{1}{4}\left(C_{28}^{n-4} + C_{28}^{n-3} + C_{28}^{n-2} + C_{28}^{n-1}\right) \tag{4.162}$$

- Model results

The numerical experiments are executed with time step Δt = 3.0min. The initial number of particles, n_p^0, is set to 160000. Various concentration distributions are compared with finite difference results (WENDY) and observed data. It is emphasized that the diffusion coefficient is chosen such that the finite difference simulation shows a well correspondence with the observed data. The results of the particle model should correspond with the solutions produced by the finite difference model.

Computed concentration profiles as well as observed field data are summarized in figure (4.49). The outcome of the particle model with respect to a grid cell is plotted against the concentration values given at grid nodes obtained by the finite difference model. It is reasonable to expect that the computed cell-averaged concentration values should be enclosed by the concentration values at the adjacent nodes. The computed concentration distributions observed in the interior of the estuary shows a good resemblance with observed concentrations. Near the estuary entrance particle model results deviate from the corresponding finite difference results. This deviation is acceptable for small simulation times - say two tide cycles - see figure (4.49h). However, as seen in figure (4.50b) and figure (4.50c), the weak boundary condition is hard to reproduce. In contrast the resemblance in case of a Dirichlet boundary is remarkable - see figure (4.50a). At node 1 (estuary exit), with approximately 30 particles in cell 1, the particle model reproduces the finite difference solutions fairly well. Other numerical experiments focusing on the number of particles, the time step and the accuracy of the regression technique, showed that complicated boundary conditions are satisfied in particle models only if the mass flux through the boundary is accurately estimated. The simulation is very sensitive for inaccurate approximations of the coefficients R_1, R_2 and R_3. The use of regression techniques benefits the robustness of the model. In applications with crude geometry schematization and complicated boundary conditions the particle model is applicable but not recommendable.

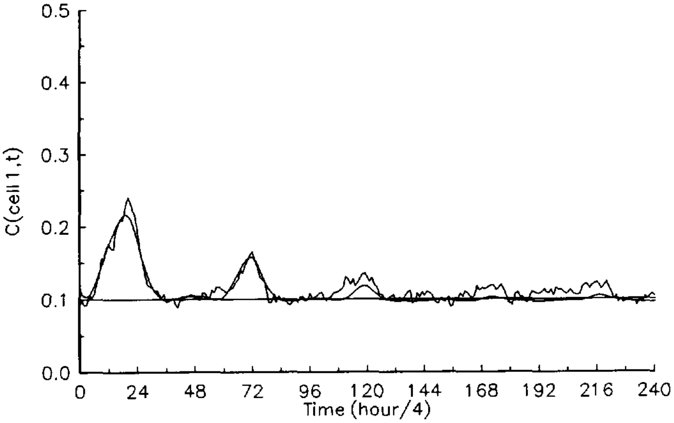


Figure (4.49a): Concentration distribution at node 1 and 2.

scattered line: mean concentration in cell 1 (particle model). This value is enclosed by the concentration at node 1 and node 2 given by straight lines (finite difference model).

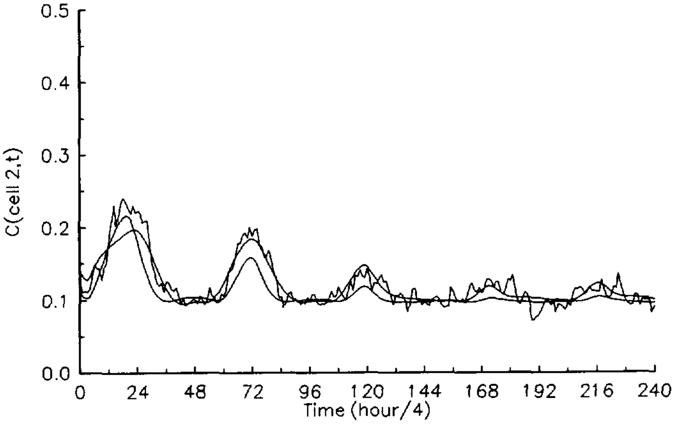


Figure (4.49b): Concentration distribution at node 2 and 3.

scattered line: mean concentration in cell 2 (particle model). This value is enclosed by the concentration at node 2 and node 3 given by straight lines (finite difference model).

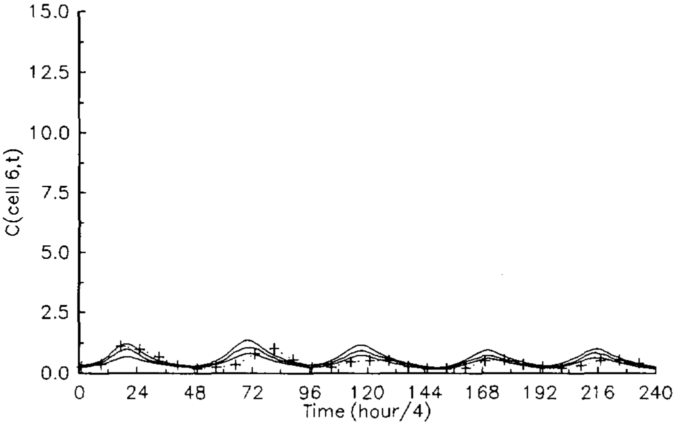


Figure (4.49c): Concentration distribution at node 6 and 7.

scattered line: mean concentration in cell 6 (particle model). This value is enclosed by the concentration at node 6 and node 7 given by straight lines (finite difference model). plus: observed concentrations at node 6.

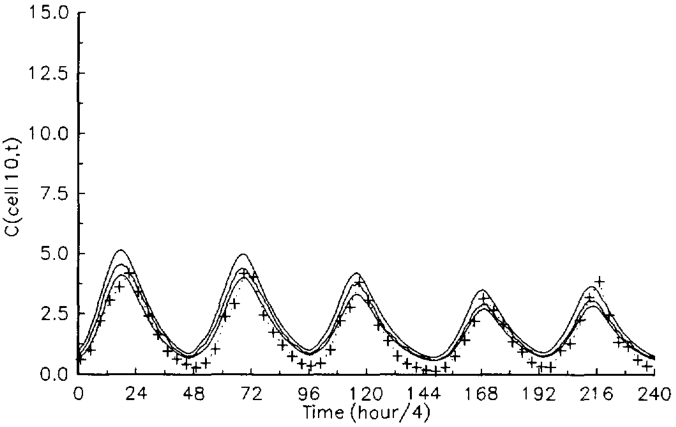


Figure (4.49d): Concentration distribution at node 10 and 11.

scattered line: mean concentration in cell 10 (particle model). This value is enclosed by the concentration at node 10 and node 11 given by straight lines (finite difference model). plus: observed concentrations at node 10.

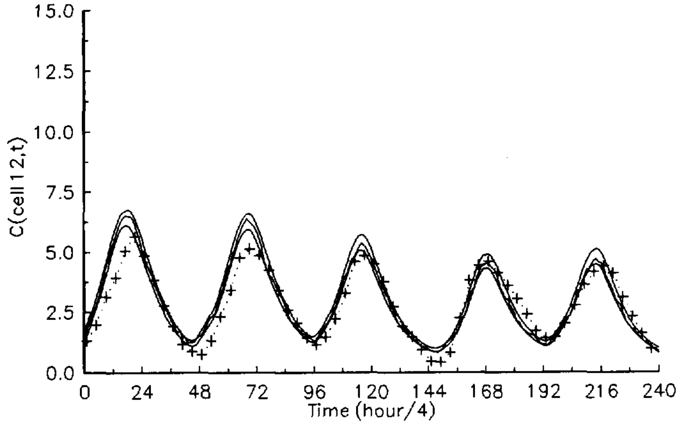


Figure (4.49e): Concentration distribution at node 12 and 13.

scattered line: mean concentration in cell 12 (particle model). This value is enclosed by the concentration at node 12 and node 13 given by straight lines (finite difference model). plus: observed concentrations at node 12.

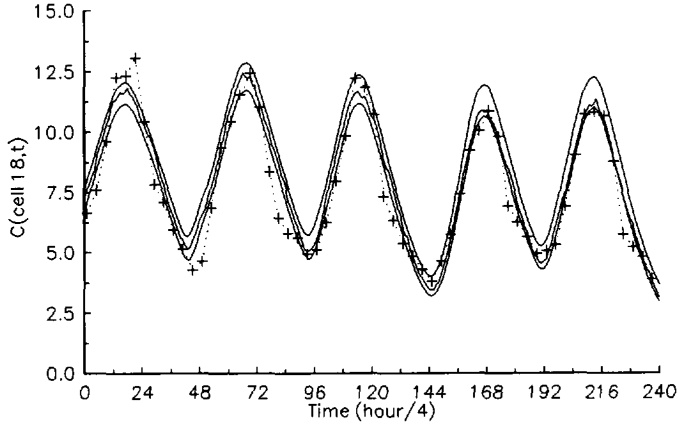


Figure (4.49f): Concentration distribution at node 18 and 19.

scattered line: mean concentration in cell 18 (particle model). This value is enclosed by the concentration at node 18 and node 19 given by straight lines (finite difference model). plus: observed concentrations at node 18.

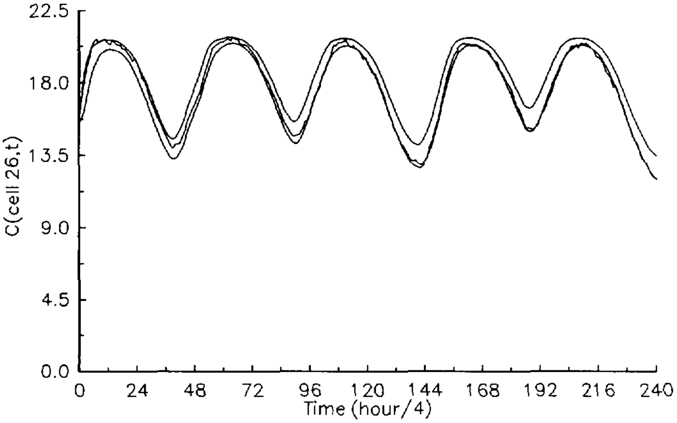


Figure (4.49g): Concentration distribution at node 26 and 27.

scattered line: mean concentration in cell 26 (particle model). This value is enclosed by the concentration at node 26 and node 27 given by straight lines (finite difference model).

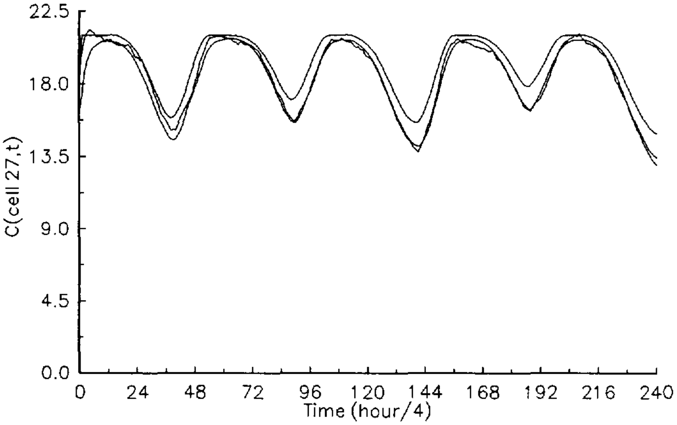


Figure (4.49h): Concentration distribution at node 27 and 28.

scattered line: mean concentration in cell 27 (particle model). This value is enclosed by the concentration at node 27 and node 28 given by straight lines (finite difference model).

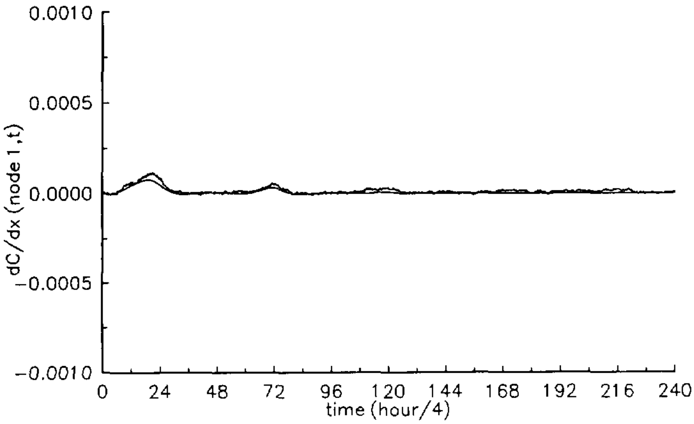


Figure (4.50a): Estimated concentration derivative at node 1 in particle model used in boundary procedure (scattered line), compared with finite difference result (straight line).

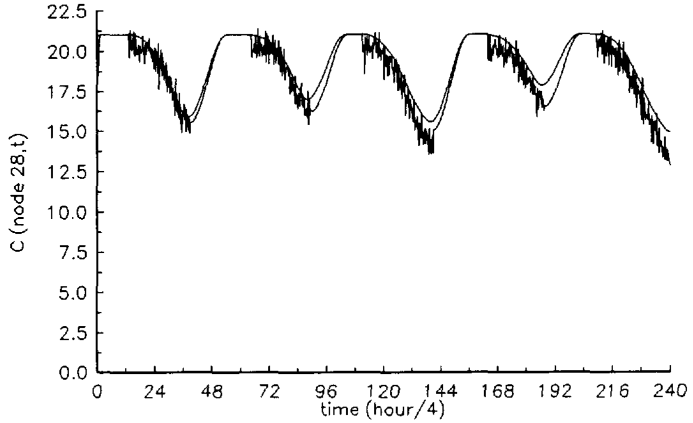


Figure (4.50b): Estimated concentration at node 28 in particle model used in boundary procedure (scattered line), compared with finite difference result (straight line).

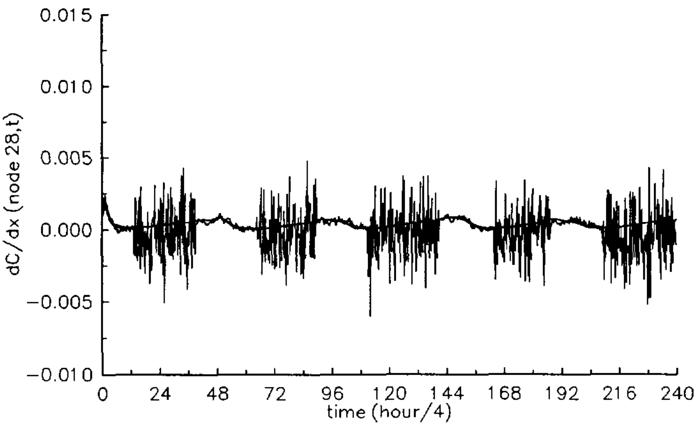


Figure (4.50c): Estimated concentration derivative at node 28 in particle model used in boundary procedure (scattered line), compared with finite difference result (straight line).

Chapter 5

The combined Particle/Euler transport model

5.1 Introduction

Transport phenomena that are described with the advection-diffusion equation are numerically solved in various ways. Each solution technique is developed such that some predefined physical and computational requirements are satisfied. These requirements involve accuracy, computation time, conservation (of mass), positivity, effects of discontinuities in source terms and/or boundary conditions, representation of shock waves, and resolution. Finite difference methods are extensively investigated and it is concluded that each phenomenon needs its own numerical solution technique to obtain optimal results - see VREUGDENHIL and KOREN (1993) or POURQUIÉ (1990). In the preceding chapter the particle method has been studied. The particle method is above all extremely suitable in discharge applications. The method lacks numerical diffusion and guarantees positivity. A serious drawback to be mentioned is its possible excessive computation time and its robustness handicap in boundary treatment. In this chapter it is tried to combine the favourable aspects of the particle model with those of the finite difference approach. It results in a procedure that minimizes the adverse consequences of the individual methods. For instance, in regions where the concentration distribution is smooth the finite difference method is applied (far field simulation), while in regions where steep concentration gradients are present the particle method will be applied (near field simulation).

Advantages and disadvantages of numerical methods in the simulation of continuous discharge problems are listed below. This somewhat crude generalized discrimination may provoke discussion. It is not suggested that all finite difference methods show these so-called disadvantageous aspects. However, to minimize, for instance, grid dependency it is stated that a special numerical treatment is necessary.

A possible combined particle/Euler method can be constructed such that particle simulations and finite difference approximations are treated successively. Then, initially a continuous discharge problem is simulated with the particle model. In course of time when the concentration distribution has become smooth the computations are continued with a finite difference method - see BLOKLAND and HEEMINK (1993). This type of particle/Euler methods is not considered here, since in the vicinity of the discharge location the concentration distribution will never become smooth. A steep concentration gradient will always be present. A particle/Euler method will be described that simulates the transport process with a particle model and a finite difference model simultaneously.

	give credit for (advantages)	blame for (disadvantages)
particle methods	- applicability in coarse grids - minimization of numerical diffusion - strict positivity - accurate advection simulation - easy implementation	- large computational effort and memory storage - discrete distributions - poor robustness features when imposing complicated boundary conditions
finite difference methods	- computational efficiency - smooth distributions	- grid dependency - numerical diffusion - spurious oscillations

5.2 Particle/Euler description

The combined particle/Euler transport model splits the concentration distribution, C, into two contributions: a background concentration C^E and a foreground concentration C^P, such that

$$C^E + C^P = C \tag{5.1}$$

C^E is associated with the contribution of the finite difference approximation, while C^P represents the contribution of the particle model. The procedure will be outlined with respect to the following (generalized) one-dimensional advection-diffusion equation

$$\begin{cases} \dfrac{\partial C}{\partial t} + u\dfrac{\partial C}{\partial x} + \gamma C = \dfrac{\partial}{\partial x}\left(D\dfrac{\partial C}{\partial x}\right) + S & x \in \Omega \quad t > t_0 \\ C(x,t_0) = C_I(x) & x \in \Omega \quad t = t_0 \\ C(r,t) = b_D(r,t) & r \in \partial\Omega \quad t \geq t_0 \end{cases} \tag{5.2}$$

Equation (5.2) models advection, eddy diffusion and exponential decay of a passive contaminant in a domain Ω with boundary $\partial\Omega$ involving source terms. A Dirichlet boundary condition is recognized. The reduction to one dimension is not essential. The extension to three dimensions is straightforward.

To make relation (5.1) practicable the procedure assumes a linear advection-diffusion equation. Two alternative particle/Euler methods are distinguished. The first alternative is constructed such that the particle method contributes for the source term, while the finite difference model accounts for boundary conditions. The second alternative starts with the particle model with absorbing boundaries. The particle model solution is partially regarded as a source term in the finite difference model. So the finite difference model is activated by sources of particles and by boundary conditions.

Subsequently, the corresponding mathematical descriptions are summarized.

Alternative I:

Due to linearity equation (5.2) is taken apart into two sets of equations, each defining C^P and C^E:

$$\begin{cases} \dfrac{\partial C^P}{\partial t} + u\dfrac{\partial C^P}{\partial x} + \gamma C^P = \dfrac{\partial}{\partial x}\left(D\dfrac{\partial C^P}{\partial x}\right) + S \\ C^P(x,\tau) = C_{IP}^{\tau}(x) \\ C^P(r,t) = 0 \end{cases} \tag{5.3a}$$

$$\begin{cases} \dfrac{\partial C^E}{\partial t} + u\dfrac{\partial C^E}{\partial x} + \gamma C^E = \dfrac{\partial}{\partial x}\left(D\dfrac{\partial C^E}{\partial x}\right) \\ C^E(x,\tau) = C_{IE}^{\tau}(x) \\ C^E(r,t) = b_D(r,t) \end{cases} \tag{5.3b}$$

At specified time $t = \tau \geq t_0$ the concentration distribution is spread over the particle simulation and the finite difference simulation. This procedure defines a "new" initial concentration distribution for the individual algorithms satisfying

$$C_{IP}^{\tau}(x) + C_{IE}^{\tau}(x) = C(x,\tau) \tag{5.3c}$$

The contribution C_{IE}^{τ} is chosen such that certain requirements about the smoothness of the concentration distribution C^E are guaranteed.

To solve equation (5.3b) spatial and time coordinates are discretized. For that purpose Δx and Δt are specified. Equation (5.3a) is solved as in chapter 4 by computation of particle displacements travelling for a period Δt with a particle withdrawal at the boundary. Each time step, at $\tau = t_n = t_0 + n\Delta t$, relation (5.3c) is evaluated. The smoothness requirement is given by

$$\frac{\Delta x}{C_{IE}^{t_n}}\left|\frac{\partial C_{IE}^{t_n}}{\partial x}\right| < \delta \tag{5.4}$$

with δ a small dimensionless model parameter. This inequality provides that first order truncation errors are small, i.e., $O(\delta)$. An inconvenient side effect of the algorithm is that each time step a reinitialization procedure is required. It has been observed in chapter 4 that reinitialization procedures for the particle method may introduce artificial spreading. Therefore, another alternative is proposed which makes reinitialization redundant.

Alternative II:

This second alternative particle/Euler procedure is similar to alternative I with the exception of the reinitialization procedure. Particle model simulations contribute as a source term in the finite difference formulation

$$\begin{cases} \dfrac{\partial C^P}{\partial t} + u\dfrac{\partial C^P}{\partial x} + \gamma C^P = \dfrac{\partial}{\partial x}\left(D\dfrac{\partial C^P}{\partial x}\right) + S - S^P \\ C^P(x,t_0) = C_{I^P}(x) \\ C^P(r,t) = 0 \end{cases} \tag{5.5a}$$

$$\begin{cases} \dfrac{\partial C^E}{\partial t} + u\dfrac{\partial C^E}{\partial x} + \gamma C^E = \dfrac{\partial}{\partial x}\left(D\dfrac{\partial C^E}{\partial x}\right) + S^P \\ C^E(x,t_0) = C_{I^E}(x) \\ C^E(r,t) = b_D(r,t) \end{cases} \tag{5.5b}$$

The initial conditions for the individual simulations satisfy

$$C_{I^P}(x) + C_{I^E}(x) = C_I(x) \tag{5.5c}$$

Again the finite difference method contributes for boundary conditions, while the source term S is implemented in the particle model. The source term S^P, introduced by the model, is implemented as a particle withdrawal in the particle simulation and as a corresponding mass injection in the finite difference simulation. If a particle is withdrawn from a particular cell its mass gives rise to a source S^P. The intensity of this source is determined by the particle mass divided by the grid cell volume. The criterium for particles to be withdrawn is based upon an equation similar to (5.4)

$$\frac{\Delta x}{C^E}\left|\frac{\partial C^E}{\partial x}\right| < \delta \tag{5.6}$$

Inequality (5.6) expresses that steep gradients with respect to C^E are to be avoided.

5.3 Particle/Euler simulations

The particle/Euler method shows the full advantage in advection dominated flows applied to either steep initial concentration distributions or continuous discharges. Therefore, two numerical experiments are executed to illustrate the effectiveness of the method.

