

A Finite Volume Solver
for the Simulation of
Transport Processes

Fahmi Naifar

A Finite Volume Solver for the Simulation of Transport Processes

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J. T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op dinsdag 24 januari 2006 om 10.30 uur

door

Fahmi Naifar

Master of Science in Hydrology and Water Resources
International Institute for Infrastructural, Hydraulic
and Environmental Engineering,
Delft, The Netherlands.

geboren te Sfax, Tunesië.

Dit proefschrift is goedgekeurd door de promotors:

Prof. dr. ir. A. W. Heemink
Prof. dr. ir. G. S. Stelling

Toegevoegd promotor:

Dr. P. Wilders

Samenstelling promotiecommissie:

Rector magnificus,	voorzitter.
Prof. dr. ir. A. W. Heemink,	Technische Universiteit Delft, promotor.
Prof. dr. ir. G. S. Stelling,	Technische Universiteit Delft, promotor.
Dr. P. Wilders,	Technische Universiteit Delft, toegevoegd promotor.
Prof. dr. ir. K. Maalel,	Ecole Nationale d'Ingénieurs de Tunis.
Prof. dr. ir. P. Wesseling,	Technische Universiteit Delft.
Prof. dr. ir. A.E. Mynett,	Technische Universiteit Delft/UNESCO-IHE.
Prof. dr. J.G. Verwer,	Universiteit van Amsterdam.

Naifar, Fahmi.

A Finite Volume Solver for the Simulation of Transport Processes /
PhD Thesis Delft University of Technology.
With summary in Dutch.

ISBN 90-8559-142-2

Copyright ©2006 F. Naifar

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 written permission from the copyright holder.

Printed by [OPTIMA] Grafische Communicatie, Rotterdam

To my family

Acknowledgments

This thesis contains the results of my research performed at the Delft Institute of Applied Mathematics (DIAM) of Delft University of Technology. I would like to express my sincere thanks to all those who have contributed to the completion of this work.

I am very grateful to my three supervisors:

- First, I would like to thank my daily supervisor Dr. Peter Wilders. I am grateful to, and gratified by, his attentiveness, guidance, remarks and discussions which brought this thesis to reality. I also would like to thank him for his careful reading of this manuscript and for his suggestions for improvement.
- I am also grateful to my promoter, Prof. Dr. Ir. Arnold Heemink for his wise support, for his help and for providing such a good working atmosphere within his research group.
- A special note of appreciation is due to my second promoter, Prof. Dr. Ir. G.S. Stelling. His suggestions and advises were always of great value during the whole period of the project.

I would like to thank Drs M.J. Nieman, editor of English, for her careful review of my thesis, while she was in a busy period of her work.

Because of Kees Lemmens and Eef Hartman, I became addicted to Linux OS. Nevertheless, I deeply thank both of them for introducing me to this wonderful world. Linux was indeed an essential tool to carry out the present work. See page 93.

I would like to acknowledge the National Institute for Coastal and Marine Management in the Netherlands (RIKZ) for allowing to use their software package SIMONA and associated data.

I have a long list of friends which I need to thank. I am not going to present it, but I kindly ask each member of this list to find here the expression of gratitude for the nice time I spent in the Netherlands.

Finally, I thank my mother, my father, my wife, and my children for their patience during my stay in Delft. I also thank the rest of family for their support.

Contents

1	Introduction	1
1.1	Tout à la mer	1
1.2	Object of the thesis	1
1.3	Outline of the thesis	3
2	An Eulerian 2-D finite volume solver for transport processes	5
2.1	Introduction	5
2.2	Governing equation	6
2.3	Flow computation	7
2.3.1	Shallow water equations	7
2.3.2	Flow solver	8
2.4	Solution procedure for the transport problem	9
2.4.1	Unstructured grid	10
2.4.2	Finite volume formulation	11
2.4.3	Approximation of the advective flux	12
2.4.4	Boundary treatment of the advective flux	13
2.4.5	Approximation of the diffusive flux	14
2.4.6	Boundary treatment for the diffusive flux	15
2.4.7	Time integration	15
2.5	General layout of the model	16
2.6	Data structure	18
2.7	Positivity of the 2-D transport solver in real-life applications	20
2.7.1	Positive advection	20
2.7.2	Constancy condition and data preprocessing	23
2.7.3	Conservative flow computations	24
2.7.4	Consistency	24
2.7.5	Dry-wet procedure	25
2.8	Improvement of the computational efficiency in the 2-D case	28
2.8.1	Time integrated flow data	29
2.8.2	Periodic continuation	30
2.9	Conclusions	31

3	Numerical Results of the 2-D solver	35
3.1	Introduction	35
3.2	Real-life application	35
3.3	Time integrated flow data	40
3.4	Periodic continuation	44
3.5	Combination of periodic continuation and time integrated flow data .	45
3.6	Conclusions	46
4	Development of a 3-D transport solver	49
4.1	Introduction	49
4.2	Governing equation	50
4.3	3-D grid	50
4.4	Layer-averaged transport equation	51
4.5	3-D finite volume formulation	54
4.6	Flux approximation	56
4.6.1	Available data from TRIWAQ	56
4.6.2	Horizontal advection flux	56
4.6.3	Vertical advection term	57
4.6.4	Horizontal diffusion flux	58
4.6.5	Vertical diffusive flux	58
4.6.6	Remaining terms	59
4.7	The semi discrete system	61
4.8	Time integration	62
4.9	General layout of the model	63
4.10	Data structure	63
4.10.1	Numbering of grid objects	64
4.10.2	Array indexing in 3-D	64
4.10.3	Matrix storage	64
4.11	Boundary conditions	65
4.12	Conclusions	65
5	Numerical results of the 3-D solver	69
5.1	Introduction	69
5.2	Flow data preprocessing	69
5.3	Test case	71
5.3.1	Grid	71
5.3.2	Hydrodynamic coefficients	72
5.3.3	Numerical results for the test case	72
5.4	Application to the Dutch coast	93
5.4.1	The grid	93
5.4.2	Hydrodynamic coefficients	93
5.4.3	Check for the constancy condition	93
5.4.4	Real-life simulation	93
5.5	Conclusions	102