Experiment I: One-dimensional square wave test

The presence of numerical diffusion and spurious oscillations in finite difference methods is very well illustrated by the (one-dimensional) square-wave test. This test simulates the advection of an initial step concentration distribution. A first order upwind scheme is strictly positive, but produces a noticeable numerical diffusion at the back and front of the square wave. The solution of a second order upwind method is dominated by spurious oscillations. Negative concentrations are produced. Oscillations grow in amplitude as time evolves and the method becomes unstable. A filter technique to avoid negative concentrations suppresses the amplitude of the oscillations, which results in a stable solution technique. Still, the front and back of the square wave are smoothed showing that numerical diffusion is introduced. These effects become less severe in cases with gradually space-varying concentrations. For smooth initial distributions the error is dominated by the Courant number instead of the (steep) concentration gradient. The influence of the smoothness of the concentration distribution on the absolute error is quantified in a simple dimensionless advection test. The initial concentration is given in figure (5.1). The smoothness of the distribution is controlled by a parameter r. If r tends to zero the square wave test is obtained. The advection equation is given by

$$\frac{\partial C}{\partial t} + \frac{\partial q}{\partial x} = 0 \qquad \text{with} \qquad q = uC \tag{5.7}$$

Equation (5.7) is solved by means of finite differencing. With $C_i^n = C(i\Delta x, n\Delta t)$ the discretized analogue of (5.7) may be

$$\frac{C_i^{n+1} - C_i^n}{\Delta t} + \frac{q_i^n - q_{i-1}^n}{\Delta x} = 0 \tag{5.8}$$

For purpose of demonstration u is set to one. The following explicit finite difference schemes are used to demonstrate characteristic numerical aspects:

(i) First order upwind

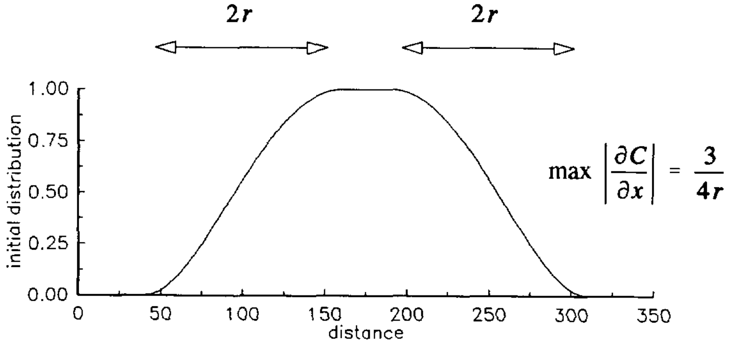


Figure (5.1): Initial concentration distribution. Smoothness is controlled by r.

$$q_i = C_i \tag{5.9a}$$

(ii) Second order upwind without filter

$$q_i = \tfrac{1}{2}(3C_i - C_{i-1}) \tag{5.9b}$$

(iii) Second order upwind with filter (avoiding local maxima and minima)

$$q_i = \begin{cases} C_i & \{\tfrac{1}{2}(3C_i - C_{i-1}) > C_{i+1} \wedge C_{i-1} < C_i\} \\ & \vee \{\tfrac{1}{2}(3C_i - C_{i-1}) < C_{i+1} \wedge C_{i-1} > C_i\} \\ \tfrac{1}{2}(3C_i - C_{i-1}) & \text{elsewhere} \end{cases} \tag{5.9c}$$

More accurate results are to be expected from higher order finite difference schemes. This is illustrated by means of a fourth order semi-implicit time splitting scheme with filter:

(iv) Fourth order time-splitting method with Forester filter. This procedure is accomplished per half time step. It is defined as

Step 1:
$$\frac{C_i^{n+1/2} - C_i^n}{\tfrac{1}{2}\Delta t} + \frac{q_i^{n+1/2} - q_{i-1}^{n+1/2}}{\Delta x} = 0 \text{ with } q_i = \frac{10C_i - 5C_{i-1} + C_{i-2}}{6} \tag{5.9d-1}$$

Step 2:
$$\frac{C_i^{n+1} - C_i^{n+1/2}}{\tfrac{1}{2}\Delta t} + \frac{q_i^{n+1/2} - q_{i-1}^{n+1/2}}{\Delta x} = 0 \text{ with } q_i = \frac{C_{i+1} + C_i}{2} \tag{5.9d-2}$$

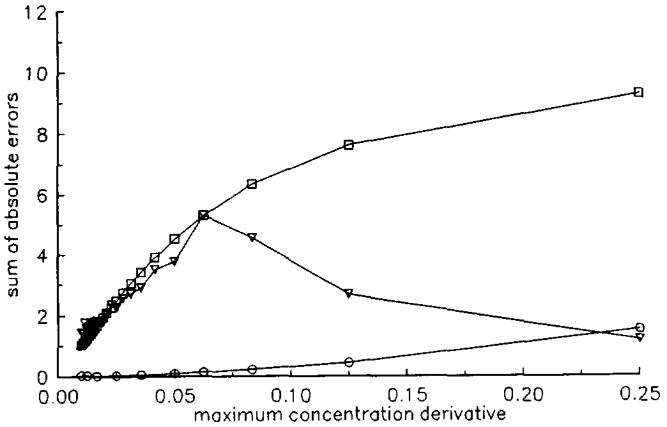


Figure (5.2a): Absolute errors ($\sum_i |\bar{C}_i - C_i|$, where a bar indicates the numerical method) of advection tests that distinguish between the smoothness of the initial concentration. The experiments are executed with a Courant number = 0.5.

box: first order upwind - experiment (i),
triangle down: second order upwind with filter - experiment (iii),
circle: fourth order time splitting with Forester filter - experiment (iv).

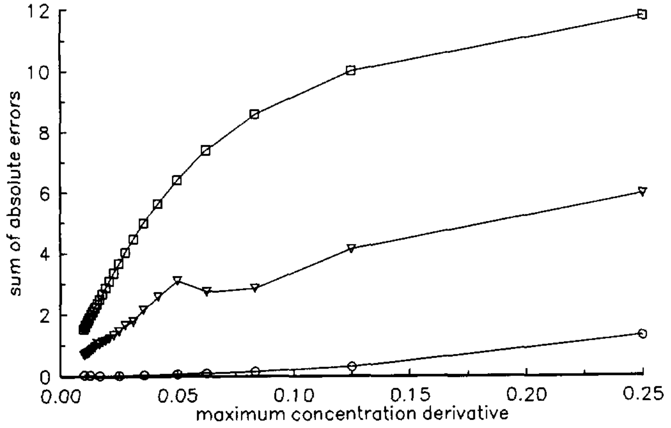


Figure (5.2b): Absolute errors ($\sum_i |\bar{C}_i - C_i|$, where a bar indicates the numerical method) of advection tests that distinguish between the smoothness of the initial concentration. The experiments are executed with a Courant number = 0.25. Legend as in figure (5.2a).

The upwind scheme (5.9d-1) and the central scheme (5.9d-2) are chosen such that the total scheme, obtained by elimination of $C_i^{n+\frac{1}{2}}$, results in a fourth order scheme. In addition, after each step, a so-called Forester filter is applied to suppress spurious waves. This filter, described in FORESTER (1977), is only active in regions where spurious waves are generated and propagated by the numerical scheme. So, locally the distribution is smoothed by the introduction of numerical diffusion. The Forester filter is defined as an iterative conservative smoothing process given by

$$C_i^{[p+1]} = C_i^{[p]} + \frac{\mu}{2}\left((C_{i+1} - C_i)(\phi_{i+1} + \phi_i) - (C_i - C_{i-1})(\phi_i + \phi_{i-1})\right)^{[p]}$$

$$C_i^{[0]} = \begin{cases} C_i^{n+\frac{1}{2}} & \text{step } 1 \\ C_i^{n+1} & \text{step } 2 \end{cases} \tag{5.9d-3}$$

with iteration index $p = 0,..,P_{\max} - 1$, diffusion parameter $\mu = 0.25$, and

$$\phi_i = \begin{cases} 1 & C_{i-1} < 0 \ \vee \ C_i < 0 \ \vee \ C_{i+1} < 0 \\ 0 & \text{elsewhere} \end{cases} \tag{5.9d-4}$$

Figure (5.2) shows the absolute error as a function of the maximum (initial) concentration gradient in case of $\Delta x = 1$. Figure (5.3) gives the computed concentration distribution at $T = 100$. The experiments show that - see figure (5.2) and figure (5.3):

- First order approximations result in solutions affected by numerical diffusion.
- In cases where steep gradients are present second order approximations need a filter technique to obtain stable results. For smooth applications the spurious oscillations remain undisturbed.
- Higher order procedures applied in cases with steep gradients show spurious oscillations in the vicinity of the front and back of the distributions. For smooth distributions the approximation resembles the analytical solution extremely well.
- The absolute error is dominated by the concentration gradient. The influence of the Courant number is less important. It is however noted that a first order upwind scheme with Courant number = 1.0 yields exact solutions due to point-to-point transfer.
- Smooth concentration distributions are accurately advected by the fourth order time splitting scheme.

A possible remedy in avoiding the unrealistic computed front (or back) of a steep concentration distribution is known as grid refinement. This procedure is inefficient, since a grid refinement is unnecessary in smooth regions. An alternative remedy is found by the combined particle/Euler method. Alternative I of section 5.2 will be demonstrated. The initial

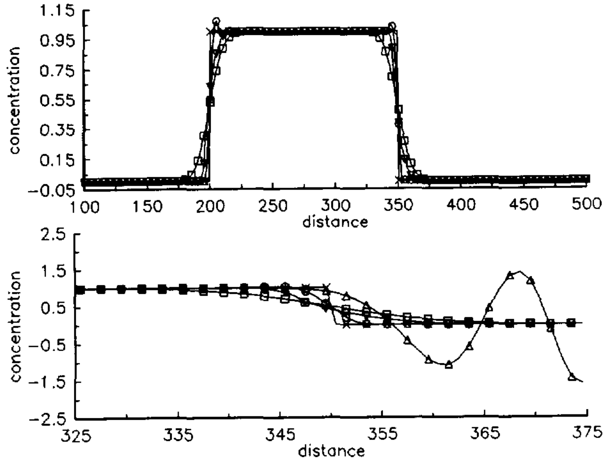


Figure (5.3a): Simulation of advected square wave (TOP) - enlargement (BOTTOM). Courant number = 0.25.

cross:	analytical solution, i.e., shifted initial distribution,
box:	first order upwind - experiment (i),
triangle up:	second order upwind - experiment (ii) (appears only in enlargement),
triangle down:	second order upwind with filter - experiment (iii),
circle:	fourth order time splitting with Forester filter - experiment (iv).

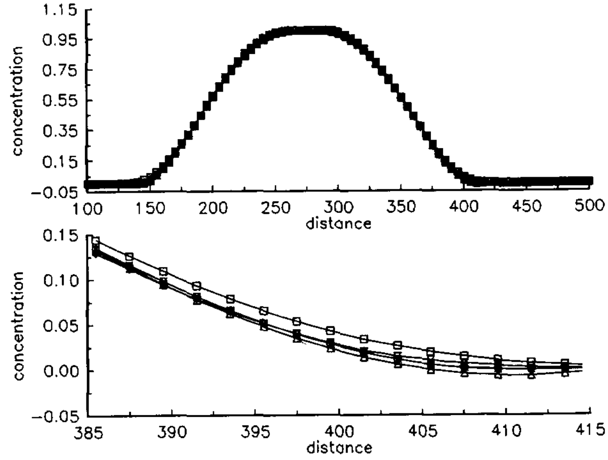


Figure (5.3b): Simulation of advected smooth distribution (TOP) - enlargement (BOTTOM). Maximum derivative equals 0.0125 (r = 60), Courant number = 0.25, Legend as in figure (5.3a).

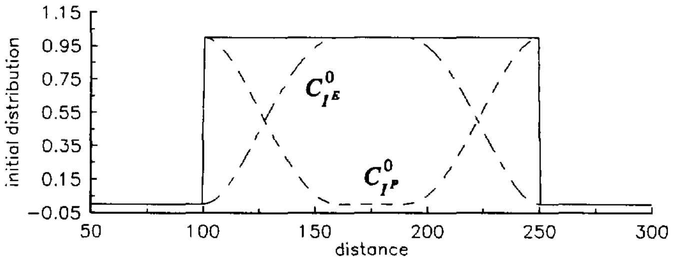


Figure (5.4): Initial (subdivided) distribution of combined particle/Euler method.
$C_I(x) = C^0_{IP}(x) + C^0_{IE}(x)$, max $\left|\partial C^0_{IE}/\partial x\right| = 3/(4r) = 0.025$ $(r = 30)$.

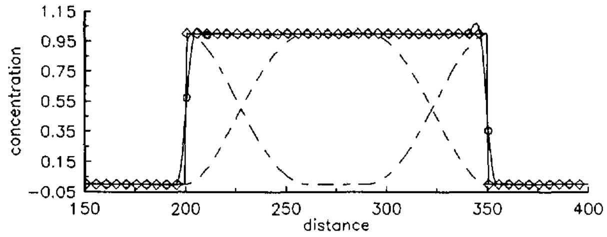


Figure (5.5a): Simulation of advected square wave using the combined particle/Euler method with a fourth order time splitting scheme with Forester filter. Courant number = 0.5.

solid line: analytical square wave,

diamond: particle/Euler simulation,

dashed lines: particle and finite difference contributions,

circle: fourth order time splitting procedure with Forester filter.

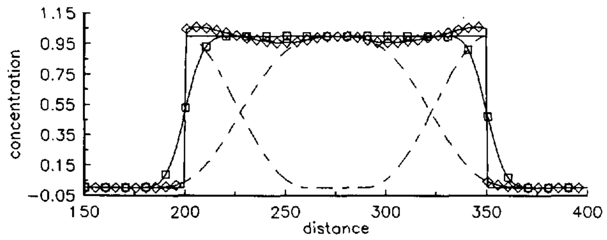


Figure (5.5b): Simulation of advected square wave using the combined particle/Euler method with a first order upwind scheme. Courant number = 0.5. Legend as in figure (5.5a) and

box: first order upwind.

square wave is taken apart as shown in figure (5.4). The finite difference procedure is submitted to the smooth distribution. If the fourth order time splitting method is applied accurate results are obtained. The unrealistic representation of the front and back of the square wave in case of a strict finite difference simulation is removed by the combined method - see figure (5.5a). If the first order upwind scheme is used the numerical diffusion contributes in an unrealistic representation of the square wave - see figure (5.5b). It is concluded that the combined particle/Euler method is only effective if smooth finite difference solutions are hardly affected by numerical diffusion. The fourth order time splitting scheme with Forester filter meets this requirement.

Experiment II: Continuous discharge

Numerical simulations of continuous discharges are especially executed in favour of hot water discharges of hydroelectric power stations. These simulations are more complicated than simulations of e.g., dissolved matter transport due to the noticeable presence of a density current. Even if density influences are of negligible importance, finite difference simulations are easily affected by numerical artifacts. Especially in two- and three-dimensional applications the smoothness effect due to numerical diffusion is troublesome. In continuous discharge applications this phenomenon will always be present in the vicinity of the discharge location. As mentioned the particle model overcomes this inconvenience, but introduces a computational efficiency problem since the number of computational particles increase monotonicly in time. These adverse aspects are tried to overcome with the combined particle/Euler procedure. Here a mathematical-physical description of a two-dimensional discharge in an unbounded domain is considered. In case of constant coefficients the solution is known analytically. It serves the study of qualitative and quantitative numerical aspects. The governing equation becomes

$$\left\{\begin{array}{l} \dfrac{\partial C}{\partial t} + u\dfrac{\partial C}{\partial x} + v\dfrac{\partial C}{\partial y} = D\left(\dfrac{\partial^2 C}{\partial x^2} + \dfrac{\partial^2 C}{\partial y^2}\right) + S(x,y,t) \\ C(x,y,0) = 0 \end{array}\right. \tag{5.10}$$

with constant velocity components u and v, and a constant eddy diffusivity D. The source S is expressed by

$$S = \frac{\hat{S}}{H}\,\delta\left(x-x_0,y-y_0\right) \tag{5.11}$$

with constant $\hat{S}$ in kg/s per unit area and H the constant water depth. The solution of (5.10) equals

$$C(x,y,t) = \frac{\hat{S}}{H}\int_0^t \frac{1}{4\pi D\tau}\exp\left[-\frac{\left(x-x_0-u\tau\right)^2+\left(y-y_0-v\tau\right)^2}{4D\tau}\right]d\tau \tag{5.12}$$

In the combined simulation the particle model accounts for the simulation near the discharge location, while the finite difference model is activated in regions where the concentration allows a smooth background concentration contribution. Alternative II of section 5.2 will be demonstrated, in which the source term S^P as it stimulates the background concentration in cell (m,n), is defined as

$$S^P_{m,n} \text{ (per unit time) } = \begin{cases} \dfrac{\mu_p}{\Delta x\Delta y}\dfrac{1}{H} & C^E \text{ remains smooth near cell } (m,n) \\ 0 & \text{elsewhere} \end{cases} \tag{5.13}$$

The condition "C^E remains smooth" is submitted to various procedures. Two alternatives are discussed:

(i) For each cell one particle, if present, is admitted in the background concentration field. As a consequence, a linear distribution C^E is obtained. Its steepness mainly depends on the Courant number. Small Courant numbers give rise to steep concentration gradients. A serious drawback, especially in advection dominated flows, is that the front of the plume is simulated by the background concentration, since the particle model is active strictly near the discharge location.

(ii) As time evolves the background concentration increases due to the flow characteristics and the definition of the source S^P. With S^P as in (5.13) the condition "C^E remains smooth" is translated into

$$\left|\left(2C^E_{m+i,n+j} + S^P_{m+i,n+j}\Delta t\right) - \left(2C^E_{m,n} + S^P_{m,n}\Delta t\right)\right| < \delta\frac{\hat{S}\,\Delta t}{\Delta x\Delta y H} \tag{5.14}$$

providing $C^P_{m+i,n+j} > 0 \ \wedge \ C^P_{m,n} > 0$, for all $(i,j) \in \{(1,0),(0,1),(-1,0),(0,-1)\}$. The arising concentration values are evaluated at the old time levels. For long simulation times the (smooth) far field concentration field is completely simulated by the finite difference model. Isolated particles are immediately admitted providing $\delta > 1/n_p^0$. The dimensionless parameter δ determines the maximum admissible increase of the (background) concentration per time step. A small δ denotes that C^E is allowed to increase gradually in time. Background concentration gradients are restricted.

The numerical experiments are judged by observing contour plots of constant concentration values. The following numerical and physical data are used:

$u = v = ½\sqrt{2}$ m/s, $D = 0.5\,\mathrm{m}^2/\mathrm{s}$, $\hat{S} = 100$ kg/s per unit area, $H = 5$ m and $T = 400$ s. $\Delta x = \Delta y = 5$ m, $\Delta t = 2$ s, and $n_p^0 = 100$.

The contour plots corresponding to the individual model results are summarized in figure (5.6). The experiments distinguish:

Figure (5.6b): the finite difference simulation using the first order upwind scheme,
Figure (5.6c): the particle simulation, and
Figure (5.6d): the finite difference simulation using the fourth order time splitting scheme with Forester filter.

Figure (5.7) shows the results of the combined particle/Euler model with $\delta = 0.05$ as it arises in (5.14). The background field is computed with:

Figure (5.7a): the first order upwind scheme,
Figure (5.7b): the fourth order time splitting scheme with Forester filter, and
Figure (5.7c): as in figure (5.7b) except for $n_p^0 = 500$.

The analytical solution is given in (5.12). Corresponding contour plots are shown in figure (5.6a). The experiments compared with the analytical results lead to:

- The solution obtained with the first order upwind scheme is smoothed by numerical diffusion - see figure (5.6b).

- The error in the particle simulation results is dominated by the number of particles inside the flow region (resolution problem) - see figure (5.6c).

- The solution obtained with the fourth order time splitting scheme gives adequate results. The error is dominant near the discharge location - see figure (5.6d).

- Particle/Euler simulations are less affected by numerical diffusion than the individual finite difference simulations - see figure (5.7a) in comparison with figure (5.6b). With respect to accurate finite difference schemes the overall error is only reduced if the number of particles is large enough to overcome the resolution problem - see figure (5.7b) and (5.7c). Especially in figure (5.7c) it is observed that the error near the discharge location is reduced in comparison with the finite difference model results sketched in figure (5.6d). In figure (5.8) the number of particles inside the flow region is given as a function of time. It is seen that the maximum number of particles active in the particle/Euler simulations converges to a certain limit.

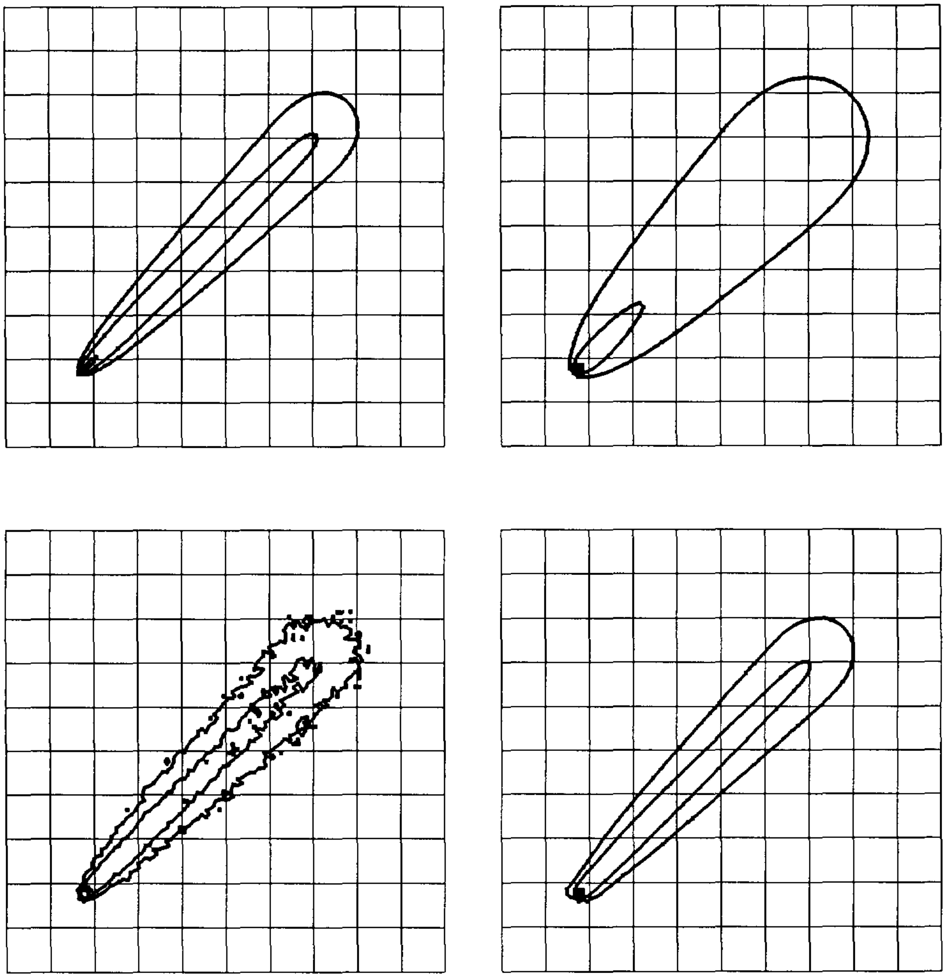


Figure (5.6): Iso-concentration contour lines of continuous discharge in two-dimensional unbounded domain. Each plotted grid cell contains 10 x 10 computational grid cells. Lines represent:

$C_i = C_{\max} \exp\left(-i^2/2\right)$ for i = ½, 1, 2, 3 with $C_{\max} = \max\limits_{x,y} C(x,y,T)$

5.6a (TOP-LEFT): analytical solution given in (5.12),

5.6b (TOP-RIGHT): first order upwind simulation,

5.6c (BOTTOM-LEFT): particle simulation, $n_p^0 = 100$,

5.6d (BOTTOM-RIGHT): fourth order time splitting simulation with Forester filter.

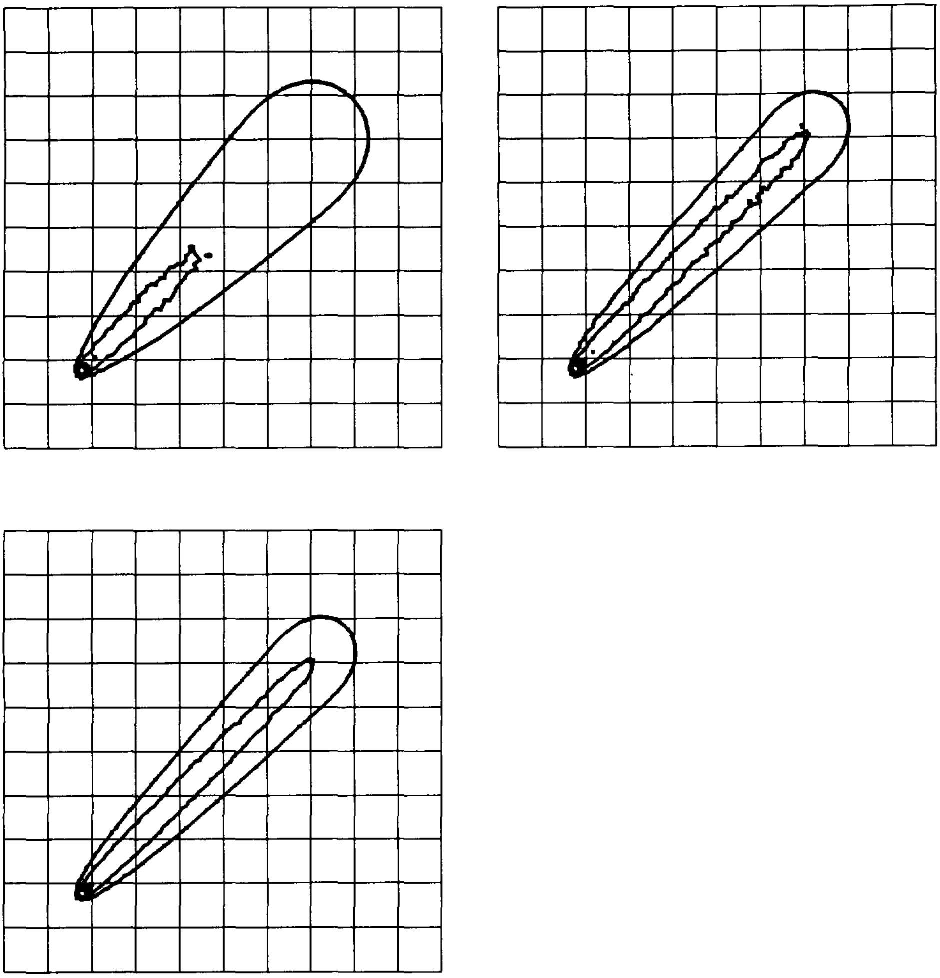


Figure (5.7): Iso-concentration contour lines of continuous discharge in two-dimensional unbounded domain. Each plotted grid cell contains 10 x 10 computational grid cells. Lines represent:

$C_i = C_{\max} \exp\left(-i^2/2\right)$ for $i = \frac{1}{2}, 1, 2, 3$ with $C_{\max} = \max\limits_{x,y} C(x,y,T)$

<u>Particle/Euler simulations</u> in which the background concentration is computed with

5.7a (TOP-LEFT): first order upwind, $n_p^0 = 100$,

5.7b (TOP-RIGHT): fourth order time splitting with Forester filter, $n_p^0 = 100$,

5.7c (BOTTOM-LEFT): fourth order time splitting with Forester filter, $n_p^0 = 500$.

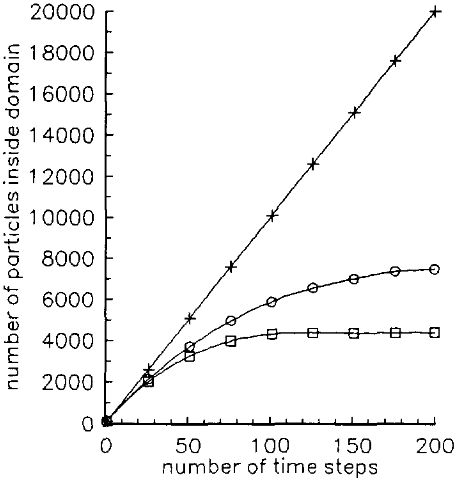


Figure (5.8): Number of particles inside computational domain as a function of time, $n_p^0 = 100$.

plus:	particle model,
circle:	particle/Euler model with fourth order time splitting and Forester filter,
box:	particle/Euler model with first order upwind.

It is concluded that the combined particle/Euler method is attractive in continuous discharge applications. It is recommended to apply an accurate numerical scheme for solving the background concentration field in a sense that numerical diffusion is hardly introduced for smooth distributions. A beneficial aspect of the method is that the number of computational particles is, in principle, restricted by the procedure instead of the computer capacity.

Chapter 6

Concluding remarks and discussion

6 Concluding remarks and discussion

The investigation employed in this thesis contribute to three-dimensional numerical modelling of transport phenomena in shallow water. The discussion is mainly restricted to advection-diffusion type transport models. For that purpose the numerical description of a hydrodynamic model that solves the water level elevation and the flow patterns is outlined together with alternative methods in solving the advection-diffusion equation. Special attention is paid to a numerical model based on particle tracking. This chapter summarizes the specific conclusions from this research project.

- The three-dimensional hydrodynamic shallow water flow solver

- The hydrodynamic model is formulated in terms of boundary-fitted coordinates. The basic motivation in doing so is based on numerical accuracy considerations. The sigma-transformation is applied to simplify problems associated with the free-water surface. However, the shallow water approximation (hydrostatic pressure assumption) leads to complex three-dimensional transformed equations, especially with respect to the viscous terms. It has been pointed out that the numerical procedure yields accurate results if the viscous terms are evaluated in the Cartesian space with a finite volume approach in which fluxes are calculated as strict horizontal fluxes.
- Nonlinear advection terms are studied according to STELLING (1984) in appendix D. Approximations of water level and velocity distributions depend on the chosen discretization scheme. If a conservative advection scheme is applied no artificial head loss is generated. These observations mainly serve qualitative requirements.
- More important than the discretization scheme of the advection terms is the applied turbulence model. Quantitative results are dominated by the used turbulence model.

- Three-dimensional transport models that solve the advection-diffusion equation

Three different types of models are compared. The first type is based on an Eulerian description of the transport process. The second type is based on a Lagrangian approach. Both alternatives are combined in the third approach.

(1) Finite difference simulation (Eulerian approach)

- In the Eulerian approach the advection-diffusion equation in transformed coordinates is discretized. As in the hydrodynamic model the transformation relations result in complex diffusive contributions. Accurate approximations are obtained if these terms are treated with respect to the original Cartesian coordinate system. If the transformed equations are directly discretized truncation errors may give rise to an artificial transport referred to as artificial creeping.

- The approximation of the Jacobian influences the accuracy of the approximation. It is deduced that best accurate results are obtained if the Jacobian is related to the exact grid cell volume.
- Finite difference approximations are useful in the simulation of far field concentration distributions. If steep concentration gradients are absent a more than second order accurate finite difference scheme has been described with minimal numerical diffusion.

(2) Particle tracking (Lagrangian approach)

- The particle tracking model is based on the similarity between the advection-diffusion equation and the Fokker-Planck equation. Many positions of particles are computed to determine a mass distribution which is associated with the concentration distribution. The hydrodynamic shallow water flow solver yields discrete flow information. This information is used in the computation of particle displacements. Since the hydrodynamic model is formulated in transformed coordinates, the particle model has also been developed in terms of such a coordinate system. By application of the Itô-convention it turns out that three-dimensional particle displacements are influenced by four contributions: (i) local flow velocity, (ii) space-varying diffusivity, (iii) curvature terms and (iv) stochastic forcing (eddy diffusivity). The curvature terms are of first order importance and their magnitude are determined by the diffusion coefficients.
- An algorithm is developed that accounts for various boundary conditions. The procedure estimates the mass flux through the boundary. Together with the prescribed boundary condition it determines the intensity of a source positioned at the boundary. The procedure yields consistent solutions in case of e.g., Robbin boundaries or Thatcher-Harleman boundary conditions. Special cases involve the zero-concentration boundary condition (particle absorbtion) and the zero-flux boundary condition (particle reflection). Perpendicular reflection implies $\partial C/\partial n = 0$. In anisotropic media a reflection principle that accounts for $D\nabla C \cdot n = 0$ is more appropriate.
- Three algorithms are studied to convert the mass density of discrete particles into concentrations. These algorithms are referred to as (i) numerical defined distribution (counting the frequency of observations), (ii) point spread function (summing the significant particle influences with Gaussian weight function), and (iii) maximum entropy distribution (minimizing the uncertainty given the first few moments of particle contributions). The first alternative depends on the grid cell dimensions. Meaningless results are obtained if the spreading in the frequency observations is either too small or too big. User independent results are obtained using point spread functions or the maximum entropy distribution. The latter is favourite in one-dimensional (unbounded)

applications. Especially the point spread function conversion algorithm deserves further research in applications with bounded geometry.

- Initialization of particles is realized by uniform injection. This procedure may cause specific spreading at initial state especially in coarse grids. In practical experiments the error induced by the applied injection procedure is negligible.

Numerical approximation of particle displacements

- An accurate advection algorithm is developed to compute the deterministic displacement of particles. This algorithm yields streamlines in a discretely defined flow field obtained with a conservative Eulerian method. Intersection of streamlines is impossible; closed streamlines are obtained in circulating flows. The efficiency of the algorithm is improved by considering higher order approximations of the velocity modulus along the streamline. Application is not only restricted to particle tracking models. It is also applicable in contour algorithms and in fractional step methods in which the advection algorithm accounts for the advective contribution.
- The approximated particle tracks are first order accurate in time. For that reason the stochastic contribution is realized with a uniform random number generator instead of a traditional Gaussian random number generator. Consequently, the maximum stochastic displacement is limited. This matter of probably minor importance benefits the numerical implementation.
- The stochastic displacement is preceded by the deterministic displacement.
- When dealing with a discrete Eulerian flow field, given with respect to a transformed grid, particle displacements can be computed in the transformed space. It is recommended to compute the deterministic displacement in the transformed space, while the stochastic contribution is calculated with respect to the original Cartesian coordinate system. Doing so, complex curvature terms do not have to be evaluated and reflection of particles at closed boundaries is easily established. In mildly curved grids the computation may be performed totally in the transformed domain, ignoring curvature terms. With mildly curved it is meant that the ratio (C_E) between the curvature terms and the remaining displacement contributions does not exceed the value 2. It is illustrated that omission of curvature terms in diffusion dominated applications with extremely yanked grids result in nonphysical concentration distributions.
- To account for decay contributions the mass evolution of a single particle is described with an exponential mass reduction law.

Experimental results

- Particle models are hardly affected by numerical diffusion.
- Qualitative results are efficiently obtained with relatively small number of particles.

Quantitative results require a rather large amount of particles. An increase of initial particles should be regarded as an increase of the number of process evaluations.

- Interpretation of particle simulations is sometimes complicated by observed scatter. The scatter is reduced by an increase of computational particles or the use of, for instance, point spread functions. Smoothness requirements are more easily realized by finite difference simulations.
- Comparison between finite difference approximations and particle approximations gives a good insight in the numerical diffusion present in the finite difference model.
- In general applications involving open boundaries particles may leave the computational domain. To maintain a sufficient resolution a reinitialization procedure is desired. The procedure that doubles the resolution at each reinitialization occurrence by splitting each particle in two shares gives adequate results.
- Roughly up to 10000 particles the computation time of particle models can compete with the computation time of finite difference simulations. However, particle models are suitable for parallel computations. Effective use of such parallel computers benefits competitiveness.

(3) Combined finite difference/particle simulations

- An effective transport solver is developed that distinguishes between far- and near field concentration distributions. Near field computations are executed with the particle model (foreground concentration). The far field concentration is resolved by an accurate finite difference scheme with minimal numerical diffusion (background concentration). The decision whether particles stimulate the background concentration is based on the steepness of the background concentration profile. Doing so, the accuracy of the method is dominated by the Courant number instead of the presence of a steep concentration gradient. Another advantage to be mentioned is that the number of computational particles converges to a certain limit. So, both accuracy and computational efficiency are realized.

- Discussion

- The hydrostatic pressure relation is mainly applied to serve numerical implementation. However, from a numerical point of view this condition may be weakened. For instance, advection terms arising in the vertical momentum equations do not necessarily have to be ignored. Although these terms might be physically redundant it is possible that transformed equations that include these terms may become easier to handle numerically.
- The applicability of the advection-diffusion equation is limited. Especially in discharge applications the concentration evolution is poorly described just after the moment of release. It is believed that so-called random flight models overcome this infirmity - see

HEEMINK (1990). For these models the numerical considerations of the described particle model are immediately applicable. A combined finite difference/random flight model has favourable characteristics in prospect.

- Extension to suspended sediment transport in shallow water environment requires advanced physical knowledge about sediment behaviour near boundaries. Prescribed boundary conditions are resolved by finite difference models as well as particle models. Although boundary conditions are more easy to handle with finite differences the particle model appears to be an alternative. The stochastic approach in boundary treatment may yield more flexible physical descriptions that are worth to examine.

References

References

Berkowicz, R. and L.P. Prahm, (1980)
On the spectral turbulent diffusivity theory for homogeneous turbulence.
Journal of Fluid Mechanics, Vol. 100, Part 2, pp433-448.

Blokland, P.A. and A.W. Heemink, (1993)
A random flight model in general coordinates.
Presented at the international conference on stochastic and statistical methods in hydrology and environmental engineering, june 21-23, 1993, Waterloo, Canada, proceedings in preparation.

Boogaard, H.F.P. van den, M.J.J. Hoogkamer and A.W. Heemink, (1993)
Parameter identification in particle models.
Stochastic Hydrology and Hydraulics, Vol. 7, No. 2, pp109-130.

Booij, R., (1991)
Eddy viscosity in a harbour.
Proceedings of the XXIVth IAHR Congress Madrid, ppC081-C090.

Booij, R., (1994)
Preliminary report on measurements in a scale model of rectangular harbours along a river.
Department of Civil Engineering, Delft University of Technology, in preparation.

Boris, J.P. and D.L. Book, (1976)
Flux-corrected transport. II.
Journal of Computational Physics. Vol. 20, pp397-431.

Buning, P.G., (1989)
Numerical algorithms in CFD post-processing.
In: Computer graphics and flow visualization in computational fluid dynamics.
Lecture series Von Karman Institute for Fluid Dynamics 1989-07.
Institution von Karman de Dynamique des fluides, Rhode Saint Genese, 1989.

Chandrasekhar, S., (1943)
Stochastic problems in physics and astronomy.
Reviews of Modern Physics, Vol. 15, No. 1, pp1-89.

Cole, P. and G.V. Miles, (1983)
Two-dimensional model of mud transport.
Journal of Hydraulic Engineering, Vol. 109, No. 1, pp1-12.

Corrsin, S., (1974)
Limitations of gradient transport models in random walks and in turbulence.
Advances in Geophysics, Vol. 18A, pp25-60.

Csanady, G.T., (1973)
Turbulent diffusion in the environment.
Reidel, Dordrecht.

Cuvelier, C., (1987)
Differential equations of viscous fluid flow in general coordinates.
Department of Mathematics, Delft University of Technology, lecture notes.

Dashevsky, M.L., (1975)
A semiinvariant method of closing the equations for moments in statistical analysis of nonlinear systems.
Problems of Control and Information Theory, Vol. 4, No. 1, pp317-326.

Derksen, R.W. and P.J. Sullivan, (1990)
Moment approximations for probability density functions.
Combustion and Flame, Vol. 81, pp378-391.

Dimou, K.N. and E.E. Adams, (1989)
2-D particle tracking model for estuary mixing.
In: Estuarine and Coastal Modeling; proceedings of the conference, Newport, Rhode Island, november 15-17, 1989, ed.: M.L. Spaulding, pp472-481.
ASCE, New York, 1990.

Dowson, D.C. and A. Wragg, (1973)
Maximum-entropy distributions having prescribed first and second moments.
IEEE Transactions on Information Theory, Vol. IT-19, pp689-693.

Durbin, P.A., (1983)
Stochastic differential equations and turbulent dispersion.
NASA Reference Publication, No. 1103.

Dyer, K.R., (1977)
Estuaries: a physical introduction.
Wiley, London.

Dynkin, E.B. and A.A. Juschkewitsch, (1969)
Sätze und aufgaben über Markoffsche prozesse.
Springer, Berlin.

Engquist, B., P. Lötstedt and B. Sjögreen, (1989)
Nonlinear filters for efficient shock computation.
Mathematics of Computation, Vol. 52, No. 186, pp509-537.

Feller, W., (1971)
An introduction to probability theory and its applications; Volume 2, second edition.
Wiley, New York.

Fischer, H.B., E.J. List, R.C.Y. Koh, J. Imberger, and N.H. Brooks, (1979)
Mixing in inland and coastal waters.
Academic Press, San Diego.

Fischer, H.B., (1981)

Transport models for inland and coastal waters;

symposium on predictive ability of surface water flow and transport models, Berkeley, august 1980, proceedings, ed.: H.B. Fischer, Academic Press, New York, 1981.

Fishman, G.S. and L.R. Moore, (1986)

An Exhaustive analysis of multiplicative congruential random number generators with modulus 2**31 - 1.

SIAM Journal on Scientific and Statistical Computing, Vol. 7, No. 1, pp24-45.

Fletcher, C.A.J., (1988)

Computational techniques for fluid dynamics 2; specific techniques for different flow categories.

Springer, Berlin.

Fliert, B. van de, (1988)

Stochastische simulatiemethoden; voor de beschrijving van diffusieverschijnselen in inhomogene turbulente stromen.

M-Sc. thesis, Twente University of Technology, Applied Mathematics Department (in dutch).

Forester, C.K., (1977)

Higher order monotonic convective difference schemes.

Journal of Computational Physics, Vol. 23, pp1-22.

Friedman, A., (1975)

Stochastic differential equations and applications; Volume 1.

Academic Press, New York.

Ghoniem, A.F. and F.S. Sherman, (1985)

Grid-free simulation of diffusion using random walk methods.

Journal of Computational Physics, Vol. 61, No. 1, pp1-37.

Godunov, S.K. and V.S. Ryabenki, (1964)

Theory of difference schemes.

North-Holland Publishing Company, Amsterdam.

Greiner, A., W. Strittmatter and J. Honerkamp, (1988)

Numerical integration of stochastic differential equations.

Journal of Statistical Physics, Vol. 51, No. 1/2, pp95-108.

Haworth, D.C. and S.B. Pope, (1986)

A Second-order Monte Carlo method for the solution of the Ito stochastic differential equation.

Stochastic Analysis and Applications, Vol. 4, No. 2, pp151-186.

Hayter, E.J. and A.J. Mehta, (1986)

Modelling cohesive sediment transport in estuarial waters.

Applied Mathematical Modelling, Vol. 10, pp294-303.

Heemink, A.W., (1990)
Stochastic modelling of dispersion in shallow water.
Stochastic Hydrology and Hydraulics, Vol. 4, pp161-174.
Jaynes, E.T., (1957)
Information theory and statistical mechanics.
Physical Review, Vol. 106, No. 4, pp620-630.
Jazwinski, A.H., (1970)
Stochastic processes and filtering theory.
Academic Press, New York.
Jin, X.-Y., (1993)
Quasi-three-dimensional numerical modelling of flow and dispersion in shallow water.
Ph.D. Thesis, Delft, 1993.
Jong, K. de and A.W. Heemink, (1993)
A tidally avaraged numerical transport model.
MaST*G8M, Report No. DGW-93.046.
Kampen, N.G. van., (1981)
Ito versus Stratonovich.
Journal of Statistical Physics, Vol. 24, No. 1, pp175-187.
Karlin, S. and H.M. Taylor, (1975)
A first course in stochastic processes; second edition.
Academic Press, New York.
Kester, J.A.Th.M. van and R.E. Uittenbogaard, (1990)
The σ-coordinate transformation and the basic equations of TRISULA.
Delft Hydraulics, Z81-21.
Kloeden, P.E. and E. Platen, (1992)
Numerical solution of stochastic differential equations.
Springer, Berlin.
Kranck, K., (1984)
Settling behavior of cohesive sediment.
In: Estuarine cohesive sediment dynamics; workshop on cohesive sediment dynamics with special reference to physical processes in estuaries, Tampa, November 1984, proceedings, ed.: A.J. Mehta, Lecture notes on coastal and estuarine studies, Vol. 14, pp151-169.
Springer, Berlin, 1986.
Lambert, J.D., (1973)
Computational methods in ordinary differential equations.
Wiley, Chichester.
Langendoen, E.J., (1992)
Flow patterns and transport of dissolved matter in tidal harbours.
Ph.D. Thesis, Delft, 1992.

Launder, B.E., (1991)

Introduction to the modelling of turbulence.

In: Introduction to the modeling of turbulence.

Lecture series Von Karman Institute for Fluid Dynamics 1991-02.

Institution von Karman de Dynamique des fluides, Rhode Saint Genese, 1991.

Leendertse, J.J., (1989)

A new approach to three-dimensional free-surface flow modeling.

RAND-Corporation, Santa Monica, RM-5294-PR.

Lesieur, M., (1990)

Turbulence in fluids; stochastic and numerical modelling; second revised edition.

Kluwer, Dordrecht.

Lewis, P.A.W. and E.J. Orav, (1989)

Simulation methodology for statisticians, operations analysts and engineers; Volume 1.

Wadworth and Brooks/Cole, Pacific Grove.

Mallinson, G.D. and G. de Vahl Davis, (1977)

Three-dimensional natural convection in a box: a numerical study.