6	General conclusions and recommendations	103
6.1	General conclusion	103
6.2	Recommendations	106
	Bibliography	107
A	An algebraic turbulence model for the vertical diffusion	113
	Summary	115
	Samenvatting	117
	Curriculum Vitae	119

List of Figures

2.1	Water depth parameters	6
2.2	Water depth parameters	8
2.3	Velocities and water depths at the cell faces	9
2.4	Molecule used for the evaluation of the advective flux	13
2.5	Molecule used for the evaluation of the diffusive flux	14
2.6	General layout of the solver algorithm	17
2.7	Terminology of the grid objects	18
2.8	Terminology of the grid objects	19
2.9	Cells' numbering for the triangular grid by Wilders and Fotia.	21
2.10	Staggered grid in WAQUA	26
2.11	Dry-wet neighbors	28
2.12	Total water depth at cell 10000	31
2.13	Water level elevation at some randomly chosen cells.	32
3.1	Grid of Kuststrookmodel	36
3.2	Difference between \tilde{H}_{cell} and H_{cell}	37
3.3	Constant solution	38
3.4	Spreading of the tracer ($\tau = 1min$)	39
3.5	Lowest concentration ($\tau = 1 min$)	40
3.6	Situation after 62.5 days (WAQUA simulation, $\tau=1 min$)	41
3.7	Situation after 62.5 days (continuous simulation $\tau = 15 min$)	42
3.8	Courant number	43
3.9	Situation after 62.5 days ($\tau = 1 min$)	44
3.10	Lowest concentration ($\tau = 1 min$)	45
3.11	Situation after 62.5 days ($\tau = 15 min$)	46
3.12	Simulations using different tidal period lengths.	48
4.1	3-D grid	51
4.2	Layer depth and layer elevation	51
4.3	Definition of grid elements	52
4.4	Location of the calculation points of different variables in a 3-D grid volume.	57
4.5	Variables used for the computation of the vertical diffusive flux	59
4.6	Convention for the notation of the concentration at different locations	60
4.7	Shadow volume for the computation of $[\nabla c \nabla z]_{z_k}^{z_{k-1}}$	61

4.8	3-D support molecule	62
4.9	General layout of the solver algorithm	67
4.10	Correspondence between one- and two-dimensional array indexing	68
4.11	Structure of arrays <i>ja</i> and <i>ar</i> in the 3-D case	68
5.1	Grid of the test case	71
5.2	Water level at the downstream boundary	72
5.3	Velocity field at the topmost 5 layers at time 30 minutes	74
5.4	Constant solution after 40 min	75
5.5	Layer depth and layer elevation	76
5.6	Top view of the concentration at different layers after 16 min	78
5.7	3-D representation of the iso-value 0.25 after 16 min viewed from different angles	79
5.8	Total mass in the computational domain	80
5.9	2-D simulation of the test case. Situation after 16 min.	80
5.10	TRIWAQ simulation of the concentration at the different layers	81
5.11	Top view of the concentration at different layers after 16 min	84
5.12	Time history of the concentration at cell 170	85
5.13	Time history plots at the cross-section formed by cells 168 to 173	86
5.14	Time history plots at the cross-section formed by cells 904 to 912	87
5.15	Time history plots at the cross-section formed by cells 1624 to 1633	88
5.16	Spreading of the pollutant resulting from a 2-D simulation	89
5.17	Parabolic vertical diffusion profiles.	89
5.18	Top view of the concentration at different layers after 16 min using parabolic vertical diffusion (profile I).	90
5.19	Top view of the concentration at different layers after 16 min using parabolic vertical diffusion (profile II).	91
5.20	Top view of the concentration at different layers after 16 min using the algebraic model for the vertical diffusion.	92
5.21	Spreading pattern after one month from the start of the release	95
5.22	Spreading pattern after two months from the start of the release	96
5.23	Result of a tracer release simulation (after 63 days)	97
5.24	Result of a tracer release simulation (after 63 days)	98
5.25	Location of time history graphs	99
5.26	Result of a tracer release simulation (after 63 days) using an algebraic model for the vertical diffusion	100
5.27	Total mass in the domain	101
5.28	Maximum and minimum computed concentrations.	101
A.1	Reconstruction of the velocity vector at the cell center.	114

List of Tables

2.1	Grid objects: problem variables association	18
3.1	Comparison between the results of WAQUA and the reference solution at time step 90000.	40
3.2	Comparison of the results of the time integrated flow data.	42
3.3	Results using periodic continuation and time integrated flow data	45
3.4	Comparative table of different tidal period lengths	47

Chapter 1

Introduction

1.1 Tout à la mer

It is very clear that strategies based upon the ancient idea of “*tout à la mer*”, literally meaning everything to the sea, should not be tolerated anymore. These strategies, which consider water courses, rivers, lakes, seas and oceans as final sinks able to absorb all of our wastes, can be disastrous, and the resulting threats for all forms of life are simply enormous. The list of negative impacts is very long, ranging from direct consequences on human health of poor-quality drinking water to indirect effects resulting from the systematic poisoning of the aquatic life.