Journal of Fluid Mechanics, Vol. 83, Part 1, pp1-31.

Mil'shtein, G.N., (1974)

Approximate integration of stochastic differential equations.

Theory of Probability and its Applications, Vol. 19, pp557-562.

Milshtein, G.N., (1978)

A method of second-order accuracy integration of stochastic differential equations.

Theory of Probability and its Applications, Vol. 23, pp396-401.

Mitchell, A.R. and D.F. Griffiths, (1980)

The finite difference method in partial differential equations.

Wiley, Chichester.

Molenkamp, C.R., (1967)

Accuracy of finite-difference methods applied to the advection equation.

Journal of Applied Meteorology, Vol. 7, pp160-167.

Monin, A.S. and A.M. Yaglom, (1975)

Statistical fluid mechanics: mechanics of turbulence; Volume 2.

M.I.T. Press, Cambridge.

Montminy, M., M. Leclerc, G. Martin, and P. Boudreault, (1992)

PANACHE: An interactive software to simulate steady-state two-dimensional transport-diffusion of pollutants in rivers using a particle tracking method.

In: Hydraulic engineering software IV; computer techniques and applications.

Fourth international conference on hydraulic engineering software; hydrosoft '92, Valencia, july 1992, eds.: W.R. Blain and E. Cabrera, pp27-38.

Elsevier Applied Science, London, 1992.

O'Connor, B.A. and J. Nicholson, (1988)

Mud transport modelling.

In: Physical processes in estuaries, eds.: J. Dronkers and W. van Leussen, pp532-544.

Springer, Berlin, 1988.

Øksendal, B., (1992)

Stochastic differential equations; an introduction with applications; third edition.

Springer, Berlin.

Oliger, J. and A. Sundström, (1978)

Theoretical and practical aspects of some initial boundary value problems in fluid dynamics.

SIAM Journal of Applied Mathematics, Vol. 35, pp419-446.

Onishi, Y., (1981)

Sediment-contaminant transport model.

Journal of the Hydraulics Division, ASCE, Vol. 107, No. HY9, pp1089-1107.

Palleschi, V. and M. de Rosa, (1992)

Numerical solution of the Fokker-Planck equation; II. Multidimensional case.

Physical Letters A, Vol. 163, pp381-391.

Paola, M. di and G. Falsone, (1993)

Stochastic dynamics of nonlinear systems driven by non-normal delta-correlated processes.

Journal of Applied Mechanics, Vol. 60, pp141-148.

Papoulis, A., (1965)

Probability, random variables and stochastic processes.

McGraw-Hill, Tokyo.

Pawula, R.F., (1967)

Approximation of the linear Boltzmann equation by the Fokker-Planck Equation.

Physical Review, Vol. 162, No. 1, pp186-188.

Phillips, N.G., (1957)

A coordinate system having some special advantages for numerical forecasting.

Journal of Meteorology, Vol. 14, pp184-185.

Pourquié, M., (1990)

Numerical comparison of advection schemes to be used on a coarse grid.

Laboratory for Aero- and Hydrodynamics, Delft University of Technology, MEAH-94.

Prandtl, Z.A., (1925)

Bericht über untersuchungen zur ausgebildeten turbulenz.

Zeitschrift für Angewandte Mathematik und Mechanik, Vol. 5, pp136-169.

Pugachev, V.S. and I.N. Sinitsyn, (1987)

Stochastic differential systems; analysis and filtering.

Wiley, Chichester.

Rao, N.J., J.D. Borwankar and D. Ramkrishna., (1974)
Numerical solution of Ito integral equations.
SIAM Journal on Control, Vol. 12, No. 1, pp124-139.
Richtmeyer, R.D. and K.W. Morton, (1967)
Difference methods for initial-value problems; second edition.
Wiley, New York.
Rijn, L.C. van, (1989)
The state of the art in sediment transport modelling.
In: Sediment transport modeling; proceedings of the international symposium, Hotel Inter-Continental, New Orleans, Louisiana, august 14-18, 1989, ed.: S.S.Y. Wang, pp13-32.
ASCE, New York, 1989.
Ripley, B.D., (1987)
Stochastic simulation.
Wiley, New York.
Risken, H., (1984)
The Fokker-Planck equation.
Springer, Berlin.
Rodi, W., (1980)
Turbulence models in environmental problems.
In: Prediction methods for turbulent flows; lecture series, Rhode-St. Genese, Von Karman Institute for Fluid Dynamics, january 1979, proceedings, ed.: W. Kollmann, pp259-349.
Hemisphere, Washington, 1980.
Rodi, W., (1984)
Turbulence models and their applications in hydraulics; state of the art paper.
IAHR, Delft.
Rohlf, F.J. and R.R. Sokal, (1981)
Critical values of the chi-square distribution.
In: Statistical tables; second edition.
Freeman, San Francisco, 1981.
Rubinstein, R.Y., (1981)
Simulation and the Monte Carlo method.
Wiley, New York.
Rümelin, W., (1982)
Numerical treatment of stochastic differential equations.
SIAM Journal on Numerical Analysis, Vol. 19, No. 3, pp604-613.
Sabelfeld, K.K., (1991)
Monte Carlo methods in boundary value problems.
Springer, Berlin.

Sadarjoen, A., T. van Walsum, A.J.S. Hin and F.H. Post, (1994)
Particle tracing algorithms for 3D curvilinear grids.
Presented at the 5th Eurographics workshop on visualization in scientific computing, may 30 - june 1, 1994, Rostock, Germany, in preparation.

Schuss, Z., (1980)
Theory and applications of stochastic differential equations.
Wiley, New York.

Shore, J.E. and R.W. Johnson, (1980)
Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy.
IEEE Transactions on Information Theory, Vol. IT-26, No. 1, pp26-37.

Siddall, J.N., (1983)
Propabilistic engineering design; principles and applications.
Dekker, New York.

Skilling, J., (1988)
The axioms of maximum entropy.
In: Maximum-entropy and Bayesian methods in science and engineering; Volume 1: foundations, eds.: G.J. Erickson and C.R. Smith, pp173-187.
Kluwer, Dordrecht, 1988.

Stelling, G.S., (1984)
On the construction of computational methods for shallow water flow problems.
Rijkswaterstaat Communications, No. 35, The Hague.

Stelling, G.S. and J.A.Th.M. van Kester (1994)
On the approximation of horizontal gradients in sigma co-ordinates for bathymetry with steep bottom slopes.
International Journal for Numerical Methods in Fluids, Vol. 18, pp915-935.

Taira, K., (1988)
Diffusion processes and partial differential equations.
Academic Press, Boston.

Thatcher, M.L. and R.F. Harleman, (1972)
A Mathematical model for the prediction of unsteady salinity intrusion in estuaries.
Massachusetts Institute of Technology, Report No. 144,

Thompson, J.F., Z.U.A. Warsi and C.W. Mastin, (1985)
Numerical grid generation: foundations and applications.
North-Holland Publishing Company, Amsterdam.

Thomson, D.J., (1987)
Criteria for the selection of stochastic models of particle trajectories in turbulent flows.
Journal of Fluid Mechanics, Vol. 180, pp529-556.

TRISULA, (1988)
TRISULA, Volume III: The scientific background.
Delft Hydraulics, Z-215-05.

Truesdell, C., (1954)
The kinematics of vorticity.
Indiana University Press, Bloomington.

Uffink, G.J.M., (1990)
Analysis of dispersion by the random walk method.
Ph.D. Thesis, Delft, 1990.

Verboom, G.K., G.S. Stelling and M.J. Officier, (1981)
Non-reflective boundary conditions in horizontal flow models.
In: Numerical methods for coupled problems; international conference, Swansea, september 1981, proceedings, eds. E. Hinton *et al.*, pp71-81.
Pineridge, Swansea, 1981.

Verboom, G.K., G.S. Stelling and M.J. Officier, (1982)
Boundary conditions for the shallow water equations.
In: Engineering applications for computational hydraulics; homage to Alexandre Preissmann; Volume 1, eds.: M.B. Abbott and J.A. Cunge, pp230-262.
Pitman, Boston, 1982.

Vreugdenhil, C.B., (1989)
Computational hydraulics; an introduction.
Springer, Berlin.

Vreugdenhil, C.B. and B. Koren, (1993)
Numerical methods for advection-diffusion problems.
Vieweg, Braunschweig.

Wang, Z.B., (1989)
Mathematical modelling of morphological processes in estuaries.
Ph.D. Thesis, Delft, 1989.

Yanenko, N.N., (1971)
Method of fractional steps.
Springer, Berlin.

Appendices

Appendix A: Transformation relations (I)

Curvilinear boundary fitted coordinates are frequently used to solve fluid flow problems. This transformation approach is introduced from a numerical point of view. Although the transformation from a Cartesian coordinate system (x_1,x_2,x_3) to such a curvilinear coordinate system (ξ^1,ξ^2,ξ^3) results in more complex differential equations it is possible to derive convenient and accurate discretization methods in the transformed space. Here, Cartesian grids are defined as grids with straight lines and unit cubes as cells.

In this appendix definitions and necessary transformation relations to obtain transformed equations are summarized. General comments about and a rigorous derivation of the transformation relations can be found in THOMPSON *et al.* (1985) or CUVELIER (1987).

- j^{th} component of the i^{th} covariant base vector $\left(\underline{a}_i\right)_j = \dfrac{\partial x_i}{\partial \xi^j}$ (A.1)
- j^{th} component of the i^{th} contravariant base vector $\left(\underline{a}^i\right)_j = \dfrac{\partial \xi^i}{\partial x_j}$ (A.2)
- Covariant metric tensor $G_{ij} = \underline{a}_i \cdot \underline{a}_j = G_{ji}$ (A.3)
- Jacobian of the transformation $J = \sqrt{\mathrm{Det}\{G_{ij}\}}$ (A.4)
- Christoffel symbol of the second kind $\Gamma_{ij}^k = \sum_{l=1}^{3} \dfrac{\partial^2 x_l}{\partial \xi^i \partial \xi^j}\left(\underline{a}^k\right)_l$ (A.5)
- Contravariant velocity component $U^i = \underline{a}^i \cdot \left(\underline{u} - \left(\dfrac{\partial \underline{x}}{\partial t}\right)_{\underline{\xi}}\right)$ (A.6)
- Cartesian velocity vector $\underline{u} - \left(\dfrac{\partial \underline{x}}{\partial t}\right)_{\underline{\xi}} = \sum_{i=1}^{3} U^i \underline{a}_i$ (A.7)

The subscript in (A.6) and (A.7) indicates the variable which is held constant. The Jacobian J is used to express that volume elements are being transformed according this Jacobian.

In numerical simulations the transformation defines a computational mesh. The definitions given above are geometrically interpretted in figure (A.1).

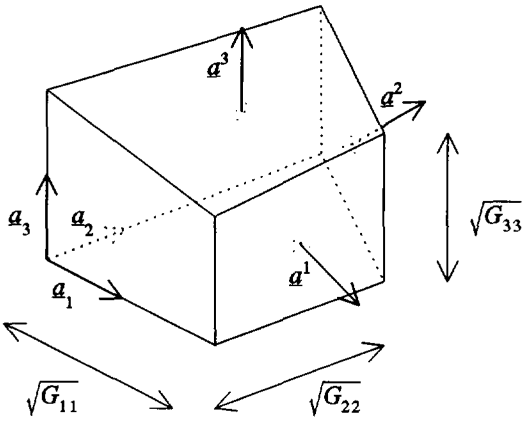


Figure (A.1): Geometric illustration of transformation definitions in hexahedral cell.

- $\sqrt{G_{ii}}$ represents the local grid size of curvilinear mesh.
- Jacobian J is related to the local grid cell volume.
- Nonorthogonality is quantified by G_{ij} $i \neq j$.
- Curvature terms are given by Γ^k_{ij}.
- Velocity vector $\underline{u}$ is written with respect to (nonmoving) covariant basis: $\underline{u} = \sum\limits_{i=1}^{3} U^i \underline{a}_i$.

The chain rule provides for a scalar quantity such as concentration

$$\frac{\partial C}{\partial x_j} = \sum_{i=1}^{3} \frac{\partial C}{\partial \xi^i} \frac{\partial \xi^i}{\partial x_j} = \sum_{i=1}^{3} \left(\underline{a}^i\right)_j \frac{\partial C}{\partial \xi^i} = \frac{1}{J} \sum_{i=1}^{3} \frac{\partial}{\partial \xi^i} \left[J \left(\underline{a}^i\right)_j C \right] \tag{A.8}$$

To obtain a conservative description, expressed by the third equality of (A.8), the following identity has been used

$$\sum_{i=1}^{3} \frac{\partial}{\partial \xi^i} \left[J \underline{a}^i \right] = \underline{0} \tag{A.9}$$

This identity can be proved with the divergence theorem applied to an infinitesimal element and reminding that the vector normal to the faces of this infinitesimal element is given by the contravariant base vector.

If a moving grid is applied the time derivatives also have to be transformed. The expression for the time derivative becomes

$$\left(\frac{\partial C}{\partial t}\right)_{\underline{\xi}} = \left(\frac{\partial C}{\partial t}\right)_{\underline{x}} + \nabla C \cdot \left(\frac{\partial \underline{x}}{\partial t}\right)_{\underline{\xi}} \tag{A.10}$$

where the subscripts indicate the variable which is held constant.

- Special relations used in chapter 2, 3 and 4

Gradient

$$\nabla C = \frac{1}{J}\sum_{i=1}^{3} \frac{\partial}{\partial \xi^i}\left[J\underline{a}^i C\right] \tag{A.11}$$

Second derivative

$$\frac{\partial^2 C}{\partial x_i \partial x_j} = \frac{1}{J}\sum_{k=1}^{3}\sum_{l=1}^{3} \frac{\partial}{\partial \xi^k}\left[\frac{\partial \xi^k}{\partial x_j}\frac{\partial}{\partial \xi^l}\left(\frac{\partial \xi^l}{\partial x_i} JC\right)\right] \tag{A.12}$$

Divergence

$$\begin{aligned}\nabla \cdot \underline{u} &= \frac{1}{J}\sum_{i=1}^{3} \frac{\partial}{\partial \xi^i}\left[J\underline{a}^i \cdot \underline{u}\right]\\ &= \frac{1}{J}\left(\frac{\partial J}{\partial t} + \sum_{i=1}^{3} \frac{\partial}{\partial \xi^i}\left[JU^i\right]\right)\end{aligned} \tag{A.13}$$

Finally two relations are given related to derivatives of transformed coordinates

$$\left(\frac{\partial \xi^i}{\partial t}\right)_{\underline{x}} = -\sum_{j=1}^{3} \frac{\partial \xi^i}{\partial x_j}\left(\frac{\partial x_j}{\partial t}\right)_{\underline{\xi}} \tag{A.14}$$

$$\frac{\partial^2 \xi^i}{\partial x_j \partial x_k} = \sum_{l=1}^{3} \frac{\partial \xi^l}{\partial x_j}\frac{\partial}{\partial \xi^l}\left(\frac{\partial \xi^i}{\partial x^k}\right) \tag{A.15}$$

Appendix B: Transformation relations (II);

a curvilinear transformation in the horizontal plane and the sigma transformation in the vertical plane

This appendix summarizes the transformation relations relative to a curvilinear transformation in the horizontal plane and the sigma transformation in the vertical plane. Some comments are made about the choice of specific transformed velocity components. The transformation relations are deduced from appendix A. Here the Cartesian coordinate system is denoted with (x,y,z), the curvilinear coordinate system with (ξ,η,σ).

The curvilinear (boundary fitted) transformation in the horizontal plane is written in general form as

$$\begin{cases} \xi = \xi(x,y) \\ \eta = \eta(x,y) \end{cases} \quad \text{or equivalently} \quad \begin{cases} x = x(\xi,\eta) \\ y = y(\xi,\eta) \end{cases} \tag{B.1}$$

The sigma transformation is defined as in PHILLIPS (1957). Originally it was used in atmospheric numerical models, but it can also be applied in free surface water flows. The sigma coordinate is defined as

$$\sigma = \frac{z - \zeta(x,y,t)}{H(x,y,t)} \quad \text{or} \quad z = \sigma H(x(\xi,\eta),y(\xi,\eta),t) + \zeta(x(\xi,\eta),y(\xi,\eta),t) \tag{B.2}$$

with H the water depth and ζ the water level elevation above the plane of reference $z = 0$. With respect to this transformation, (B1) and (B2), the covariant base vectors, the tangent vectors to the three coordinate axes, become

$$\begin{cases} \underline{a}_1 = \left(\dfrac{\partial x}{\partial \xi}, \dfrac{\partial y}{\partial \xi}, \dfrac{\partial z}{\partial \xi}\right)^T \\ \underline{a}_2 = \left(\dfrac{\partial x}{\partial \eta}, \dfrac{\partial y}{\partial \eta}, \dfrac{\partial z}{\partial \eta}\right)^T \\ \underline{a}_3 = (0\ ,\ 0\ ,\ H(x(\xi,\eta),y(\xi,\eta),t))^T \end{cases} \tag{B.3}$$

the superscript T indicates the transpose of the vector. The Jacobian of the overall transformation is expressed by

$$J = H\sqrt{G} \tag{B.4}$$

with $\sqrt{G}$ the Jacobian of the horizontal transformation (B.1)

$$\sqrt{G} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \tag{B.5}$$

The corresponding contravariant base vectors, the normal vectors to the three coordinate surfaces, become

$$\begin{cases} \underline{a}^1 = \dfrac{1}{J}\left(\underline{a}_2 \times \underline{a}_3\right) = \dfrac{1}{\sqrt{G}}\left(\dfrac{\partial y}{\partial \eta}, -\dfrac{\partial x}{\partial \eta}, 0\right)^T \\[2ex] \underline{a}^2 = \dfrac{1}{J}\left(\underline{a}_3 \times \underline{a}_1\right) = \dfrac{1}{\sqrt{G}}\left(-\dfrac{\partial y}{\partial \xi}, \dfrac{\partial x}{\partial \xi}, 0\right)^T \\[2ex] \underline{a}^3 = \dfrac{1}{J}\left(\underline{a}_1 \times \underline{a}_2\right) = \dfrac{1}{H\sqrt{G}}\begin{pmatrix} \dfrac{\partial y}{\partial \xi}\dfrac{\partial z}{\partial \eta} - \dfrac{\partial y}{\partial \eta}\dfrac{\partial z}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta}\dfrac{\partial z}{\partial \xi} - \dfrac{\partial x}{\partial \xi}\dfrac{\partial z}{\partial \eta} \\ \sqrt{G} \end{pmatrix} \end{cases} \tag{B.6}$$

From (B.6) and (A.2) it is deduced that

$$\begin{cases} \dfrac{\partial \xi}{\partial x} = \dfrac{1}{\sqrt{G}}\dfrac{\partial y}{\partial \eta} \\[2ex] \dfrac{\partial \eta}{\partial x} = -\dfrac{1}{\sqrt{G}}\dfrac{\partial y}{\partial \xi} \end{cases} \qquad \begin{cases} \dfrac{\partial \xi}{\partial y} = -\dfrac{1}{\sqrt{G}}\dfrac{\partial x}{\partial \eta} \\[2ex] \dfrac{\partial \eta}{\partial y} = \dfrac{1}{\sqrt{G}}\dfrac{\partial x}{\partial \xi} \end{cases} \tag{B.7}$$

In numerical simulations fluxes through grid cell sides are computed. For that purpose velocity components are required that are measured normal to these grid cell sides. On the other hand contravariant components are introduced. These components are related to vectors not necessarily normal to the grid cell sides. To join both ideas the contravariant components will be subjected to a particular metric. First the horizontal transformation will be discussed. The metric tensor that corresponds with the horizontal transformation (B.1) will be denoted with

$$\begin{pmatrix} G_{\xi\xi} & G_{\xi\eta} \\ G_{\xi\eta} & G_{\eta\eta} \end{pmatrix} = \begin{pmatrix} \left(\dfrac{\partial x}{\partial \xi}\right)^2 + \left(\dfrac{\partial y}{\partial \xi}\right)^2 & \dfrac{\partial x}{\partial \xi}\dfrac{\partial x}{\partial \eta} + \dfrac{\partial y}{\partial \xi}\dfrac{\partial y}{\partial \eta} \\ \dfrac{\partial x}{\partial \xi}\dfrac{\partial x}{\partial \eta} + \dfrac{\partial y}{\partial \xi}\dfrac{\partial y}{\partial \eta} & \left(\dfrac{\partial x}{\partial \eta}\right)^2 + \left(\dfrac{\partial y}{\partial \eta}\right)^2 \end{pmatrix} \tag{B.8}$$

In figure (B.1) a two-dimensional illustration is given. The vector $\vec{AB}$ points in $\underline{a}_1$ - direction. Its magnitude equals $U \, \|\underline{a}_1\|$. The vector $\vec{AC}$ corresponds with the component of $\vec{AB}$ along the $\underline{a}^1$ - axis. $\vec{AC}$ is normal to the grid cell side. So the magnitude of $\vec{AC}$ defines a velocity component that is suitable to compute fluxes through grid cell sides. The following relation is valid

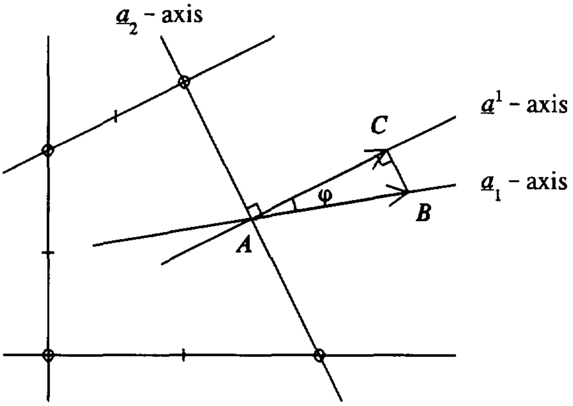


Figure (B.1): Velocity components at grid cell side in horizontal plane.