Due to the increasing importance of this issue, new fields have been established such as environmental hydraulics and water quality modeling with one major goal: to provide a better understanding of the fate of materials disposed in water bodies. These materials may undergo two major processes: transformation and transport. A lot of work has been and continues to be done on the transformation process in a large diversity of fields, namely those of chemistry, physics and biology. Transport, however, remains a crucial component in this kind of environmental studies.

1.2 Object of the thesis

In the present thesis, we intend to provide more insight in the transport into water bodies. We are particularly interested in the transport of dissolved materials which do not undergo any kind of transformation. If, in the future, transformation needs to be incorporated into a full and more global model, this can be reasonably easily achieved through model coupling. Here, we focus on pure transport.

The close relationship between the water flow and the transport of material is well known. In fact, any fundamental study of water quality requires knowledge of the pathway, volume and velocity of the water. In practical terms, the first step in any water quality modeling study is to determine “where the water goes” and how water movement affects the concentration of dissolved and suspended materials [28].

As in most academic fields, modeling is the key tool to be used for prediction. Thanks to the advances made in the field of hydrodynamics in the last decades and the availability of modern computers, numerical simulation of transport processes is becoming feasible. Flow and transport are in fact moving forward from the old combination of measurement and description towards a new era of numerical transport modeling [28].

The key objective of this thesis is the development of an innovative Eulerian two-dimensional and three-dimensional transport model able to predict transport routes in the coastal environments. This is a challenging and important research subject, relevant to a diversity of fields such as civil and environmental engineering, fisheries and oceanography. One main difficulty in this type of modeling resides in the fact that coastal zones usually present complex geometries leading to complex flows with highly varying velocities and water depths.

From a mathematical point of view, the transport of a dissolved tracer is governed by the advection-diffusion equation. This equation incorporates the two mechanisms through which a tracer is transported through a water body. The first is advection, defined as the transport due to the bulk movement of water. The second is dispersion which is the spreading resulting from small local variation of the velocity field. Molecular diffusion, which represents the spreading at the molecular level, is in general much less significant than dispersion and is often considered simply as a contributing process to the dispersive movement. Although the terminology advection-dispersion equation can be found in the literature, here we use the term advection-diffusion since the mathematics of diffusion is applied.

While the three-dimensional governing equation has been known since the 18th century, only relatively recently has the introduction of computers and numerical solutions made general solutions possible [28]. Considerable progress has been made in this context, and nowadays, modern methods are subject to constantly increasing requirements, namely:

- the possibility of representing spatial domains with complex geometries,
- the use of large computational time steps,
- robustness and positivity, and
- sufficient accuracy.

Among the latest advances in the domain of Eulerian tracer transport computation, we denote the works of Colombini and Stocchino [8] who proposed a 3-D finite difference model based on the Quickest method to solve the transport equation. Liang et al. in [24] used a collocated finite volume method to solve the coupled flow-transport equations using a MUSCL-Hancock scheme on a structured grid. In the field of tracer transport, unstructured grids and in particular triangular grids were notably used in the works of Lin et al. [26], Berzins [4], Durlofsky [16, 17] and Mazzia et al. [30]. Basically, these are developments to the early works of Venkatakrishnan and Barth who were perhaps the firsts who extended one-dimensional schemes to triangles in the field of airfoil calculations [48].

In this thesis, we intend to bring further new developments in the field of uncoupled transport computation. In particular, we solve the advection-diffusion equation

using a second order accurate cell centered finite volume method on an unstructured quadrilateral grid. This is a rather new type of applications that we want to investigate in order to achieve the objectives mentioned above. Moreover, we aim to state a clear definition of the conditions necessary to fulfill in order to obtain positive and conservative results, which is a very important aspect in any transport simulation. New techniques to improve the computational aspect of the uncoupled flow-transport models are investigated. One of these techniques consists of using time integrated flow data which can be an efficient tool to speed up the computations and reduce the required storage.

The Dutch coastal region constitutes the main real-life application of the model. In this area, the North Sea receives pollution released from rivers such as Rhine, Meuse, Scheldt, Eems and Thames, of which the major contributor is river Rhine [12]. A large number of studies has been conducted on pollution in this part of the world; we only mention as examples [12], [43], [40]. By providing a tool for gaining more insight in the transport of pollution in coastal areas, we hope to contribute to solving the pollution problem.

In general, the model built is meant to be an efficient tool for determining the spatial and temporal spreading of released tracers. Knowing the critical levels of pollution, it can assist in developing and designing measures against pollution with as much accuracy as possible in order to minimize the costs when attempting to reduce pollution. This is very desirable since pollution reduction may involve high financial costs, especially for large scale systems. Other possible uses are for instance to study remediation options for polluted bays and estuaries, to predict the impact of possible environmental incidents on fisheries and marine wild life or to model salinity intrusion effects.

1.3 Outline of the thesis

Chapter 2 of this thesis starts by presenting the 2-D transport solver and its underlying governing equation and numerical solution method. Particular emphasis is put upon measures to be taken in order to ensure positivity of the solution for real life applications. This chapter also presents two techniques that can be used to improve the computational aspect, namely the use of time integrated flow data and the use of periodic continuation. These techniques can present considerable advantages when the flow computations and transport computations are decoupled as in our case. With the time integrated flow data, the use of different time steps in the transport solver and in the flow solver becomes possible, which allows in particular to use larger time steps for the transport computations. It allows also to use flow data sets obtained from third parties without being restricted to use the same time step. For the periodic continuation, the major advantage may consist in a large reduction of the required data storage space for the flow parameters.

Chapter 3 presents and discusses the numerical results of the 2-D model. The Dutch coastal area, described in this chapter, was selected for the numerical application of the model and the major simulated scenario consists of a pollutant release close to famous harbor of Rotterdam. The solver allows to describe the transport

process and to determine when the fragile ecosystem of the northern Dutch coast becomes threatened. Based on this numerical application, the proposed techniques to improve the computational aspect are evaluated.

The approach adopted to build the 3-D version of the transport solver is presented in chapter 4. Particular attention is given to the computation of the different fluxes of the finite volume method adopted. Also, techniques to optimize the data structure at different stages of the computation are presented. This issue of optimizing the data structure is of course important because of the large amount of data present in a 3-D computation.

Chapter 5 is dedicated to the numerical application of the 3-D solver. A hypothetical as well as a real-life case are considered; the latter is also related to the Dutch coast. For those special cases, the analysis of the obtained results allows to investigate several aspects notably related to the importance of the vertical diffusion coefficient or related to the question whether a 3-D model is required for an accurate simulation or a 2-D assumption suffices.

Finally, chapter 6 contains the general conclusions of this thesis and recommendations for future related implementations.

Chapter 2

An Eulerian 2-D finite volume solver for transport processes

2.1 Introduction

The forerunner of this project was first conceived in the early nineties in response to the constantly increasing importance of numerical modeling in the environmental domain. The interest was particularly in building a modern and powerful solver able to simulate the transport processes of dissolved pollutants.

There, a two-dimensional approach was adopted together with an unstructured triangular mesh grid. The finite volume method, which was gaining in interest mainly thanks to its robustness properties, was chosen to solve the numerical problem. Results related to this work have been published, notably in [53] and [54]. The original work on the solver was followed by efforts to add further functionalities. One of the objectives was the use of hybrid grids, that is, a grid mesh not limited to triangular cells but using mixed types of cells, which is described in details in [46] and [45].

Before starting the contributions made within the present thesis to this project, this chapter gives an introductory description of different features of the 2-D transport solver. After an introduction of the governing equations, special attention is given to the computation of the flow, which provides the parameters required for the transport problem. Then, the finite volume technique adopted is presented with emphasis on the flux computation. The general computing algorithm is described as well as some aspects related to the data structure. Then follows a section on measures to be taken in order to ensure positivity of the computed concentrations. In fact, for some real-life applications, it was found that negative values can still be obtained if arbitrary hydrodynamic parameters are used. One of the main reasons behind this problem is the lack of consistency between the flow numerical scheme and the transport numerical scheme. Negative values can also appear when the flow calculations are not completely conservative or also when drying and wetting takes

place in some parts of the domain. This section brings about clearly defined pre-processing steps required to preserve the conservation properties and to avoid the appearance of negative concentrations.

The last section of this chapter presents some techniques to reduce the amount of data to be stored. Uncoupled flow and transport computations need a lot of disk space for which reason, techniques allowing the reduction of the storage are required.

2.2 Governing equation

In case of negligible density changes, the concentration of a single phase miscible tracer obeys the general three-dimensional advection-diffusion equation (or transport equation)

$$\frac{\partial c}{\partial t} + \frac{\partial uc}{\partial x} + \frac{\partial vc}{\partial y} + \frac{\partial wc}{\partial z} = \frac{\partial}{\partial x}(D_h \frac{\partial c}{\partial x}) + \frac{\partial}{\partial y}(D_h \frac{\partial c}{\partial y}) + \frac{\partial}{\partial z}(D_v \frac{\partial c}{\partial z}) + S, \quad (2.1)$$

where $c(x, y, z, t)$ is the concentration of the tracer, (u, v, w) is the velocity vector, and S is the external source term. We assume that the diffusion tensor reduces to two scalar parameters, D_h and D_v , representing the diffusion coefficients in the horizontal and vertical directions, respectively.

In case of a 2-D approach, equation (2.1) can be integrated over the total water depth to obtain a depth averaged equation. Let \bar{c} be the average value of the concentration over the depth

$$\bar{c} = \frac{1}{H} \int_{-h}^{\eta} c \, dz, \quad (2.2)$$

where h and η are the level of the water surface and that of the bed of the water body with respect to a certain reference plane, respectively, as presented in figure 2.1. $H = h + \eta$ is the total water depth.

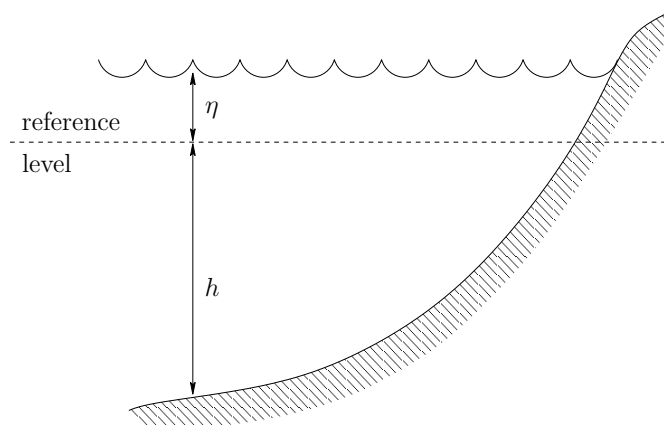


Figure 2.1: Water depth parameters

Let \mathbf{u} represent the depth-averaged velocity vector. The depth averaged transport equation can then be written under the so-called 2-DH formulation,

$$\frac{\partial H\bar{c}}{\partial t} + \nabla \cdot (H \mathbf{u} \bar{c}) = \nabla (H D_h \nabla \bar{c}) + H S. \quad (2.3)$$

This partial differential equation is a conservation law, the coefficients of which are functions of the flow parameters. These are obtained a priori using a separate model which solves the shallow water equations. The next section is related to this issue and briefly describes presentation of the flow computation.