$|\vec{AB}| = U\,\|\underline{a}_1\|$, $|\vec{AC}| = u_\xi$.

$$\|\vec{AC}\| = \|\vec{AB}\|\cos\varphi = U\,\|\underline{a}_1\|\frac{\left(\underline{a}_1\cdot\underline{a}^1\right)}{\|\underline{a}_1\|\,\|\underline{a}^1\|} = \frac{U}{\|\underline{a}^1\|} \tag{B.9}$$

$\|\vec{AC}\|$ defines the magnitude of a new velocity component that is used in numerical simulations. This magnitude is related to the components of the metric tensor given in (B.8). In a three-dimensional case the metric tensor G_{ij}, defined in (A.3), is considered. An increment $d\xi$ corresponds with an increment of arc length $\|\underline{a}_1\|d\xi = \sqrt{G_{11}}d\xi$ in the physical space, while moving along the $\underline{a}_1$ - axis. New velocity components are usually defined by $U\,/\,\|\underline{a}^1\|$, $V\,/\,\|\underline{a}^2\|$ and $W\,/\,\|\underline{a}^3\|$. These components are suitable to compute (normal) fluxes through grid cell sides. However the use of the sigma transformation yields other velocity components. To compute fluxes through vertical grid cell sides still $U\,/\,\|\underline{a}^1\|$ and $V\,/\,\|\underline{a}^2\|$ are defined. The third component is chosen in vertical direction. It simplifies the prescription of the boundary conditions at free water surface and bottom. This third component is defined as a vertical velocity component relative to constant sigma planes - see figure (2.4) of section 2.4. By definition the contravariant components are, inserting (B.6) into (A.6),

$$\begin{cases} U = \dfrac{1}{\sqrt{G}}\left(\dfrac{\partial y}{\partial\eta}u - \dfrac{\partial x}{\partial\eta}v\right) \\ V = \dfrac{1}{\sqrt{G}}\left(-\dfrac{\partial y}{\partial\xi}u + \dfrac{\partial x}{\partial\xi}v\right) \\ W = \dfrac{1}{H}\left(w - \left(\dfrac{\partial z}{\partial t}\right)_{(\xi,\eta,\sigma)} - U\dfrac{\partial z}{\partial\xi} - V\dfrac{\partial z}{\partial\eta}\right) \end{cases} \tag{B.10}$$

The following definition of velocity components is applied

$$\left\{\begin{array}{lllllll} u_\xi & = & \dfrac{1}{\|\underline{a}^1\|}U & = & \dfrac{\sqrt{G}}{\sqrt{G_{\eta\eta}}}\ U & = & \dfrac{1}{\sqrt{G_{\eta\eta}}}\left(\dfrac{\partial y}{\partial \eta}u - \dfrac{\partial x}{\partial \eta}v\right) \\ u_\eta & = & \dfrac{1}{\|\underline{a}^2\|}V & = & \dfrac{\sqrt{G}}{\sqrt{G_{\xi\xi}}}\ V & = & \dfrac{1}{\sqrt{G_{\xi\xi}}}\left(-\dfrac{\partial y}{\partial \xi}u + \dfrac{\partial x}{\partial \xi}v\right) \\ \omega & = & \|\underline{a}_3\|\, W & = & H\ W & = & w - \dfrac{1}{\sqrt{G}}\left(u_\xi\sqrt{G_{\eta\eta}}\dfrac{\partial z}{\partial \xi} + u_\eta\sqrt{G_{\xi\xi}}\dfrac{\partial z}{\partial \eta}\right) - \left(\dfrac{\partial z}{\partial t}\right)_{(\xi,\eta,\sigma)} \end{array}\right. \tag{B.11}$$

The subscript (ξ,η,σ) represents the variable which is held constant. To obtain transformed partial differential equations the following relation, derived from (B.11), is extensively used

$$\left\{\begin{array}{lll} u & = & \dfrac{1}{\sqrt{G}}\left(\dfrac{\partial x}{\partial \xi}\sqrt{G_{\eta\eta}}u_\xi + \dfrac{\partial x}{\partial \eta}\sqrt{G_{\xi\xi}}u_\eta\right) \\ v & = & \dfrac{1}{\sqrt{G}}\left(\dfrac{\partial y}{\partial \xi}\sqrt{G_{\eta\eta}}u_\xi + \dfrac{\partial y}{\partial \eta}\sqrt{G_{\xi\xi}}u_\eta\right) \end{array}\right. \tag{B.12}$$

Relation (A.7) provides

$$u\underline{e}_1 + v\underline{e}_2 + \left(w - \left(\frac{\partial z}{\partial t}\right)_{(\xi,\eta,\sigma)}\right)\underline{e}_3 = \left(\|\underline{a}^1\|u_\xi\right)\underline{a}_1 + \left(\|\underline{a}^2\|u_\eta\right)\underline{a}_2 + \left(\frac{\omega}{\|\underline{a}_3\|}\right)\underline{a}_3 \tag{B.13}$$

u_ξ and u_η define nearly horizontal velocity components with dimension m/s. These components are measured in horizontal direction. The ω - velocity component is strictly vertical. The boundary condition for ω at the free water surface becomes

$$\begin{aligned} \omega|_{\sigma=0} &= w|_{z=\zeta} - \frac{1}{\sqrt{G}}\left(u_\xi\sqrt{G_{\eta\eta}}\frac{\partial \zeta}{\partial \xi} + u_\eta\sqrt{G_{\xi\xi}}\frac{\partial \zeta}{\partial \eta}\right) - \frac{\partial \zeta}{\partial t} \\ &= w|_{z=\zeta} - \left(\frac{\partial \zeta}{\partial t} + u_\xi\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}\frac{\partial \zeta}{\partial \xi} + u_\eta\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}\frac{\partial \zeta}{\partial \eta}\right) \\ &= w|_{z=\zeta} - \left(\frac{\partial \zeta}{\partial t} + u\frac{\partial \zeta}{\partial x} + v\frac{\partial \zeta}{\partial y}\right) \end{aligned} \tag{B.14}$$

The vertical velocity and acceleration are in phase with those at the water surface (kinematic condition). It yields

$$\omega|_{\sigma=0} = 0 \tag{B.15}$$

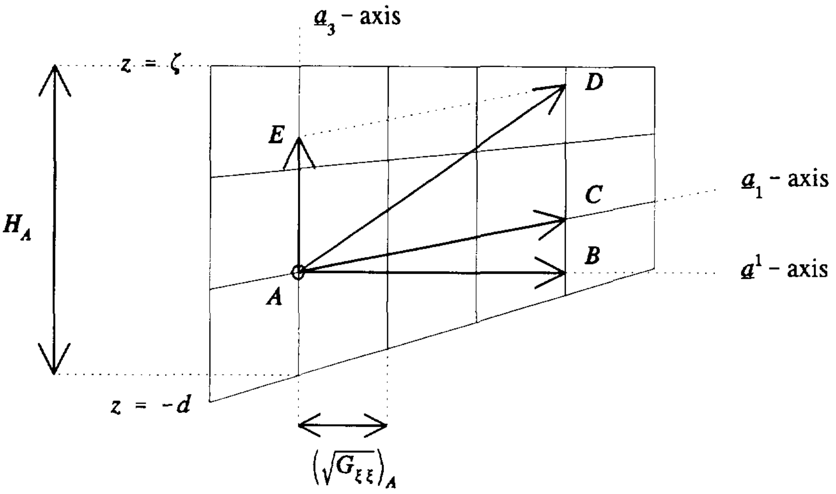

$\underset{\sim}{a}_3$ - axis
$z = \zeta$
D
E
H_A
C
$\underset{\sim}{a}_1$ - axis
B
$\underset{\sim}{a}^1$ - axis
A
$z = -d$
$\left(\sqrt{G_{\xi\xi}}\right)_A$

Figure (B.2): Velocity vector visualized in vertical plane.

$$\vec{AD} = u\underset{\sim}{e}_1 + w\underset{\sim}{e}_3 = \frac{u_\xi}{\left(\sqrt{G_{\xi\xi}}\right)_A}\underset{\sim}{a}_1 + \frac{\omega}{H_A}\underset{\sim}{a}_3,$$

$$|\vec{AB}| = u_\xi, \quad |\vec{AE}| = |\vec{CD}| = \omega, \qquad \text{and} \qquad |\vec{BC}| = \frac{u_\xi}{\left(\sqrt{G_{\xi\xi}}\right)_A}\left(\frac{\partial z}{\partial \xi}\right)_A.$$

Similar at the bottom

$$\omega|_{\sigma=-1} = 0 \tag{B.16}$$

In the remainder of this appendix some useful transformation relations are summarized.

- Christoffel symbols of the second kind with respect to the horizontal transformation

$$\left\{\begin{aligned} \Gamma^1_{11} &= \frac{\partial\xi}{\partial x}\frac{\partial^2 x}{\partial\xi^2} + \frac{\partial\xi}{\partial y}\frac{\partial^2 y}{\partial\xi^2} \\ \Gamma^1_{12} &= \frac{\partial\xi}{\partial x}\frac{\partial^2 x}{\partial\xi\partial\eta} + \frac{\partial\xi}{\partial y}\frac{\partial^2 y}{\partial\xi\partial\eta} \\ \Gamma^1_{22} &= \frac{\partial\xi}{\partial x}\frac{\partial^2 x}{\partial\eta^2} + \frac{\partial\xi}{\partial y}\frac{\partial^2 y}{\partial\eta^2} \end{aligned}\right. \qquad \left\{\begin{aligned} \Gamma^2_{11} &= \frac{\partial\eta}{\partial x}\frac{\partial^2 x}{\partial\xi^2} + \frac{\partial\eta}{\partial y}\frac{\partial^2 y}{\partial\xi^2} \\ \Gamma^2_{12} &= \frac{\partial\eta}{\partial x}\frac{\partial^2 x}{\partial\xi\partial\eta} + \frac{\partial\eta}{\partial y}\frac{\partial^2 y}{\partial\xi\partial\eta} \\ \Gamma^2_{22} &= \frac{\partial\eta}{\partial x}\frac{\partial^2 x}{\partial\eta^2} + \frac{\partial\eta}{\partial y}\frac{\partial^2 y}{\partial\eta^2} \end{aligned}\right. \tag{B.17}$$

- Dot product relations of contravariant base vectors

$$\left\{\begin{aligned} \left(\underline{a}^1\cdot\underline{a}^2\right) &= -\frac{G_{\xi\eta}}{G} \\ \left(\underline{a}^1\cdot\underline{a}^3\right) &= \frac{1}{HG}\left(-G_{\eta\eta}\frac{\partial z}{\partial\xi} + G_{\xi\eta}\frac{\partial z}{\partial\eta}\right) \\ \left(\underline{a}^2\cdot\underline{a}^3\right) &= \frac{1}{HG}\left(G_{\xi\eta}\frac{\partial z}{\partial\xi} - G_{\xi\xi}\frac{\partial z}{\partial\eta}\right) \end{aligned}\right. \tag{B.18}$$

$$\left\{\begin{aligned} \|\underline{a}^1\|^2 &= \left(\underline{a}^1\cdot\underline{a}^1\right) = \frac{G_{\eta\eta}}{G} \\ \|\underline{a}^2\|^2 &= \left(\underline{a}^2\cdot\underline{a}^2\right) = \frac{G_{\xi\xi}}{G} \\ \|\underline{a}^3\|^2 &= \left(\underline{a}^3\cdot\underline{a}^3\right) = \frac{1}{H^2}\left(1 + \frac{G_{\eta\eta}}{G}\left(\frac{\partial z}{\partial\xi}\right)^2 + \frac{G_{\xi\xi}}{G}\left(\frac{\partial z}{\partial\eta}\right)^2 - 2\frac{G_{\xi\eta}}{G}\frac{\partial z}{\partial\xi}\frac{\partial z}{\partial\eta}\right) \end{aligned}\right. \tag{B.19}$$

- Sigma derivative relations

$$\begin{aligned} \frac{\partial\sigma}{\partial x} &= \frac{1}{\sqrt{G}}\left(\frac{\partial y}{\partial\eta}\frac{\partial\sigma}{\partial\xi} - \frac{\partial y}{\partial\xi}\frac{\partial\sigma}{\partial\eta}\right) \\ \frac{\partial\sigma}{\partial y} &= \frac{1}{\sqrt{G}}\left(-\frac{\partial x}{\partial\eta}\frac{\partial\sigma}{\partial\xi} + \frac{\partial x}{\partial\xi}\frac{\partial\sigma}{\partial\eta}\right) \\ \frac{\partial\sigma}{\partial t} &= -\frac{1}{H}\left(\frac{\partial z}{\partial t}\right)_{(\xi,\eta,\sigma)} \end{aligned} \tag{B.20}$$

- Chain rule relations for $\tilde{\varphi}(\xi,\eta,\sigma,t) = \varphi(x,y,z,t)$

$$\begin{aligned} \frac{\partial\varphi}{\partial x} &= \frac{1}{\sqrt{G}}\left[\frac{\partial y}{\partial\eta}\left(\frac{\partial\tilde{\varphi}}{\partial\xi} + \frac{\partial\sigma}{\partial\xi}\frac{\partial\tilde{\varphi}}{\partial\sigma}\right) - \frac{\partial y}{\partial\xi}\left(\frac{\partial\tilde{\varphi}}{\partial\eta} + \frac{\partial\sigma}{\partial\eta}\frac{\partial\tilde{\varphi}}{\partial\sigma}\right)\right] \\ \frac{\partial\varphi}{\partial y} &= \frac{1}{\sqrt{G}}\left[-\frac{\partial x}{\partial\eta}\left(\frac{\partial\tilde{\varphi}}{\partial\xi} + \frac{\partial\sigma}{\partial\xi}\frac{\partial\tilde{\varphi}}{\partial\sigma}\right) + \frac{\partial x}{\partial\xi}\left(\frac{\partial\tilde{\varphi}}{\partial\eta} + \frac{\partial\sigma}{\partial\eta}\frac{\partial\tilde{\varphi}}{\partial\sigma}\right)\right] \\ \frac{\partial\varphi}{\partial z} &= \frac{1}{H}\frac{\partial\tilde{\varphi}}{\partial\sigma} \\ \frac{\partial\varphi}{\partial t} &= \frac{\partial\tilde{\varphi}}{\partial t} + \frac{\partial\sigma}{\partial t}\frac{\partial\tilde{\varphi}}{\partial\sigma} = \frac{\partial\tilde{\varphi}}{\partial t} - \frac{1}{H}\left(\frac{\partial z}{\partial t}\right)_{(\xi,\eta,\sigma)}\frac{\partial\tilde{\varphi}}{\partial\sigma} \end{aligned} \tag{B.21}$$

- Time derivative relations to be used in the derivation of momentum equations in transformed coordinates

In section 2.4 the momentum equations in ξ - and η - direction are written as

$$\left\{\begin{aligned} \frac{\partial u}{\partial t} &= F_u(x,y,z,t) \\ \frac{\partial v}{\partial t} &= F_v(x,y,z,t) \end{aligned}\right. \tag{B.22}$$

The time-dependent transformed expression becomes

$$\left\{\begin{aligned} \frac{\partial u_\xi}{\partial t} &= \frac{1}{\sqrt{G_{\eta\eta}}}\left(\frac{\partial y}{\partial \eta}\tilde{F}_u - \frac{\partial x}{\partial \eta}\tilde{F}_v\right) + \frac{1}{H}\left(\frac{\partial z}{\partial t}\right)_{(\xi,\eta,\sigma)}\frac{\partial u_\xi}{\partial \sigma} \\ \frac{\partial u_\eta}{\partial t} &= \frac{1}{\sqrt{G_{\xi\xi}}}\left(-\frac{\partial y}{\partial \xi}\tilde{F}_u + \frac{\partial x}{\partial \xi}\tilde{F}_v\right) + \frac{1}{H}\left(\frac{\partial z}{\partial t}\right)_{(\xi,\eta,\sigma)}\frac{\partial u_\eta}{\partial \sigma} \end{aligned}\right. \tag{B.23}$$

where $\tilde{F}_u = \tilde{F}_u(\xi,\eta,\sigma,t) = (F_u)_{(\xi,\eta,\sigma)} = F_u(x(\xi,\eta),y(\xi,\eta),z(\xi,\eta,\sigma,t),t)$ and similar for $\tilde{F}_v$. The latter terms of equation (B.23) account for the grid-point speed of the moving grid. These contributions will be denoted with G.P.S$_\xi$ and G.P.S$_\eta$ respectively. Subsequently, the contributions arising in the momentum equations will be inserted.

(i) advection

$$\left\{\begin{aligned} F_u &= -u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} - w\frac{\partial u}{\partial z} \\ F_v &= -u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y} - w\frac{\partial v}{\partial z} \end{aligned}\right. \qquad \Rightarrow \tag{B.24}$$

$$\begin{aligned} \frac{\partial u_\xi}{\partial t} = &-\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}u_\xi\frac{\partial u_\xi}{\partial \xi} - \frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}u_\eta\frac{\partial u_\xi}{\partial \eta} - \frac{\omega}{H}\frac{\partial u_\xi}{\partial \sigma} \\ &- u_\xi^{\,2}\left(\frac{\partial}{\partial \xi}\left(\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}\right) + \frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}\Gamma^1_{11}\right) \\ &- u_\xi u_\eta\left(\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial \eta}\left(\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}\right) + 2\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}\Gamma^1_{12}\right) \\ &- u_\eta^{\,2}\left(\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}}\Gamma^1_{22}\right) \end{aligned} \tag{B.25}$$

$$\begin{aligned}\frac{\partial u_\eta}{\partial t} = &-\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}u_\xi\frac{\partial u_\eta}{\partial \xi} - \frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}u_\eta\frac{\partial u_\eta}{\partial \eta} - \frac{\omega}{H}\frac{\partial u_\eta}{\partial \sigma}\\ &-{u_\xi}^2\left(\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}}\Gamma^2_{11}\right)\\ &-u_\xi u_\eta\left(\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}}\frac{\partial}{\partial \xi}\left(\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}\right) + 2\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}\Gamma^2_{12}\right)\\ &-{u_\eta}^2\left(\frac{\partial}{\partial \eta}\left(\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}\right) + \frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}\Gamma^2_{22}\right)\end{aligned} \tag{B.26}$$

G.P.S$_\xi$ and G.P.S$_\eta$ are part of the ω - velocity component

(ii) hydraulic pressure

$$\left\{\begin{aligned} F_u &= -\frac{1}{\rho}\frac{\partial P}{\partial x}\\ F_v &= -\frac{1}{\rho}\frac{\partial P}{\partial y}\end{aligned}\right. \qquad \Rightarrow \tag{B.27}$$

$$\left\{\begin{aligned} \frac{\partial u_\xi}{\partial t} &= -\frac{1}{\tilde{\rho}}\left[\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G}}\left(\frac{\partial \tilde{P}}{\partial \xi} + \frac{\partial \tilde{P}}{\partial \sigma}\frac{\partial \sigma}{\partial \xi}\right) - \frac{G_{\xi\eta}}{\sqrt{G}\sqrt{G_{\eta\eta}}}\left(\frac{\partial \tilde{P}}{\partial \eta} + \frac{\partial \tilde{P}}{\partial \sigma}\frac{\partial \sigma}{\partial \eta}\right)\right] + \mathrm{GPS}_\xi\\ \frac{\partial u_\eta}{\partial t} &= -\frac{1}{\tilde{\rho}}\left[\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G}}\left(\frac{\partial \tilde{P}}{\partial \eta} + \frac{\partial \tilde{P}}{\partial \sigma}\frac{\partial \sigma}{\partial \eta}\right) - \frac{G_{\xi\eta}}{\sqrt{G}\sqrt{G_{\xi\xi}}}\left(\frac{\partial \tilde{P}}{\partial \xi} + \frac{\partial \tilde{P}}{\partial \sigma}\frac{\partial \sigma}{\partial \xi}\right)\right] + \mathrm{GPS}_\eta\end{aligned}\right. \tag{B.28}$$

(iii) Coriolis

$$\left\{\begin{aligned} F_u &= fv\\ F_v &= -fu\end{aligned}\right. \qquad \Rightarrow \tag{B.29}$$

$$\left\{\begin{aligned} \frac{\partial u_\xi}{\partial t} &= f\left[\frac{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}{\sqrt{G}}u_\eta + \frac{G_{\xi\eta}}{\sqrt{G}}u_\xi\right] + \mathrm{GPS}_\xi\\ \frac{\partial u_\eta}{\partial t} &= -f\left[\frac{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}{\sqrt{G}}u_\xi + \frac{G_{\xi\eta}}{\sqrt{G}}u_\eta\right] + \mathrm{GPS}_\eta\end{aligned}\right. \tag{B.30}$$

(iv) vertical eddy viscosity

$$\begin{cases} F_u = \dfrac{\partial}{\partial z}\left(\nu_t^V \dfrac{\partial u}{\partial z}\right) \\ F_v = \dfrac{\partial}{\partial z}\left(\nu_t^V \dfrac{\partial v}{\partial z}\right) \end{cases} \qquad \Rightarrow \tag{B.31}$$

$$\begin{cases} \dfrac{\partial u_\xi}{\partial t} = \dfrac{1}{H^2}\dfrac{\partial}{\partial \sigma}\left(\nu_t^V \dfrac{\partial u_\xi}{\partial \sigma}\right) + \mathbf{GPS}_\xi \\ \dfrac{\partial u_\eta}{\partial t} = \dfrac{1}{H^2}\dfrac{\partial}{\partial \sigma}\left(\nu_t^V \dfrac{\partial u_\eta}{\partial \sigma}\right) + \mathbf{GPS}_\eta \end{cases} \tag{B.32}$$

(v) horizontal eddy viscosity

$$\begin{cases} F_u = 2\dfrac{\partial}{\partial x}\left(\nu_t^H \dfrac{\partial u}{\partial x}\right) + \dfrac{\partial}{\partial y}\left(\nu_t^H\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right)\right) \\ F_v = \dfrac{\partial}{\partial x}\left(\nu_t^H\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right)\right) + 2\dfrac{\partial}{\partial y}\left(\nu_t^H \dfrac{\partial v}{\partial y}\right) \end{cases} \tag{B.33}$$

Because of complexity the explicit expressions are skipped here. A complete computation does not serve numerical purposes. It is discussed in chapter 2 and 3 that the numerical treatment of the horizontal eddy viscosity is completely performed in Cartesian coordinates.

It is emphasized that ν_t^H and ν_t^V still represent eddy viscosities relative to the horizontal and vertical direction respectively. These quantities are not being transformed.

- The hydrostatic pressure relation

$$\frac{\partial \tilde{P}}{\partial \sigma} = -\tilde{\rho} g H \tag{B.34}$$

- Orthogonality relations

Orthogonality of the horizontal transformation (B.1) yields simplified expressions, because

$$G_{\xi\eta} = 0 \tag{B.35}$$

or equivalently,

$$\sqrt{G} = \sqrt{G_{\xi\xi}}\,\sqrt{G_{\eta\eta}} \tag{B.36}$$

The Christoffel symbols are usually written in terms of components of the covariant metric tensor. In case of orthogonality the expressions given in (B.17) reduce to

$$\left\{\begin{aligned} \Gamma^1_{11} &= \frac{1}{\sqrt{G_{\xi\xi}}}\frac{\partial}{\partial\xi}\left(\sqrt{G_{\xi\xi}}\right) \\ \Gamma^1_{12} &= \frac{1}{\sqrt{G_{\xi\xi}}}\frac{\partial}{\partial\eta}\left(\sqrt{G_{\xi\xi}}\right) \\ \Gamma^1_{22} &= -\frac{\sqrt{G_{\eta\eta}}}{G_{\xi\xi}}\frac{\partial}{\partial\xi}\left(\sqrt{G_{\eta\eta}}\right) \end{aligned}\right. \qquad \left\{\begin{aligned} \Gamma^2_{11} &= -\frac{\sqrt{G_{\xi\xi}}}{G_{\eta\eta}}\frac{\partial}{\partial\eta}\left(\sqrt{G_{\xi\xi}}\right) \\ \Gamma^2_{12} &= \frac{1}{\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial\xi}\left(\sqrt{G_{\eta\eta}}\right) \\ \Gamma^2_{22} &= \frac{1}{\sqrt{G_{\eta\eta}}}\frac{\partial}{\partial\eta}\left(\sqrt{G_{\eta\eta}}\right) \end{aligned}\right. \qquad \text{(B.37)}$$

The overall transformation is orthogonal only if in addition to (B.35) $\partial z/\partial\xi = \partial z/\partial\eta = 0$.