2.3 Flow computation

Transport and flow are often solved separately which means that the transport is computed with an a priori known velocity field for the whole domain and the whole simulation period. The choice for the uncoupled approach presents a number of advantages. One major advantage resides in the possibility of using different time steps for the flow and for the transport. In fact, these two phenomena have a large difference in time scale between them and the coupling between the two will hamper the use of large time step for the transport computation. Another advantage consists in the possibility of using a single flow data set to simulate several transport scenarios without having to repeat the same flow computation every time. Finally, uncoupling may allow the use of flow data sets obtained from third parties but this is not always possible, as will be concluded later in this chapter.

2.3.1 Shallow water equations

Shallow water equations are commonly used to model the oceanographic circulation and the tidal fluctuations. These depth-averaged (or 2-DH) equations basically describe the conservation of mass and momentum of water in movement.

$$\begin{aligned} \frac{\partial H u}{\partial t} + \frac{\partial H u^2}{\partial x} + \frac{\partial H u v}{\partial y} - f H v + g H \frac{\partial \eta}{\partial x} &= \tau_u^w - \tau_u^b \\ \frac{\partial H v}{\partial t} + \frac{\partial H u v}{\partial x} + \frac{\partial H v^2}{\partial y} + f H u + g H \frac{\partial \eta}{\partial y} &= \tau_v^w - \tau_v^b \\ \frac{\partial \eta}{\partial t} + \frac{\partial H u}{\partial x} + \frac{\partial H v}{\partial y} &= 0, \end{aligned} \quad (2.4)$$

Here, τ_u^w and τ_v^w are the wind stress components, τ_u^b and τ_v^b are the bottom stress components, f is the Coriolis parameter, and g is the gravitational constant. The unknown u and v are now the depth-averaged velocities. An extended discussion on the shallow water equations and the mathematical modeling of flows can be found for example in [51]. Basically, the last equation of system (2.4), the mass conservation equation, is of interest for our transport computations since the conservation of the transported substance is closely related to that of water.

2.3.2 Flow solver

For this thesis, the software package WAQUA of the Rijkswaterstaat has been used to carry out the flow computations. Rijkswaterstaat, or RWS, is the Directorate-General for Public Works and Water Management in the Netherlands. WAQUA is also one of the oldest and most commonly used computation models at the National Institute for Coastal and Marine Management (RIKZ).

WAQUA is a two dimensional hydrodynamical model which computes water levels and currents in open water. Physical domains are represented in WAQUA either by rectilinear, curvilinear or spherical structured grids. The grids must be orthogonal or nearly orthogonal. The staggered grid which forms the basis of WAQUA implies that a modeled system can be regarded as consisting of a large number of linked, column-shaped volumes of water. Figure 2.2 reproduced from [19] gives an impression of the staggered grid principle.

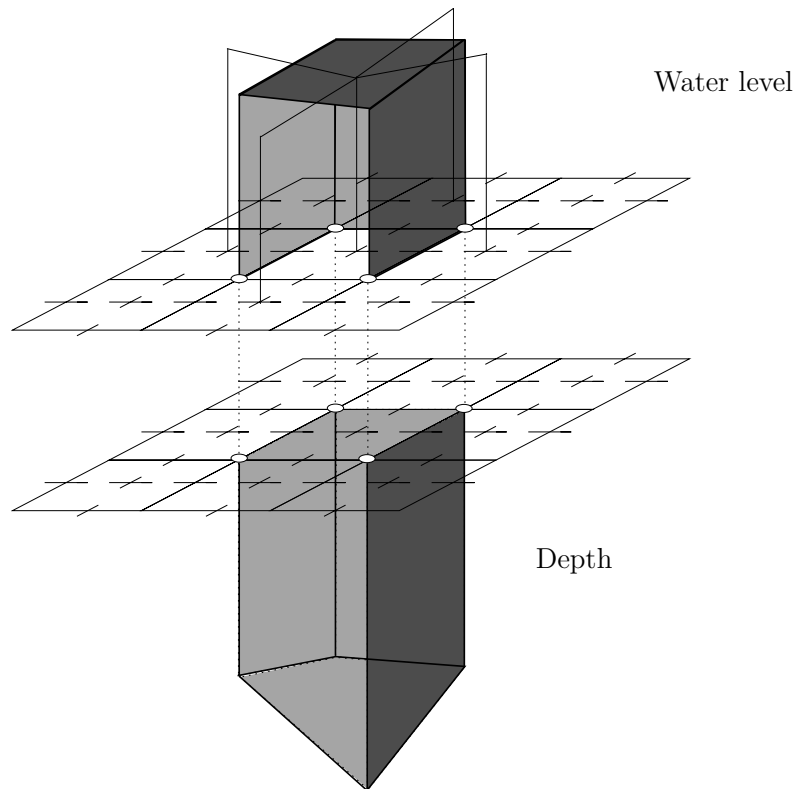


Figure 2.2: Water depth parameters

The corners of the volumes correspond to the grid depth points, the central points, to the water level points. Water flows through the sides of the volumes satisfying the conservation principle

$$storage = input - output .$$

The side of a volume is called a face. WAQUA computes the normal velocity at the midpoints of the faces. Except for the special cases where the bottom gradient

is too steep, the total water depth at the faces corresponds to the sum of the depth values averaged along the face and the water level values averaged perpendicularly to the faces.