Appendix C: Transport equation in transformed coordinates

The transport equation written in Cartesian coordinates is transformed with respect to a curvilinear transformation in the horizontal plane and the sigma transformation in the vertical plane. The used notation and transformation relations are summarized in appendix B. Simplified expressions are summarized in case of orthogonality. Inconsistent implementation of transformed transport equations may introduce an artificial transport process. The phenomenon known as artificial creeping will be discussed briefly.

- Transport equation in Cartesian coordinates (x,y,z)

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} + v\frac{\partial C}{\partial y} + w\frac{\partial C}{\partial z} = \frac{\partial}{\partial x}\left(D_H\frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(D_H\frac{\partial C}{\partial y}\right) + \frac{\partial}{\partial z}\left(D_V\frac{\partial C}{\partial z}\right) \qquad \text{(C.1)}$$

- Transport equation in transformed coordinates (ξ,η,σ); curvilinear transformation in horizontal plane, not necessarily orthogonal, and the sigma transformation in the vertical plane

$$\frac{\partial C}{\partial t} + \frac{1}{\sqrt{G}}\left(\sqrt{G_{\eta\eta}}\,u_\xi\frac{\partial C}{\partial\xi} + \sqrt{G_{\xi\xi}}\,u_\eta\frac{\partial C}{\partial\eta}\right) + \frac{\omega}{H}\frac{\partial C}{\partial\sigma} = \mathbf{DP} \qquad \text{(C.2)}$$

Due to the sigma transformation the diffusive part, **DP**, becomes complex. It equals

$$
\begin{aligned}
\mathrm{DP} = \frac{1}{\sqrt{G}} \quad & \left\{ \frac{G_{\eta\eta}}{\sqrt{G}} \left(\frac{\partial}{\partial \xi} + \frac{\partial \sigma}{\partial \xi} \frac{\partial}{\partial \sigma} \right) \left[D_H \left(\frac{\partial C}{\partial \xi} + \frac{\partial \sigma}{\partial \xi} \frac{\partial C}{\partial \sigma} \right) \right] \right. \\
& - \frac{G_{\xi\eta}}{\sqrt{G}} \left(\frac{\partial}{\partial \eta} + \frac{\partial \sigma}{\partial \eta} \frac{\partial}{\partial \sigma} \right) \left[D_H \left(\frac{\partial C}{\partial \xi} + \frac{\partial \sigma}{\partial \xi} \frac{\partial C}{\partial \sigma} \right) \right] \\
& - \frac{G_{\xi\eta}}{\sqrt{G}} \left(\frac{\partial}{\partial \xi} + \frac{\partial \sigma}{\partial \xi} \frac{\partial}{\partial \sigma} \right) \left[D_H \left(\frac{\partial C}{\partial \eta} + \frac{\partial \sigma}{\partial \eta} \frac{\partial C}{\partial \sigma} \right) \right] \\
& \left. + \frac{G_{\xi\xi}}{\sqrt{G}} \left(\frac{\partial}{\partial \eta} + \frac{\partial \sigma}{\partial \eta} \frac{\partial}{\partial \sigma} \right) \left[D_H \left(\frac{\partial C}{\partial \eta} + \frac{\partial \sigma}{\partial \eta} \frac{\partial C}{\partial \sigma} \right) \right] \right\} \\
+ \frac{D_H}{\sqrt{G}} & \left(\frac{\partial}{\partial \xi} \left(\frac{G_{\eta\eta}}{\sqrt{G}} \right) - \frac{\partial}{\partial \eta} \left(\frac{G_{\xi\eta}}{\sqrt{G}} \right) \right) \left(\frac{\partial C}{\partial \xi} + \frac{\partial \sigma}{\partial \xi} \frac{\partial C}{\partial \sigma} \right) \\
+ \frac{D_H}{\sqrt{G}} & \left(- \frac{\partial}{\partial \xi} \left(\frac{G_{\xi\eta}}{\sqrt{G}} \right) + \frac{\partial}{\partial \eta} \left(\frac{G_{\xi\xi}}{\sqrt{G}} \right) \right) \left(\frac{\partial C}{\partial \eta} + \frac{\partial \sigma}{\partial \eta} \frac{\partial C}{\partial \sigma} \right) \\
+ \frac{1}{H^2} & \frac{\partial}{\partial \sigma} \left(D_V \frac{\partial C}{\partial \sigma} \right)
\end{aligned}
\tag{C.3}
$$

In case of an orthogonal horizontal transformation **DP** reduces to

$$
\begin{aligned}
\mathrm{DP} = & \frac{1}{G_{\xi\xi}} \left(\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}} \frac{\partial}{\partial \xi} \left(\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}} \right) + \frac{\partial}{\partial \xi} + \frac{\partial \sigma}{\partial \xi} \frac{\partial}{\partial \sigma} \right) \left[D_H \left(\frac{\partial C}{\partial \xi} + \frac{\partial \sigma}{\partial \xi} \frac{\partial C}{\partial \sigma} \right) \right] \\
& + \frac{1}{G_{\eta\eta}} \left(\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}} \frac{\partial}{\partial \eta} \left(\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}} \right) + \frac{\partial}{\partial \eta} + \frac{\partial \sigma}{\partial \eta} \frac{\partial}{\partial \sigma} \right) \left[D_H \left(\frac{\partial C}{\partial \eta} + \frac{\partial \sigma}{\partial \eta} \frac{\partial C}{\partial \sigma} \right) \right] \\
& + \frac{1}{H^2} \frac{\partial}{\partial \sigma} \left(D_V \frac{\partial C}{\partial \sigma} \right)
\end{aligned}
\tag{C.4}
$$

- Conservative description

The advective part of equation (C.2) can be written in conservative form if the continuity equation is taken into account. Then C × (equation (2.23a) of section 2.4) + H × (equation (C.2)) yields

$$
\frac{\partial (HC)}{\partial t} + \frac{1}{\sqrt{G}} \left(\frac{\partial}{\partial \xi} \left(H \sqrt{G_{\eta\eta}} u_\xi C \right) + \frac{\partial}{\partial \eta} \left(H \sqrt{G_{\xi\xi}} u_\eta C \right) \right) + \frac{\partial (\omega C)}{\partial \sigma} = H \times \mathrm{DP}
\tag{C.5}
$$

In case of an orthogonal transformation in the horizontal plane and in addition

$\partial\sigma/\partial\xi = \partial\sigma/\partial\eta = 0$ the conservative transport equation becomes

$$\begin{aligned}
&\frac{\partial(HC)}{\partial t} + \frac{1}{\sqrt{G}}\left(\frac{\partial}{\partial\xi}\left(H\sqrt{G_{\eta\eta}}u_\xi C\right) + \frac{\partial}{\partial\eta}\left(H\sqrt{G_{\xi\xi}}u_\eta C\right)\right) + \frac{\partial(\omega C)}{\partial\sigma} \\
&= \frac{H}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\left(\frac{\partial}{\partial\xi}\left(D_H\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}}\frac{\partial C}{\partial\xi}\right) + \frac{\partial}{\partial\eta}\left(D_H\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}}\frac{\partial C}{\partial\eta}\right)\right) \\
&+ \frac{1}{H}\frac{\partial}{\partial\sigma}\left(D_V\frac{\partial C}{\partial\sigma}\right)
\end{aligned} \tag{C.6}$$

- Artificial creeping

An incomplete transformation may introduce an incorrect vertical advection contribution. In case of an orthogonal horizontal transformation the vertical advection term becomes, joining terms $\partial C/\partial\sigma$ from (C.2) using (C.4)

$$\begin{aligned}
&\frac{\partial C}{\partial\sigma}\left\{\frac{\omega}{H} - D_H\left(\frac{\partial^2\sigma}{\partial\xi^2} + \frac{\partial^2\sigma}{\partial\eta^2}\right)\right. \\
&\quad - \frac{\partial\sigma}{\partial\xi}\left(\frac{\partial D_H}{\partial\xi} + \frac{D_H}{\sqrt{G}}\frac{\partial}{\partial\xi}\left(\frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}}\right) + \frac{\partial}{\partial\sigma}\left(D_H\frac{\partial\sigma}{\partial\xi}\right)\right) \\
&\quad \left. - \frac{\partial\sigma}{\partial\eta}\left(\frac{\partial D_H}{\partial\eta} + \frac{D_H}{\sqrt{G}}\frac{\partial}{\partial\eta}\left(\frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}}\right) + \frac{\partial}{\partial\sigma}\left(D_H\frac{\partial\sigma}{\partial\eta}\right)\right)\right\} \\
&= \frac{\partial C}{\partial\sigma}\left\{\frac{\omega - w_a}{H}\right\}
\end{aligned} \tag{C.7}$$

The first term of w_a in (C.7) is often negative. This is illustrated as follows. Assuming $\partial\zeta/\partial\xi = 0$, the following expressions hold

$$\begin{aligned}
\frac{\partial\sigma}{\partial\xi} &= -\frac{\sigma}{H}\frac{\partial H}{\partial\xi} \\
\frac{\partial^2\sigma}{\partial\xi^2} &= \frac{\sigma}{H}\left[\frac{2}{H}\left(\frac{\partial H}{\partial\xi}\right)^2 - \frac{\partial^2 H}{\partial\xi^2}\right]
\end{aligned} \tag{C.8}$$

With negligible bottom curvature the second expression of (C.8) is negative, bearing in mind that $\sigma \leq 0$. So, the sigma transformation gives rise to an additional positive vertical velocity $-w_a$. It is shown in KESTER and UITTENBOGAARD (1990) that this contribution is properly balanced by an additional vertical diffusion term as can be derived from equation (C.4). An incomplete transformation will result in an upward migration of substance.

Appendix D: Advection schemes

The shallow water equations describe, among other things, the advection property of physical quantities. In this appendix the propagation properties of approximate solutions of the advection equation will be studied.

To obtain an accurate and stable discretization scheme for the advection equation various schemes are subjected to the one-dimensional hyperbolic initial-value problem given by

$$\frac{\partial u}{\partial t} + V_F \frac{\partial u}{\partial x} = 0 \tag{D.1}$$

with frozen advection constant V_F. The initial condition represents a single Fourier component with wave speed σ, i.e.,

$$u(x,0) = \exp[i\sigma x] \tag{D.2}$$

This test problem is taken from STELLING (1984). This source includes various spatial discretizations that will be extended here. Besides the propagation properties of (D.1) an advection scheme will be proposed that is suitable for the discretization of the nonlinear shallow water equations. Then the frozen coefficient V_F is substituted by the velocity u itself. The nonlinear advection equation can be written in conservative form. With $V_F = u$ it becomes

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\tfrac{1}{2}u^2\right) = 0 \tag{D.3}$$

Numerical consequences of conservative versus nonconservative discretizations will be reviewed.

- Spatial discretization

Returning to test problem (D.1) a semi-discrete system of ordinary differential equations is given by

$$\begin{cases} \dfrac{\partial u_m}{\partial t} + \mathrm{D}u_m = 0 \\ u_m(0) = \exp[i\sigma m\Delta x] \end{cases} \tag{D.4}$$

with **D** a spatial difference operator and Δx the constant grid size.

The following discretizations are studied, providing $V_F > 0$:

1) 1st order upwind: $\dfrac{\partial u_m}{\partial t} + V_F \dfrac{u_m - u_{m-1}}{\Delta x} = 0$
2) 2nd order upwind: $\dfrac{\partial u_m}{\partial t} + V_F \dfrac{3u_m - 4u_{m-1} + u_{m-2}}{2\Delta x} = 0$
3) 2nd order central: $\dfrac{\partial u_m}{\partial t} + V_F \dfrac{u_{m+1} - u_{m-1}}{2\Delta x} = 0$
4) 3rd order upwind/central: $\dfrac{\partial u_m}{\partial t} + V_F \dfrac{u_{m+2} + 4u_{m+1} + 18u_m - 28u_{m-1} + 5u_{m-2}}{24\Delta x} = 0$
5) 4th order upwind/central: $\dfrac{\partial u_m}{\partial t} + V_F \dfrac{3u_{m+1} + 10u_m - 18u_{m-1} + 6u_{m-2} - u_{m-3}}{12\Delta x} = 0$

Substitution of $u_m(t) = \exp\left[i\sigma m \Delta x - \tilde{\mathbf{D}}t\right]$ into (D.4) yields the following expressions for the Fourier transform $\tilde{\mathbf{D}}$ of **D**

1) $\tilde{\mathbf{D}} = \dfrac{V_F}{\Delta x}\left(1 - e^{-i\sigma\Delta x}\right)$
2) $\tilde{\mathbf{D}} = \dfrac{V_F}{2\Delta x}\left(3 - 4e^{-i\sigma\Delta x} + e^{-2i\sigma\Delta x}\right)$
3) $\tilde{\mathbf{D}} = \dfrac{V_F}{2\Delta x}\left(e^{i\sigma\Delta x} - e^{-i\sigma\Delta x}\right)$
4) $\tilde{\mathbf{D}} = \dfrac{V_F}{24\Delta x}\left(e^{2i\sigma\Delta x} + 4e^{i\sigma\Delta x} + 18 - 28e^{-i\sigma\Delta x} + 5e^{-2i\sigma\Delta x}\right)$
5) $\tilde{\mathbf{D}} = \dfrac{V_F}{12\Delta x}\left(3e^{i\sigma\Delta x} + 10 - 18e^{-i\sigma\Delta x} + 6e^{-2i\sigma\Delta x} - e^{-3i\sigma\Delta x}\right)$

An accuracy measure is found by comparison of the approximate solution $u_m(t)$ of (D.4) with the analytical solution $u(m\Delta x, t)$ of (D.1). Their ratio equals

$$\frac{u_m(t)}{u(m\Delta x, t)} = \exp\left[-\tilde{\mathbf{D}}t + i\sigma V_F t\right] \tag{D.5}$$

The damping/amplification of the approximate solution is quantified by the absolute value of the right-hand side of (D.5). This so-called relative amplitude is given by

$$\exp\left[-\mathrm{Re}\left(\tilde{\mathbf{D}}t\right)\right] \tag{D.6}$$

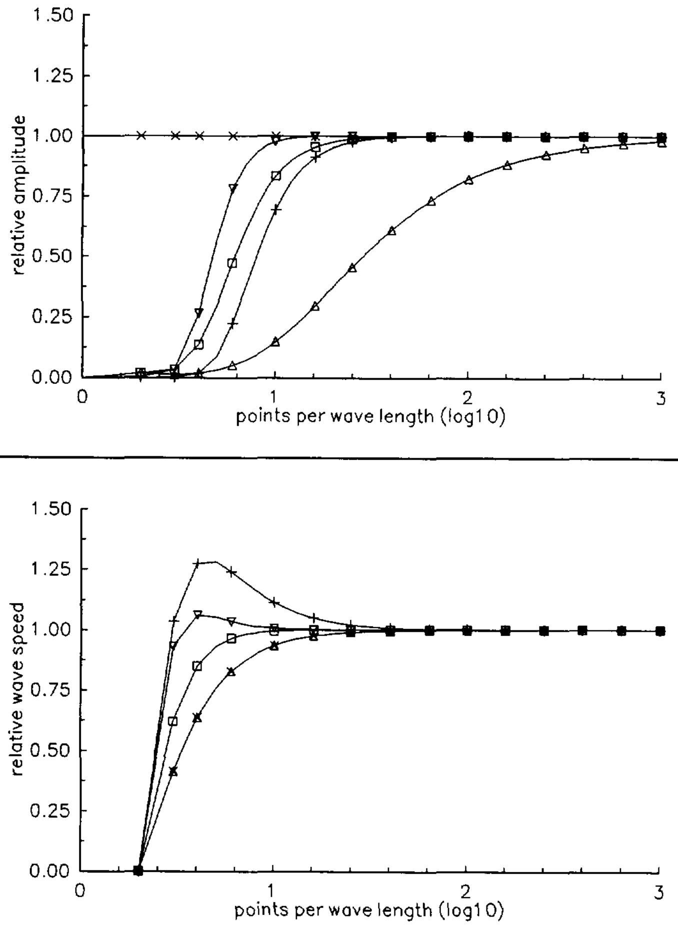


Figure (D.1): Relative amplitude and relative wave speed for one wave period, i.e., $t\sigma = 2\pi$ with $V_p = 1$.

TOP:	relative amplitude (D.6), and
BOTTOM:	relative wave speed (D.7).

Number of points per wave length $= \dfrac{2\pi}{\sigma\Delta x}$.

triangle up:	1st order upwind,	triangle down:	4th order upwind/central,
plus:	2nd order upwind,	cross:	2nd order central,
box:	3rd order upwind/central.		

The phase shift in the approximated solution is related to the imaginary part of the exponential argument of the right-hand side of (D.6). The so-called relative wave speed is then defined as

$$\frac{\mathrm{Im}\left(\tilde{\mathrm{D}}t\right)}{\sigma V_F t} \tag{D.7}$$

The relative amplitude and relative wave speed are plotted in figure (D.1) as a function of the number of points used to represent one wave length. It is observed that the fourth order upwind/central method has nevertheless partly leading and partly lagging phase errors, but provides accurate propagation properties.

- Time discretization

Subsequently, following STELLING (1984), the numerical scheme is discretized in time. A two step method splits the operator D such that $\mathrm{D} = \tfrac{1}{2}\left(\mathrm{D}_1 + \mathrm{D}_2\right)$. The two step split method is written in the following form

$$\begin{aligned} &\text{Step 1:} && \frac{u_m^* - u_m^l}{\tfrac{1}{2}\Delta t} + \mathrm{D}_1 u_m^l = 0 \\ &\text{Step 2:} && \frac{u_m^{l+1} - u_m^*}{\tfrac{1}{2}\Delta t} + \mathrm{D}_2 u_m^{l+1} = 0 \end{aligned} \tag{D.8}$$

A superscript $*$ refers to the intermediate result at time $t = (l+\tfrac{1}{2})\Delta t$ with Δt the time increment. The five operators D given above are split according:

$$1)\ \begin{cases} \text{Step 1:} & \dfrac{u_m^* - u_m^l}{\tfrac{1}{2}\Delta t} + V_F \dfrac{u_m^l - u_{m-1}^l}{\Delta x} = 0 \\[2ex] \text{Step 2:} & \dfrac{u_m^{l+1} - u_m^*}{\tfrac{1}{2}\Delta t} + V_F \dfrac{u_m^{l+1} - u_{m-1}^{l+1}}{\Delta x} = 0 \end{cases} \tag{D.9}$$

$$2)\ \begin{cases} \text{Step 1:} & \dfrac{u_m^* - u_m^l}{\tfrac{1}{2}\Delta t} + V_F \dfrac{3u_m^l - 4u_{m-1}^l + u_{m-2}^l}{2\Delta x} = 0 \\[2ex] \text{Step 2:} & \dfrac{u_m^{l+1} - u_m^*}{\tfrac{1}{2}\Delta t} + V_F \dfrac{3u_m^{l+1} - 4u_{m-1}^{l+1} + u_{m-2}^{l+1}}{2\Delta x} = 0 \end{cases} \tag{D.10}$$

$$3)\left\{\begin{array}{ll}\text{Step 1:} & \dfrac{u_m^* - u_m^l}{\frac{1}{2}\Delta t} + V_F\dfrac{u_{m+1}^l - u_{m-1}^l}{2\Delta x} = 0 \\[2ex] \text{Step 2:} & \dfrac{u_m^{l+1} - u_m^*}{\frac{1}{2}\Delta t} + V_F\dfrac{u_{m+1}^{l+1} - u_{m-1}^{l+1}}{2\Delta x} = 0\end{array}\right. \qquad (D.11)$$

$$4)\left\{\begin{array}{ll}\text{Step 1:} & \dfrac{u_m^* - u_m^l}{\frac{1}{2}\Delta t} + V_F\dfrac{u_{m+2}^l + 4u_{m+1}^l - 4u_{m-1}^l - u_{m-2}^l}{12\Delta x} = 0 \\[2ex] \text{Step 2:} & \dfrac{u_m^{l+1} - u_m^*}{\frac{1}{2}\Delta t} + V_F\dfrac{3u_m^{l+1} - 4u_{m-1}^{l+1} + u_{m-2}^{l+1}}{2\Delta x} = 0\end{array}\right. \qquad (D.12)$$

$$5)\left\{\begin{array}{ll}\text{Step 1:} & \dfrac{u_m^* - u_m^l}{\frac{1}{2}\Delta t} + V_F\dfrac{u_{m+1}^l - u_{m-1}^l}{2\Delta x} = 0 \\[2ex] \text{Step 2:} & \dfrac{u_m^{l+1} - u_m^*}{\frac{1}{2}\Delta t} + V_F\dfrac{10u_m^{l+1} - 15u_{m-1}^{l+1} + 6u_{m-2}^{l+1} - u_{m-3}^{l+1}}{6\Delta x} = 0\end{array}\right. \qquad (D.13)$$

Stable schemes provide that (small) errors made during calculations will not be amplified. These stability aspects are investigated by the behaviour of $\left|u_m^l - u(m\Delta x, l\Delta t)\right|$ as (i)l tends to infinity for fixed Δx, Δt (absolute stability) and (ii) the mesh is refined $\Delta x \downarrow 0$, $\Delta t \downarrow 0$ for fixed $l\Delta t$ (zero stability). To obtain a basis upon which to draw a conclusion about stability a single Fourier component is substituted into the equations and its propagation properties are analyzed. Elimination of the intermediate quantities and substitution of $u_m^l = \rho^l \exp[im\sigma\Delta x]$ yields the following expression for the propagation factor ρ

$$\rho = \frac{1 - \frac{1}{2}\tilde{D}_1\Delta t}{1 + \frac{1}{2}\tilde{D}_2\Delta t} \qquad (D.14)$$

With $|\rho| \leq 1$ for all $0 < \sigma\Delta x < \pi$, still providing $V_F > 0$, the schemes are said to be unconditionally stable. This analysis is only a tool to gain confidence in the numerical schemes. Numerical experiments are still necessary to verify the stability aspects, especially with respect to bounded regions. More details about the concepts of stability analysis are given in e.g., STELLING (1984), RICHTMYER and MORTON (1967) or GODUNOV and RYABENKI (1964).