WAQUA solves the system of shallow water equations (2.4) using an ADI staggered time integration method over two half time steps. A detailed description of this method can be found in [42]. For a cell of quadrilateral shape, the discretized form of the continuity equation in (2.4) resulting from the ADI method reads as follows

$$H_{cell}^{n+1} - H_{cell}^n = \frac{\tau}{A} \left(\frac{H_1^{n+1}V_1^{n+1} + H_1^nV_1^n}{2}l_1 + \frac{H_2^{n+1}V_2^{n+1} + H_2^nV_2^n}{2}l_2 + H_3^{n+\frac{1}{2}}U_3^{n+\frac{1}{2}}l_3 + H_4^{n+\frac{1}{2}}U_4^{n+\frac{1}{2}}l_4 \right). \quad (2.5)$$

In this equation, U_e and V_e , $e = 1, \dots, 4$, are the normal velocities at the midpoint of the face as presented in figure 2.3-a. l_e , $e = 1, \dots, 4$, are the lengths of the faces and n is the time index. H_e , $e = 1, \dots, 4$, are the total water depths at the midpoint of the faces. Referring to figure 2.3-b, H_e is computed in WAQUA as follows

$$H_e = \frac{\eta(c_1) + \eta(c_2)}{2} + \frac{h(v_1) + h(v_2)}{2}. \quad (2.6)$$

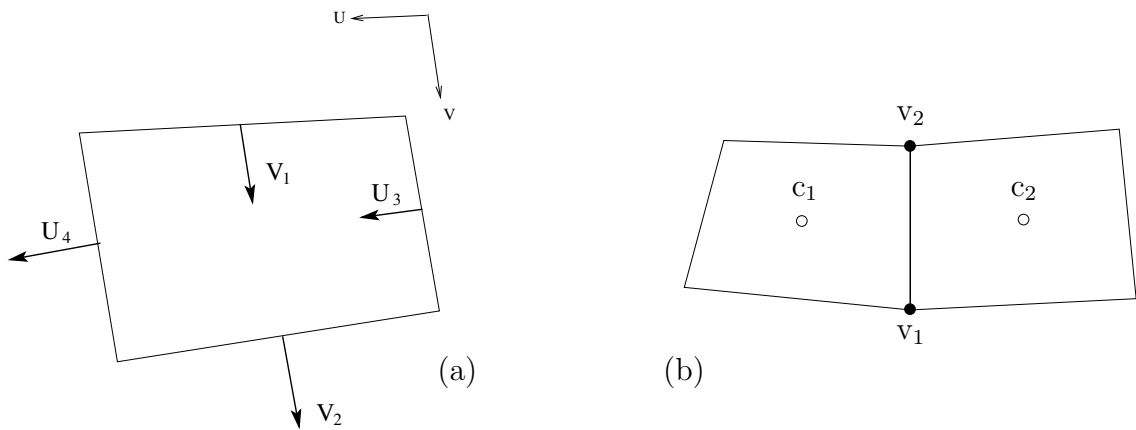


Figure 2.3: Velocities and water depths at the cell faces

Finally, we notice that the 3-D version of WAQUA is called TRIWAQ and will be used as the flow simulation tool in chapters 4 and 5.

2.4 Solution procedure for the transport problem

In general, numerical methods used to solve the advection-diffusion equation are classified as Eulerian, Lagrangian or mixed Eulerian-Lagrangian [33]. In the Lagrangian approach, transport is computed by tracking the movement along characteristics without using a fixed grid. These methods perform reasonably well in case of advective dominated problems by essentially eliminating numerical diffusion but

they usually suffer from local mass balance errors due to the use of interpolated velocities. They also may suffer from numerical instabilities and computational difficulties due to the lack of a fixed grid [55].

On the contrary to Lagrangian methods, Eulerian methods solve directly the advection-diffusion equation at grid points, usually using numerical methods such as finite difference or finite element. They present the advantage of being generally mass conservative and of handling diffusion dominated problems accurately. They are also more adapted to long term simulations which makes them better fitted to scenarios of continuous tracer release for example. It is well known however that they are susceptible to excessive numerical diffusion and artificial oscillations.

Mixed Eulerian-Lagrangian methods try to combine the advantages of both methods. An interesting comparison between the different methods can be found in [56].

A relatively more recent sub-class of the Eulerian methods consists of the higher-order finite volume methods originally developed in the field of computational fluid dynamics. These methods have the advantage of reducing numerical diffusion thanks to their higher order of accuracy and, in the same time, they can prevent the appearance of artificial oscillations when flux limiters are used.

In this thesis, we are interested in this last category of methods which showed to be a good compromise to the techniques mentioned above. The implementation of this type of methods for unstructured grids is a challenging task especially for regions with complex geometry which is often the case when dealing with coastal areas.

2.4.1 Unstructured grid

One of the advantages of using unstructured grids is the great flexibility in discretizing domains with complex geometries, which often occurs when the application deals with natural systems. This was one of the arguments behind the selection of the finite volume method for solving our advection-diffusion problem since this method is very well adapted to the use of unstructured grids (although structured grids can also be used). An extensive comparison between structured and unstructured grids can be found in [14].