The linearized stability analysis gives that the solution methods (D.9) until (D.13) are unconditionally stable. In case of the nonlinear advection equation (D.3), including boundary conditions, the stability and accuracy aspects become complex. Due to nonlinearity the propagation is determined by interaction between individual Fourier components. This may affect the expected stability properties based on linear analysis. The schemes described above will be extended to the nonlinear case including boundary conditions. The corresponding schemes will be summarized. Considering equation (D.3) the given schemes can be written in conservative or nonconservative form. Here a spatial difference operator D is said to be conservative if either D represents a spatial discretization of the advection term written in conservative form, or the computation of the volume flux through a grid cell boundary is independent of the evaluation point. For instance the central difference scheme of (D.11) is conservative.

- Discretization schemes in a bounded domain

In a bounded domain M intervals of lengths Δx are defined. The inner u - velocity points are denoted with u_m for $m = 1,..,M-1$ located at the interval sides. The nonlinear analogue of (D.8) becomes

$$\text{Step 1:} \quad \frac{u_m^* - u_m^l}{\frac{1}{2}\Delta t} + \mathrm{D}_1\left(u_m^l, u_m^*\right) = 0$$

$$\text{Step 2:} \quad \frac{u_m^{l+1} - u_m^*}{\frac{1}{2}\Delta t} + \mathrm{D}_2\left(u_m^{l+1}, u_m^*\right) = 0 \tag{D.15}$$

If the inflow condition is prescribed ($u_0 > 0$) the following schemes are distinguished:

- Nonconservative schemes for (D.3), based on (D.9) until (D.13) with $V_F = u_m^*$:

Scheme 1

$$\text{Step 1:} \quad \mathrm{D}_1 = u_m^* \frac{u_m^l - u_{m-1}^l}{\Delta x} \qquad m = 1,..,M-1$$

$$\text{Step 2:} \quad \mathrm{D}_2 = u_m^* \frac{u_m^{l+1} - u_{m-1}^{l+1}}{\Delta x} \qquad m = 1,..,M-1 \tag{D.16}$$

Scheme 2

$$\text{Step 1:} \quad \mathrm{D}_1 = \begin{cases} u_m^* \dfrac{u_m^l - u_{m-1}^l}{\Delta x} & m = 1 \\[2ex] u_m^* \dfrac{3u_m^l - 4u_{m-1}^l + u_{m-2}^l}{2\Delta x} & m = 2,..,M-1 \end{cases}$$

(D.17)

$$\text{Step 2:}\quad \mathrm{D}_2 = \begin{cases} u_m^* \dfrac{u_m^{l+1} - u_{m-1}^{l+1}}{\Delta x} & m = 1 \\[2ex] u_m^* \dfrac{3u_m^{l+1} - 4u_{m-1}^{l+1} + u_{m-2}^{l+1}}{2\Delta x} & m = 2,..,M-1 \end{cases}$$

Scheme 3

$$\text{Step 1:}\quad \mathrm{D}_1 = \begin{cases} u_m^* \dfrac{u_{m+1}^{l} - u_m^{l}}{\Delta x} & m = 1 \\[2ex] u_m^* \dfrac{u_{m+2}^{l} + 4u_{m+1}^{l} - 4u_{m-1}^{l} - u_{m-2}^{l}}{12\Delta x} & m = 2,..,M-3 \\[2ex] u_m^* \dfrac{u_{m+1}^{l} - u_{m-1}^{l}}{2\Delta x} & m = M-2 \\[2ex] u_m^* \dfrac{3u_m^{l} - 4u_{m-1}^{l} + u_{m-2}^{l}}{2\Delta x} & m = M-1 \end{cases}$$

(D.18)

$$\text{Step 2:}\quad \mathrm{D}_2 = \begin{cases} u_m^* \dfrac{u_m^{l+1} - u_{m-1}^{l+1}}{\Delta x} & m = 1 \\[2ex] u_m^* \dfrac{3u_m^{l+1} - 4u_{m-1}^{l+1} + u_{m-2}^{l+1}}{2\Delta x} & m = 2,..,M-1 \end{cases}$$

Scheme 4

$$\text{Step 1:}\quad \mathrm{D}_1 = \begin{cases} u_m^* \dfrac{u_{m+1}^{l} - u_{m-1}^{l}}{2\Delta x} & m = 1,..,M-2 \\[2ex] u_m^* \dfrac{u_m^{l} - u_{m-1}^{l}}{\Delta x} & m = M-1 \end{cases}$$

(D.19)

$$\text{Step 2:}\quad \mathrm{D}_2 = \begin{cases} u_m^* \dfrac{-2u_{m+2}^{l+1} + 9u_{m+1}^{l+1} - 6u_m^{l+1} - u_{m-1}^{l+1}}{6\Delta x} & m = 1 \\[2ex] u_m^* \dfrac{u_{m+1}^{l+1} + 6u_m^{l+1} - 9u_{m-1}^{l+1} + 2u_{m-2}^{l+1}}{6\Delta x} & m = 2 \\[2ex] u_m^* \dfrac{10u_m^{l+1} - 15u_{m-1}^{l+1} + 6u_{m-2}^{l+1} - u_{m-3}^{l+1}}{6\Delta x} & m = 3,..,M-2 \\[2ex] u_m^* \dfrac{8u_m^{l+1} - 15u_{m-1}^{l+1} + 9u_{m-2}^{l+1} - 2u_{m-3}^{l+1}}{3\Delta x} & m = M-1 \end{cases}$$

At the inner points these schemes are first, second, third and fourth order consistent in space respectively. All schemes are second order consistent in time. The boundary treatment is chosen such that the order of consistency in space is one less than the order at the inner points after elimination of the intermediate quantity. Away from the boundary all arising algebraic equations are solved in one sweep. In scheme 3 and 4 the efficiency in solving the algebraic equations is damaged near the boundary, since the upwind character has disappeared. It hardly affects the overall efficiency.

- Conservative schemes for (D.3):

Scheme 5

$$\begin{aligned} &\text{Step 1:}\quad \mathrm{D}_1 = \frac{\tfrac{1}{2}u_m^* u_m^l - \tfrac{1}{2}u_{m-1}^* u_{m-1}^l}{\Delta x} && m = 1,..,M-1 \\ &\text{Step 2:}\quad \mathrm{D}_2 = \frac{\tfrac{1}{2}u_m^* u_m^{l+1} - \tfrac{1}{2}u_{m-1}^* u_{m-1}^{l+1}}{\Delta x} && m = 1,..,M-1 \end{aligned} \tag{D.20}$$

Scheme 6

$$\begin{aligned} &\text{Step 1:}\quad \mathrm{D}_1 = \begin{cases} \dfrac{\tfrac{1}{2}u_m^* u_m^l - \tfrac{1}{2}u_{m-1}^* u_{m-1}^l}{\Delta x} & m = 1 \\[2ex] \dfrac{3\left(\tfrac{1}{2}u_m^* u_m^l\right) - 4\left(\tfrac{1}{2}u_{m-1}^* u_{m-1}^l\right) + \tfrac{1}{2}u_{m-2}^* u_{m-2}^l}{2\Delta x} & m = 2,..,M-1 \end{cases} \\ &\text{Step 2:}\quad \mathrm{D}_2 = \begin{cases} \dfrac{\tfrac{1}{2}u_m^* u_m^{l+1} - \tfrac{1}{2}u_{m-1}^* u_{m-1}^{l+1}}{\Delta x} & m = 1 \\[2ex] \dfrac{3\left(\tfrac{1}{2}u_m^* u_m^{l+1}\right) - 4\left(\tfrac{1}{2}u_{m-1}^* u_{m-1}^{l+1}\right) + \tfrac{1}{2}u_{m-2}^* u_{m-2}^{l+1}}{2\Delta x} & m = 2,..,M- \end{cases} \end{aligned} \tag{D.21}$$

Scheme 7

$$\text{Step 1:}\quad \mathrm{D}_1 = \begin{cases} u_m^* \dfrac{u_{m+1}^l - u_{m-1}^l}{2\Delta x} & m = 1,..,M-2 \\[2ex] u_m^* \dfrac{u_m^l - u_{m-1}^l}{\Delta x} & m = M-1 \end{cases} \tag{D.22}$$

$$\text{Step 2:}\quad D_2 = \begin{cases} u_m^* \dfrac{u_{m+1}^{l+1} - u_{m-1}^{l+1}}{2\Delta x} & m = 1,..,M-2 \\[2ex] u_m^* \dfrac{u_m^{l+1} - u_{m-1}^{l+1}}{\Delta x} & m = M-1 \end{cases}$$

<u>Scheme 8</u>

$$\text{Step 1:}\quad D_1 = \begin{cases} u_m^* \dfrac{u_{m+1}^{l} - u_{m-1}^{l}}{2\Delta x} & m = 1,..,M-2 \\[2ex] u_m^* \dfrac{u_m^{l} - u_{m-1}^{l}}{\Delta x} & m = M-1 \end{cases}$$

(D.23)

Step 2:

$$D_2 = \begin{cases} \dfrac{-2\left(\tfrac{1}{2}u_{m+2}^* u_{m+2}^{l+1}\right) + 9\left(\tfrac{1}{2}u_{m+1}^* u_{m+1}^{l+1}\right) - 6\left(\tfrac{1}{2}u_m^* u_m^{l+1}\right) - \tfrac{1}{2}u_{m-1}^* u_{m-1}^{l+1}}{6\Delta x} & m = 1 \\[2ex] \dfrac{\tfrac{1}{2}u_{m+1}^* u_{m+1}^{l+1} + 6\left(\tfrac{1}{2}u_m^* u_m^{l+1}\right) - 9\left(\tfrac{1}{2}u_{m-1}^* u_{m-1}^{l+1}\right) + 2\left(\tfrac{1}{2}u_{m-2}^* u_{m-2}^{l+1}\right)}{6\Delta x} & m = 2 \\[2ex] \dfrac{10\left(\tfrac{1}{2}u_m^* u_m^{l+1}\right) - 15\left(\tfrac{1}{2}u_{m-1}^* u_{m-1}^{l+1}\right) + 6\left(\tfrac{1}{2}u_{m-2}^* u_{m-2}^{l+1}\right) - \tfrac{1}{2}u_{m-3}^* u_{m-3}^{l+1}}{6\Delta x} & m = 3,..,M-2 \\[2ex] \dfrac{8\left(\tfrac{1}{2}u_m^* u_m^{l+1}\right) - 15\left(\tfrac{1}{2}u_{m-1}^* u_{m-1}^{l+1}\right) + 9\left(\tfrac{1}{2}u_{m-2}^* u_{m-2}^{l+1}\right) - 2\left(\tfrac{1}{2}u_{m-3}^* u_{m-3}^{l+1}\right)}{3\Delta x} & m = M-1 \end{cases}$$

Scheme 5 is first order consistent in space. Schemes 6 and 7 are second order consistent in space. Scheme 7 is not written in conservative form, but represents a conservative scheme at the inner points due to central differencing. Scheme 8 is the fourth order conservative analogue of scheme 4. The conservative analogue of scheme 3 is inefficient. The corresponding matrix system contains 5 non-zero diagonals. Each diagonal contains coefficients for u_{m+2}, u_{m+1}, u_m, u_{m-1}, u_{m-2}. The matrix system can not be solved in less than three sweeps. For that reason this scheme will be left out of considerations. Finally two schemes are proposed in which the central differencing scheme is also written in conservative form.

- Conservative schemes for (D.3) continued:

Scheme 9

$$
\text{Step 1:}\quad \mathrm{D}_1 = \begin{cases} \dfrac{\tfrac{1}{2}u^{*}_{m+1}u^{l}_{m+1} - \tfrac{1}{2}u^{*}_{m-1}u^{l}_{m-1}}{2\Delta x} & m = 1,..,M-2 \\[2ex] \dfrac{\tfrac{1}{2}u^{*}_{m}u^{l}_{m} - \tfrac{1}{2}u^{*}_{m-1}u^{l}_{m-1}}{\Delta x} & m = M-1 \end{cases}
$$

$$
\text{Step 2:}\quad \mathrm{D}_2 = \begin{cases} \dfrac{\tfrac{1}{2}u^{*}_{m+1}u^{l+1}_{m+1} - \tfrac{1}{2}u^{*}_{m-1}u^{l+1}_{m-1}}{2\Delta x} & m = 1,..,M-2 \\[2ex] \dfrac{\tfrac{1}{2}u^{*}_{m}u^{l+1}_{m} - \tfrac{1}{2}u^{*}_{m-1}u^{l+1}_{m-1}}{\Delta x} & m = M-1 \end{cases} \tag{D.24}
$$

Scheme 10

$$
\text{Step 1:}\quad \mathrm{D}_1 = \begin{cases} \dfrac{\tfrac{1}{2}u^{*}_{m+1}u^{l}_{m+1} - \tfrac{1}{2}u^{*}_{m-1}u^{l}_{m-1}}{2\Delta x} & m = 1,..,M-2 \\[2ex] \dfrac{\tfrac{1}{2}u^{*}_{m}u^{l}_{m} - \tfrac{1}{2}u^{*}_{m-1}u^{l}_{m-1}}{\Delta x} & m = M-1 \end{cases}
$$

Step 2:

$$
\mathrm{D}_2 = \begin{cases} \dfrac{-2\left(\tfrac{1}{2}u^{*}_{m+2}u^{l+1}_{m+2}\right) + 9\left(\tfrac{1}{2}u^{*}_{m+1}u^{l+1}_{m+1}\right) - 6\left(\tfrac{1}{2}u^{*}_{m}u^{l+1}_{m}\right) - \tfrac{1}{2}u^{*}_{m-1}u^{l+1}_{m-1}}{6\Delta x} & m = 1 \\[2ex] \dfrac{\tfrac{1}{2}u^{*}_{m+1}u^{l+1}_{m+1} + 6\left(\tfrac{1}{2}u^{*}_{m}u^{l+1}_{m}\right) - 9\left(\tfrac{1}{2}u^{*}_{m-1}u^{l+1}_{m-1}\right) + 2\left(\tfrac{1}{2}u^{*}_{m-2}u^{l+1}_{m-2}\right)}{6\Delta x} & m = 2 \\[2ex] \dfrac{10\left(\tfrac{1}{2}u^{*}_{m}u^{l+1}_{m}\right) - 15\left(\tfrac{1}{2}u^{*}_{m-1}u^{l+1}_{m-1}\right) + 6\left(\tfrac{1}{2}u^{*}_{m-2}u^{l+1}_{m-2}\right) - \tfrac{1}{2}u^{*}_{m-3}u^{l+1}_{m-3}}{6\Delta x} & m = 3,..,M-2 \\[2ex] \dfrac{8\left(\tfrac{1}{2}u^{*}_{m}u^{l+1}_{m}\right) - 15\left(\tfrac{1}{2}u^{*}_{m-1}u^{l+1}_{m-1}\right) + 9\left(\tfrac{1}{2}u^{*}_{m-2}u^{l+1}_{m-2}\right) - 2\left(\tfrac{1}{2}u^{*}_{m-3}u^{l+1}_{m-3}\right)}{3\Delta x} & m = M-1 \end{cases} \tag{D.25}
$$

The linear stability analysis does not discriminate between conservative and nonconservative schemes. Further accuracy aspects are studied by means of a numerical experiment. The experiment deals with the one-dimensional Bernoulli equation in a shallow water channel with a threshold.

- A numerical experiment: flow over a threshold

The Bernoulli equation arises from the stationary momentum equation neglecting shear stresses. The simplified one-dimensional shallow water equations are, reminding $H = \zeta + d$:

The continuity equation: $$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(Hu) = 0 \qquad \text{(D.26a)}$$

The momentum equation: $$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial \zeta}{\partial x} = 0 \qquad \text{(D.26b)}$$

Without a friction contribution the stationary state solution is known for given bathymetry. This solution will be used to analyze the global differences between various spatial discretizations considering subcritical flow. In three dimensions a spatial staggered grid is used. Although in one dimension it is known that the non-staggered box scheme gives accurate solutions only staggered grids will be discussed - see MITCHELL and GRIFFITHS (1980). Solving (D.26) numerically together with prescribed boundary conditions yields an approximate water level elevation and a velocity distribution along the channel. Here, the main interest involves the accuracy of the approximated water level elevation. The analytical stationary state solution shows that the observed (global) head loss over the threshold is absent (no energy losses). This is illustrated in figure (D.2) where the stationary state solution is plotted given a prescribed bottom configuration.

Numerical issues will be outlined continued with numerical experiments. These experiments focus on the presence of numerical head loss.

- Grid staggering

The length of the channel, L, is subdivided into M intervals. The inflow boundary is positioned at $x = ½\Delta x$, while the outflow condition is located at $x = L = M\Delta x$. The grid is sketched in figure (D.3) with $\zeta_m = \zeta(m\Delta x)$, $u_m = u((m+½)\Delta x)$ and $d_m = d((m+½)\Delta x)$.

- Boundary conditions

The staggered grid makes it easy to implement boundary conditions. At the outflow boundary the water level is prescribed.

$$\zeta = \zeta^{\text{outflow}}(t) \qquad \text{(D.27)}$$

At the inflow boundary the velocity at stationary state is given. These boundary conditions are known as reflective boundary conditions. The waves coming from the interior will numerically be reflected at the boundary. This is physically incorrect. To avoid oscillations

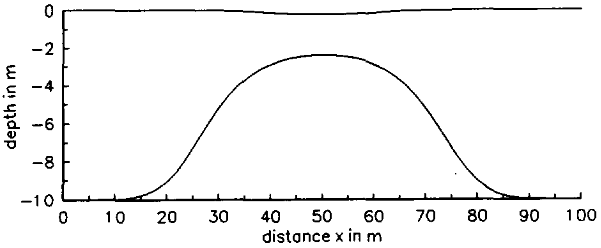


Figure (D.2): Geometry of channel (bathymetry and water level).

introduced by reflections at the boundary, dissipation is needed. A way of introducing (artificial) dissipation is to make use of the stabilizing effect of so-called Riemann invariants - see OLIGER and SUNDSTRÖM (1978). These Riemann invariants are defined as (nonphysical) quantities that remain constant while moving along the characteristics in the absence of friction - see VREUGDENHIL (1989). The ingoing Riemann invariant is given by $u + 2\sqrt{gH}$. With reflection coefficient ϵ, the stabilizing effect is experienced by setting

$$u + \epsilon\frac{\partial}{\partial t}\left(u + 2\sqrt{gH}\right) = u^{\text{inflow}}(t) \tag{D.28}$$

Doing so, the velocity will be prescribed at stationary state. Moreover, waves coming from the interior will be partially let through - see VERBOOM *et al.* (1981). The coefficient ϵ is not dimensionless. It should be chosen sufficiently small to maintain accuracy at stationary state and it should be chosen sufficiently large such that (D.28) acts as a non-reflective boundary condition. An acceptable choice equals the time needed for a wave to move through the interior - see VERBOOM *et al.* (1982).

- Discretized time splitting scheme

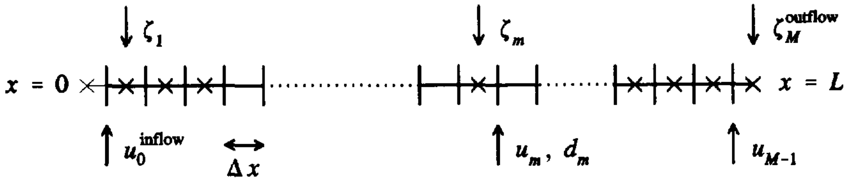


Figure (D.3): One-dimensional staggered grid, computational domain ranges from ½Δx to $L = M\Delta x$.

Analogous to section 2.5 a time splitting scheme is proposed. Due to nonlinearity the implicit parts need an iterative procedure. The iteration index will be denoted with a superscript $[q]$. The discretization of the continuity equation is taken from STELLING (1984). The implicit treatment of the continuity equation at step 1 is performed with a locally linear scheme derived from

$$\frac{\partial}{\partial x}(Hu) = \frac{\partial}{\partial x}(du) + \zeta\frac{\partial u}{\partial x} + u\frac{\partial \zeta}{\partial x} \tag{D.29}$$

The continuity equation will be evaluated at $x = m\Delta x$, while the momentum equation is evaluated at $x = (m+\frac{1}{2})\Delta x$. For the inner points, $m = 1,..,M-1$, the following procedure arises

<u>Step 1</u>, with $\zeta^{[0]} = \zeta^l$ and $u^{[0]} = u^l$, $q = 1,..,\mathrm{Q}$:

$$\left\{\begin{aligned} &\frac{\zeta_m^{[q]} - \zeta_m^l}{\frac{1}{2}\Delta t} + \frac{(du)_m^{[q]} - (du)_{m-1}^{[q]}}{\Delta x} + \zeta_m^{[q-1]}\frac{u_m^{[q]} - u_{m-1}^{[q]}}{\Delta x} \\ &\qquad + \frac{1}{2}\left[u_m^{[q-1]}\frac{\zeta_{m+1}^{[q]} - \zeta_m^{[q]}}{\Delta x} + u_{m-1}^{[q-1]}\frac{\zeta_m^{[q]} - \zeta_{m-1}^{[q]}}{\Delta x}\right] = 0 && \text{(D.30a)} \\ &\frac{u_m^{[q]} - u_m^l}{\frac{1}{2}\Delta t} + \mathrm{D}_1\left(u_m^l\,,\,u_m^{[q]}\right) + g\frac{\zeta_{m+1}^{[q]} - \zeta_m^{[q]}}{\Delta x} = 0 && \text{(D.30b)} \end{aligned}\right.$$

<u>Step 2</u>, with $\zeta^* = \zeta^{[\mathrm{Q}]}$ and $u^* = u^{[\mathrm{Q}]}$:

$$\left\{\begin{aligned} &\frac{\zeta_m^{l+1} - \zeta_m^*}{\frac{1}{2}\Delta t} + \frac{\left(\frac{1}{2}\left(\zeta_{m+1}^* + \zeta_m^*\right) + d_m\right)u_m^* - \left(\frac{1}{2}\left(\zeta_m^* + \zeta_{m-1}^*\right) + d_{m-1}\right)u_{m-1}^*}{\Delta x} = 0 && \text{(D.31a)} \\ &\frac{u_m^{l+1} - u_m^*}{\frac{1}{2}\Delta t} + \mathrm{D}_2\left(u_m^{l+1}\,,\,u_m^*\right) + g\frac{\zeta_{m+1}^* - \zeta_m^*}{\Delta x} = 0 && \text{(D.31b)} \end{aligned}\right.$$

Equations (D.30a) and (D.31a) are also valid for $m = 1$ by setting $\zeta_0 = \zeta_1$. The algebraic equations are completed with algebraic equations valid near the boundaries. At the inflow boundary, $m = 0$, equation (D.28) is discretized

<u>Step 1</u>, with $\zeta^{[0]} = \zeta^l$ and $u^{[0]} = u^l$, $q = 1,..,\mathrm{Q}$:

$$u_m^{[q]} + \epsilon\left(\frac{u_m^{[q]} - u_m^l}{\frac{1}{2}\Delta t} + \sqrt{\frac{g}{\zeta_{m+1}^l + d_m}}\;\frac{\zeta_{m+1}^{[q]} - \zeta_{m+1}^l}{\frac{1}{2}\Delta t}\right) = u^{\text{inflow}} \tag{D.32a}$$

Step 2, with $\zeta^* = \zeta^{[Q]}$ and $u^* = u^{[Q]}$:

$$u_m^{l+1} + \epsilon \left(\frac{u_m^{l+1} - u_m^*}{\frac{1}{2}\Delta t} + \sqrt{\frac{g}{\zeta_{m+1}^* + d_m}} \, \frac{\zeta_{m+1}^{l+1} - \zeta_{m+1}^*}{\frac{1}{2}\Delta t} \right) = u^{\text{inflow}} \qquad \text{(D.32b)}$$

At the outflow boundary, $m = M$, the water level is prescribed as in (D.27). For both steps

$$\zeta_m = \zeta^{\text{outflow}} \qquad \text{(D.33)}$$

The advection operators D_1 and D_2 arising in (D.30b) and (D.31b) complete the construction of a closed set of algebraic equations. From these equations the unknowns are solved individually or simultaneously. To avoid accumulation of rounding-off errors when solving the matrix system, the arising matrix should be diagonally dominant. This sufficient condition is realized if

$$\max_m \left| \frac{\Delta t}{\Delta x} u_m \right| \le 2 \qquad \text{(D.34)}$$

The threshold is sketched in figure (D.2). Other physical and numerical data are given by

Length of channel:	$L = 100\text{m}$
Acceleration due to gravity:	$g = 10\text{m/s}^2$
Inflow velocity at stationary state:	$u^{\text{inflow}} = 0.5\text{m/s}$
Fixed water level elevation at outflow boundary:	$\zeta^{\text{outflow}} = 0\text{m}$
Maximum depth:	$\max d = 10\text{m}$

Spatial increment (number of intervals):	$\Delta x = 4\text{m}$ $(M = 25)$
Time increment:	$\Delta t = 1.6\text{s}$
Reflection parameter:	$\epsilon = 600\text{s}$
Maximum number of iterations in step 1:	$Q = 5$

The numerical errors in the approximation of the stationary state water level elevation are plotted in figure (D.4), (D.5) and (D.6) classified to the advection schemes given above. The nonconservative schemes introduce an artificial head loss - see figure (D.4). This is not observed when dealing with conservative schemes - see figure (D.5). The observed head loss, $\Delta\zeta$, can be associated with the Chezy coefficient C (friction parameter)

$$g\frac{\partial \zeta}{\partial x} + \frac{g}{C^2}\frac{u|u|}{H} = 0 \quad \rightarrow \quad C \approx \sqrt{\frac{Lu|u|}{H|\Delta\zeta|}} \qquad \text{(D.35)}$$

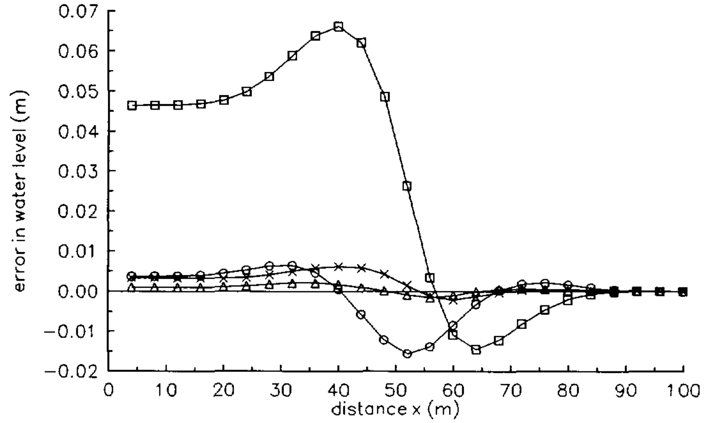


Figure (D.4): Numerical error in water level approximation: $\zeta_m - \zeta(m\Delta x)$.

Nonconservative advection schemes.

box: scheme 1 (1st order upwind), cross: scheme 3 (3rd order upwind/central),

circle: scheme 2 (2nd order upwind), triangle up: scheme 4 (4th order upwind/central).

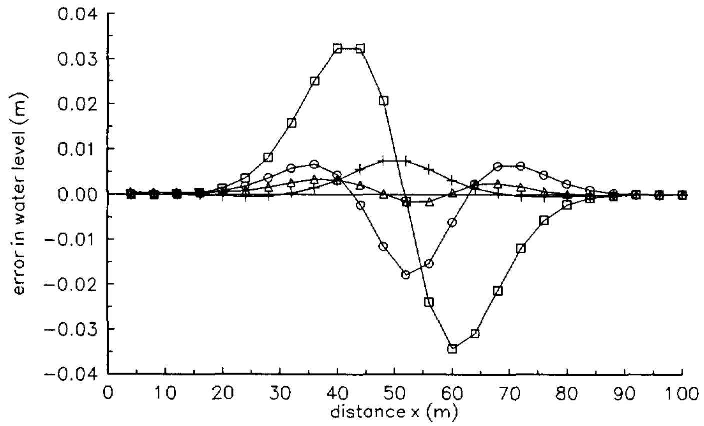


Figure (D.5): Numerical error in water level approximation: $\zeta_m - \zeta(m\Delta x)$.

Conservative advection schemes.

box: scheme 5 (1st order upwind), plus: scheme 7 (2nd order central),

circle: scheme 6 (2nd order upwind), triangle up: scheme 8 (4th order upwind/central).

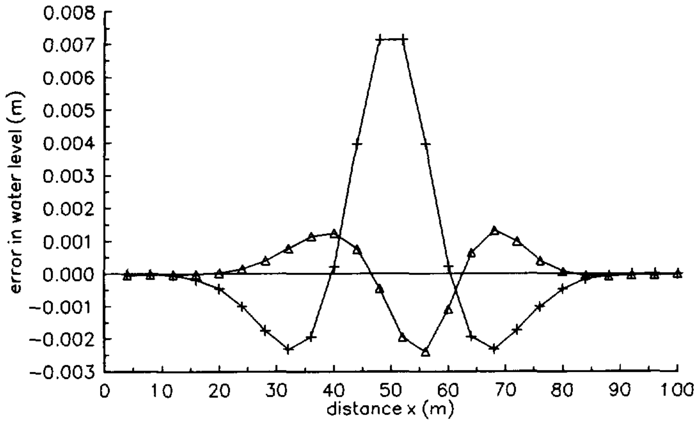


Figure (D.6): Numerical error in water level approximation: $\zeta_m - \zeta(m\Delta x)$.

Conservative advection schemes (continued) - discretization written in conservative form.

plus: scheme 9 (2nd order central),

triangle up: scheme 10 (4th order upwind/central).

For instance, the first order upwind scheme gives $C \approx 7.5\text{m}^{1/2}/\text{s}$ which would indicate a very large friction.

Scheme 9 (central differencing) and scheme 10 (fourth order central/upwind) give the smallest absolute errors - see figure (D.6). The central differencing scheme however is less dissipative and converges slowly to stationary state. Therefore scheme 10 is favourable. To gain confidence in the exact order of accuracy the spatial step has been halved. It turned out that the order of accuracy for scheme 10 can be estimated with 2.5 for the water level approximation and with 2.9 for the velocity distribution. The name fourth order central/upwind scheme is slightly overdone.

Appendix E: Reflection principles at closed boundaries

A particle that contacts a closed boundary is reflected to exclude transfer of mass through such a boundary. A traditional reflection procedure applies perpendicular reflection. In section 4.3.6 an alternative reflection principle is suggested. It accounts for the principal directions of the diffusion tensor. Reflection procedures will be described in detail. An illustration is given by studying the asymptotic behaviour of a stationary uniform concentration distribution in a two-dimensional reservoir.

It is assumed that the coordinate axes coincide with the principal directions of the diffusion tensor. Principal values of the diffusion tensor are denoted by D_I and D_{II}. For diffusion coefficients that are constant in space, a random walk that simulates the diffusion process is given by

$$\begin{cases} \Delta X = \sqrt{6D_I\,\Delta t}\; R_1 \\ \Delta Y = \sqrt{6D_{II}\Delta t}\; R_2 \end{cases} \tag{E.1}$$

with R_1 and R_2 uniform random numbers in $(-1,1)$ and Δt the time step. In the one-dimensional case, a particle that suffers reflection while it is moving in x-direction, is retraced towards its starting point. This procedure implies $\partial C/\partial x|_{\text{boundary}} = 0$ - see CHANDRASEKHAR (1943). Perpendicular reflection in two or three dimensions implies:

$$\frac{\partial C}{\partial n} = \nabla C \cdot \underline{n} = 0 \tag{E.2}$$

with $\underline{n}$ the vector normal to the boundary. To guarantee a zero flux boundary condition:

$$0 = \underline{q} \cdot \underline{n} = D \nabla C \cdot \underline{n} = \nabla C \cdot D^T \underline{n} \tag{E.3}$$

Condition (E.3) leads to a reflection principle illustrated in figure (E.1b); a reflected particle is on the line through $\underline{X}(\Delta t)$ with directional vector $D^T\underline{n}$. In case of perpendicular reflection directional vector $\underline{n}$ is used - see figure (E.1a).

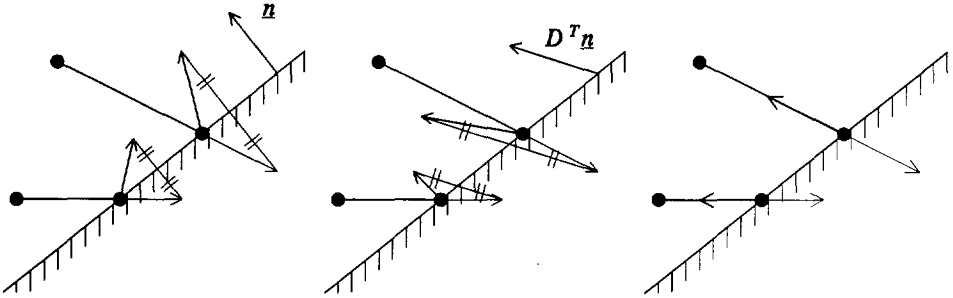


Figure (E.1): Reflection principles.

E.1a (LEFT): perpendicular reflection,
E.1b (CENTRE): reflection in direction $D^T\underline{n}$,
E.1c (RIGHT): reflection in direction of starting point.

- Application of reflection principles in a (two-dimensional) uniform random walk

Consider a particle that starts in $(0,0)$ at time $t_0 = 0$. Away from closed boundaries, the step density function $\bar{f}(x,y\,|\,\xi,\eta)$ for a particle that is observed at (x,y) started at (ξ,η), is a uniform distribution over an area $\left(\xi - \sqrt{6D_I\Delta t}\ ,\ \xi + \sqrt{6D_I\Delta t}\right) \times \left(\eta - \sqrt{6D_{II}\Delta t}\ ,\ \eta + \sqrt{6D_{II}\Delta t}\right)$. Near closed boundaries the step density function is ruled by the applied reflection procedure - see figure (E.2). After N steps, the density function of the particle position is obtained from:

$$\begin{cases} f^1(x,y) & = \bar{f}(x,y\,|\,0,0) \\ f^{n+1}(x,y) & = \displaystyle\iint_{\mathbb{R}^2} f^n(\xi,\eta)\,\bar{f}(x,y\,|\,\xi,\eta)\; d\xi d\eta \end{cases} \tag{E.4}$$

The reflection principle of figure (E.1b) implies that $f^N(x,y)$ converges, as N tends to infinity, to the solution, $C(x,y,t)$, of (providing $N\Delta t = t$)

$$\begin{array}{ll} \dfrac{\partial C}{\partial t} = D_I \dfrac{\partial^2 C}{\partial x^2} + D_{II} \dfrac{\partial^2 C}{\partial y^2} & \\ D\nabla C \cdot \underline{n} = 0 & \text{boundary condition (zero flux)} \\ C(x,y,0) = \delta(x,y) & \text{initial condition (Dirac-delta function)} \end{array} \tag{E.5}$$

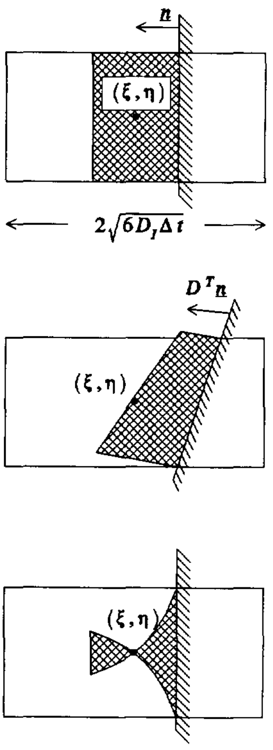
- Uniform concentration distribution in a closed reservoir

A necessary requirement of reflection principles applied to diffusion processes is that a uniform concentration distribution remains uniform as time evolves. To compute time-dependent concentration values, a two-dimensional reservoir is covered with a 10 × 10 rectangular grid. The asymptotic behaviour is quantified by the error measure

$$\epsilon^l = \frac{\displaystyle\sum_{m=1}^{10}\sum_{n=1}^{10}\left(C^l_{m,n} - C^0\right)^2}{\displaystyle\sum_{m=1}^{10}\sum_{n=1}^{10}\left(\bar{C}^0_{m,n} - C^0\right)^2} \tag{E.6}$$

C^0 represents the uniform initial concentration value. $\bar{C}^0_{m,n}$ and $C^l_{m,n}$ denote the representation of the initial condition and the simulated cell-averaged concentration at time $t = t_l$ respectively. Both values refer to cell (m,n). The correct asymptotic behaviour is obtained if C^l approximates C^0 as accurate as $\bar{C}^0$, i.e.,

$$\epsilon^l = O(1) \qquad \forall\ l \geq 1 \tag{E.7}$$



Ω: computational domain (e.g., inside reservoir)

$V_{(\xi,\eta)}$: $\left(\xi-\sqrt{6D_I\Delta t}\,,\ \xi+\sqrt{6D_I\Delta t}\right)\times\left(\eta-\sqrt{6D_{II}\Delta t}\,,\ \eta+\sqrt{6D_{II}\Delta t}\right)$

The position of a particle after reflection is given by

$$\varphi : V_{(\xi,\eta)}\backslash\Omega \to \Omega : (x,y) \mapsto \left(\varphi_x(x,y), \varphi_y(x,y)\right)$$

The step density function (transition probability density function) is then determined by

$$\bar{f}(x,y\,|\,\xi,\eta) = \frac{1\ \chi_{\{(x,y)\in V_{(\xi,\eta)}\cap\Omega\}} + |J|\ \chi_{\{(x,y)\in\Omega,\ \varphi^{-1}(x,y)\in V_{(\xi,\eta)}\backslash\Omega\}}}{4\sqrt{6D_I\Delta t}\sqrt{6D_{II}\Delta t}}$$

with χ the indicator function - see (4.16) of section 4.2.2, and

$$|J| = \left|\frac{\partial\varphi_x}{\partial x}\frac{\partial\varphi_y}{\partial y} - \frac{\partial\varphi_x}{\partial y}\frac{\partial\varphi_y}{\partial x}\right|$$

In (E.2a) and (E.2b): $|J| = 1$

In (E.2c): $|J|$ is singular at $(x,y) = (\xi,\eta)$

Figure (E.2): Step density function near a closed boundary. Shaded area denotes $\varphi\left(V_{(\xi,\eta)}\backslash\Omega\right)$.

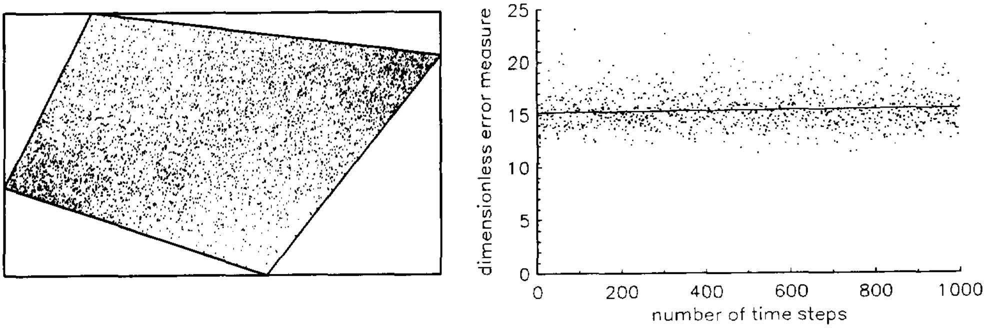
E.2a (TOP): perpendicular reflection,

E.2b (CENTRE): reflection in direction $D^T\underline{n}$,

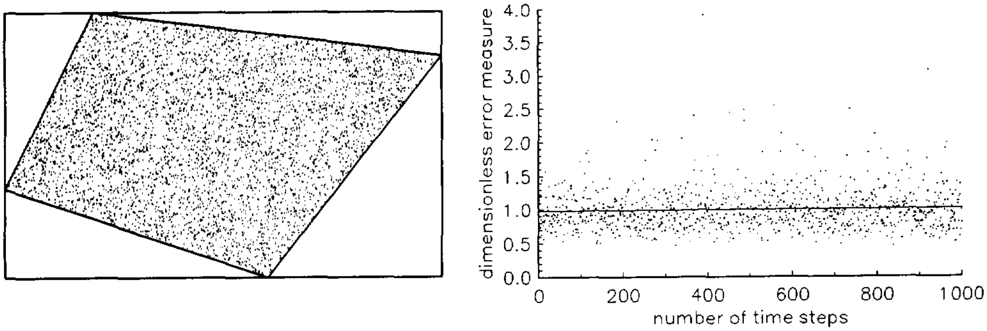
E.2c (BOTTOM): reflection in direction of starting point.

If $D_I \neq D_{II}$ perpendicular reflection fails, while the procedure suggested above is adequate - see figure (E.3). In addition to these two reflection principles, another reflection principle is introduced which simply retraces the particle towards its starting point - see figure (E.1c). Although this procedure yields the correct asymptotic behaviour - see figure (E.3c), it is not suitable for general simulation purposes, since the step density function - see figure (E.2c), is such that f^N, the solution of (E.4), does not converge to the solution of (E.5) ($N\Delta t = t$, $D_I \neq 0$, $D_{II} \neq 0$).

Particle suffers perpendicular reflection



Particle suffers reflection using $D^T\underline{n}$



Particle retraces towards starting point

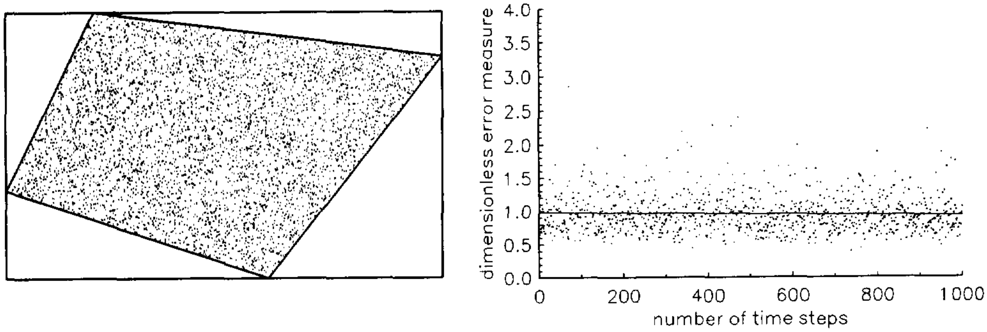


Figure (E.3): LEFT: asymptotic particle distribution in two-dimensional reservoir,
RIGHT: error measure e^l as a function of time steps (plots include regression line).

Physical and numerical data: D_I = 2.0 m^2/s, D_{II} = 0.1 m^2/s, ANISOTROPY
Δt = 1.0 s, T = 1000 s, n_p = 4900.

Dankwoord

Dit proefschrift zou onrecht aangedaan worden als ik het resultaat van deze zo gedenkwaardige periode niet zou afsluiten met een dankwoord. Op het gevaar af ironisch over te komen, wil ik u, lezer, als eerste bedanken voor het bereiken van pagina 229.

Het verrichten en voltooien van een promotie-onderzoek vereist financiële ondersteuning, (wetenschappelijke) begeleiding, vertrouwen, (emotionele) steun en voldoende ontspanning.

Ik dank dan ook de commissie Beek voor de toekenning van dit onderzoek, de Technische Universiteit Delft en de Lelystichting voor financiële ondersteuning en Integraal Waterbeheer voor de bijdrage bij het tot stand komen van dit manuscript.

Vertrouwen en begeleiding heb ik mogen genieten van mijn promotor Guus Stelling en al mijn collega's van de sectie vloeistofmechanica. Aanvullende steun werd geboden door de leden van de gebruikerscommissie numerieke stromingen. Het enthousiaste meedenken van Henk van den Boogaard, werkzaam bij het Waterloopkundig Laboratorium, wil ik zeker niet onvermeld laten. Het is overbodig op te merken dat zonder jullie steun dit manuscript geen proefschrift had mogen heten.

De noodzakelijke ontspanning werd mij aangereikt door vele vrienden zowel op als naast de werkvloer. Ik heb veel genoten, met name tijdens biljarten, uitjes, discussies, badminton, reizen, koffiepauzes, etentjes, spelletjes, concerten, etcetera. Ik weet dat onze vriendschap geen speciale bevestiging nodig heeft. Een uitzondering maak ik voor mijn ouders en mijn broer die zeker een blijvende plaats innemen binnen mijn vriendenkring. Ik heb mij altijd gerealizeerd dat ik in het bijzonder met jullie steun mag welvaren.

Curriculum Vitae

Daan Dunsbergen werd geboren op 16 augustus 1963 te Waalwijk. Na het behalen van het VWO-Atheneum-B diploma begon hij een wiskundestudie aan de Rijksuniversiteit Utrecht. Het kandidaatsexamen met bijvak econometrie werd afgelegd op 2 juli 1984. De daaropvolgende doctoraalstudie omvatte hoofdvakken in de toegepaste wiskunde. Op 31 augustus 1987 studeerde hij af bij prof.dr.ir. W. Eckhaus die hem begeleidde bij het onderzoek naar de toepassing van singuliere storingsrekening in het oceaanmodel van Munk. Tevens werd een bevoegdheid behaald tot het geven van onderwijs in wiskunde.

In de periode 1988-1989 vervulde hij zijn militaire dienstplicht. Hij was aangesteld als pelotonscommandant bij een bewakingseenheid (regiment van Heutsz). Hij was verantwoordelijk voor de uitvoering van diverse vredestaken van het peloton.

Van 1 september 1989 tot 1 maart 1994 was hij als assistent in opleiding werkzaam bij de sectie Vloeistofmechanica, vakgroep Waterbouwkunde, faculteit Civiele Techniek aan de Technische Universiteit Delft. Het grootste deel van de onderzoeksresultaten is beschreven in het onderhavige proefschrift. Na de beëindiging van het formele dienstverband werd het promotie-onderzoek afgerond.

Op 1 oktober 1994 start hij als projectmedewerker aan het Rijksinstituut voor Kust en Zee /RIKZ, alwaar hij statistisch golfonderzoek gaat verrichten voor het bepalen van (hydraulische) randvoorwaarden.