In the context of this thesis, the major numerical application concerns the Dutch coast, and as mentioned in section 2.3.2, the flow parameters are computed using the WAQUA/TRIWAQ package based on a structured grid concept. The use of the same grid for the transport presents a number of advantages, notably:

- huge previous efforts have already been done to construct and improve this particular grid of the Dutch coast and it is really worth to benefit from this previous valuable work,
- the reconstruction of a new grid is an enormous task which goes out anyway from the scope of the present thesis,
- in general, the use of different grids results in the necessity of further interpolations to get the parameters at the required places. This is not always

appreciated because the flow conservation becomes difficult to ensure. As will be detailed later in this section, the location of the velocity computational points in WAQUA/TRIWAQ are particularly convenient to ensure some conservation properties for the transport computations.

Although we use exactly the same grid as in WAQUA, it is addressed by our solver as to be unstructured with quadrilateral shaped unit volumes. In fact, the output of the hydrodynamic computations as well as all the time-independent data are reformatted in order to satisfy the input requirement of our unstructured grid-based solver.

The staggered aspect of WAQUA/TRIWAQ grids presents an advantage for our transport solver. The velocities located at the midpoints of the edges and the total water depths at cell centers are exactly what we need to ensure conservation properties. The cell-centered finite volume method to be used in this project appears therefore as a natural choice to ensure conservation. In fact, in this type of discretization, concentration values are located at cell centers, coinciding therefore with the total water depths and the fluxes of pollutant coming in and going out through the faces where the velocities are exactly known. In this context, Perot presents in [35] a discussion on the attractive conservation properties of unstructured staggered mesh schemes.

Further details about unstructured grids techniques can be found in [3], [29], [38], [47].

2.4.2 Finite volume formulation

Finite volume methods are a class of discretization schemes that have proven highly successful in solving a large variety of conservation law systems. They are extensively used in fluid mechanics, meteorology, electromagnetics, semi-conductor device simulation, model of biological processes and many other engineering areas governed by conservative systems that can be written in integral control volume form [2], [20]. References for a general introduction to this class of methods can be found for example in [52] and [36].

The general idea of the finite volume method consists in subdividing the spatial domain into “finite volumes”, or grid cells, and keeping track of an approximation of the integral of the unknown over each of the volumes. The values are modified each time step by the fluxes through the edges of the grid cells. Because they are based on the integral formulation, finite volume methods are closer to the physics than other methods solving the partial differential equation [20], [36]. They are particularly effective when the solution contains discontinuities.

Then, we formulate our finite volume scheme. The physical domain is covered with a grid formed by quadrilateral shaped elements. The area of one element is denoted by A and the boundary ∂A is composed of four elements e of length l_e . The depth-averaged concentration \bar{c} in (2.3) is now simply represented by c . Let $\mathbf{f} = H\mathbf{u}c$. and $\mathbf{g} = HD_h\nabla c$. Equation (2.3) then becomes

$$\frac{\partial Hc}{\partial t} + \nabla \cdot \mathbf{f} = \nabla \cdot \mathbf{g} + HS. \quad (2.7)$$

Integrating (2.7) over one grid cell and applying Green's divergence theorem, we get

$$\int_A \frac{\partial Hc}{\partial t} dA + \sum_{e \in \partial A} \left(\int_e \mathbf{f} \cdot \mathbf{n}_e d\sigma \right) = \sum_{e \in \partial A} \left(\int_e \mathbf{g} \cdot \mathbf{n}_e d\sigma \right) + \int_A H S dA, \quad (2.8)$$

where \mathbf{n}_e denotes the outward unit normal vector to the cell edge e and the discrete sum is calculated over the edges surrounding the cell. The edge integrated advective fluxes $\int_e \mathbf{f} \cdot \mathbf{n}_e d\sigma$ and $\int_e \mathbf{g} \cdot \mathbf{n}_e d\sigma$ are approximated by $f_e l_e$ and $g_e l_e$, where f_e and g_e are the advective and diffusive numerical fluxes, respectively. Now, we can write

$$A \frac{\Delta Hc}{\tau} + \sum_e (f_e l_e) = \sum_e (g_e l_e) + A H S, \quad (2.9)$$

where τ is the time step.

2.4.3 Approximation of the advective flux

In general, the approximation of the numerical advective flux along the edges depends on the location of the flow variables. Recall that in our case, the normal velocities at the midpoints of the edges are available from the flow solver. Therefore, fluxes are computed at the same location on the boundaries ∂A .

Different forms for the advective numerical flux can be found in the literature. See for example [7], [9], [26], [16], [17], [27], [34], [54], and [53]. Following the last 6 references, we use the upwind Enquist-Osher function, which reads

$$f_e(c_e^L, c_e^R) = \frac{u_e + |u_e|}{2} H_e c_e^L + \frac{u_e - |u_e|}{2} H_e c_e^R, \quad (2.10)$$

where c_e^L and c_e^R are the estimated values of the concentration at the left and right sides of edge e . H_e and u_e are the water depth and the normal velocity at edge e , respectively.

For the sake of second-order accuracy, we proceed with a linear reconstruction of the concentration c_e at cells' interfaces. The molecule depicted in figure 2.4 is used as a basis for this reconstruction. In this figure, we call cell 1 and 2 primary cells and cell 3 to 8 secondary cells. The side of the edge at which the concentration gradient is computed depends on the direction of the flow.

The Green-Gauss technique is used to compute the directional gradient. Let us consider the case of figure 2.4, where the flow at edge e is directed from cell 1 to cell 2. Let \mathbf{t}_{1e} be the vector pointing from the center of cell 1 to the midpoint of edge e and A_{134} the area of the shadow triangle delimited by points $(P_1 P_3 P_4)$. It holds that $A_{134} = \frac{1}{2} |\mathbf{t}_{13} \cdot \mathbf{n}_{41}|$ with \mathbf{n}_{41} being the normal to vector \mathbf{t}_{41} . The gradient of the concentration ∇_1 in cell center 1 can be estimated as

$$\nabla_1 \approx \frac{1}{A_{134}} \int_{(134)} \nabla c dA = \frac{1}{A_{134}} \int_{\partial(134)} c \mathbf{n} ds. \quad (2.11)$$

Approximation of the integral using the midpoint rule gives

$$\nabla_1 \approx \frac{1}{A_{134}} \sum_{e \in \partial(134)} c_e \mathbf{n}_e = \frac{\mathbf{n}_{41}(c_1 - c_3) + \mathbf{n}_{13}(c_1 - c_4)}{2A_{134}}. \quad (2.12)$$

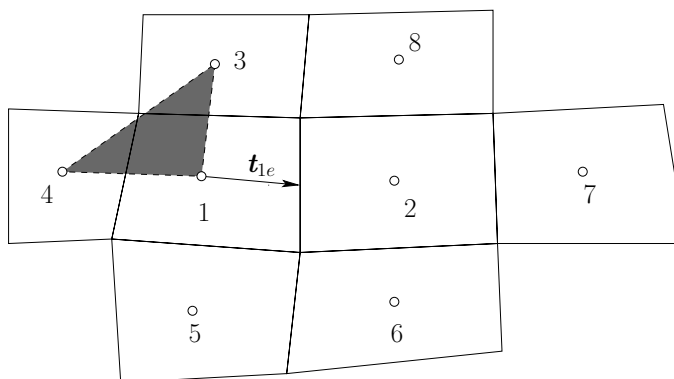


Figure 2.4: Molecule used for the evaluation of the advective flux

It is well known that the use of limiters comes hand in hand with the reconstruction of edge values in order to avoid spurious oscillations. Therefore, the final reconstructed concentration c_e has the following form

$$c_e = c_1 + \psi \delta_{e1} , \quad (2.13)$$

where $\delta_{e1} = \nabla_1 \cdot \mathbf{t}_{1e}$ and ψ is a limiting function.

It is important here to mention the interesting result obtained by Wilders and Fotia, who have studied a system of the form

$$AH \frac{\Delta c}{\tau} = -F(c, \mathbf{v}) . \quad (2.14)$$

This system is similar to our system (2.9) but without the diffusion and source terms and in which the water depth values were assumed to be constant or “frozen” in time. They have found out that provided a suitable choice of the limiter together with a grid which satisfies the so-called TVD triangulation, one can ensure positive solution. A presentation of this result is provided in section 2.7.1 and further details can be found in [54].

With our quadrilateral mesh grid, we would prefer to benefit from this positivity property. This is one reason for keeping the triangular form of the shadow area and using only two out of the three secondary cells to compute the concentration gradient. Although it can be re-adapted so that the shadow area includes all four points, we leave it as it is for the time being notably because now we are left with some flexibility in choosing which two secondary cells to use. In fact, the selection is made in such way that one of the positivity conditions to be presented later in section 2.7.1 remains satisfied.

2.4.4 Boundary treatment of the advective flux

There are three different types of boundary treatments for the edges located at the boundaries of the domain. A zero-flux condition is adopted for closed boundaries or walls. In this case, the advective flux is zero and there is no need to any computation.

For open boundaries, one of two cases may apply: if the normal velocity at the edge is directed to the inside of the domain, a prescribed value of the concentration should be given; otherwise, there is no boundary condition and a simple upwind procedure is applied.

2.4.5 Approximation of the diffusive flux

The evaluation of the numerical diffusive flux reduces basically to computing the concentration gradient along each face of the domain. We propose to use the shadow volume limited by points $(P_1P_aP_2P_b)$ in figure 2.5 in order to compute the gradient of c along face e . In this figure, points P_1 to P_6 are the cell centers where the concentration is computed. These points constitute the basic molecule for the diffusive flux computation. Point P_a is the midpoint of the line (P_4, P_5) and point P_b that of the line (P_3, P_6) .

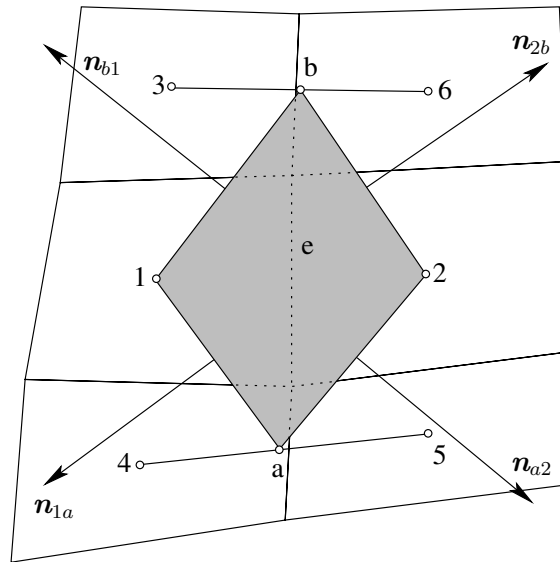


Figure 2.5: Molecule used for the evaluation of the diffusive flux

Green-Gauss contour integration around the shadow volume gives

$$\nabla c \approx \frac{1}{A_{1a2b}} \int_A \nabla c dA = \frac{1}{A_{1a2b}} \int_{\partial A} c \mathbf{n} ds, \quad (2.15)$$

where \mathbf{n} represents the unit normal vector and A_{1a2b} is the area of the shadow volume. It holds that $A_{1a2b} = \frac{1}{2} \overrightarrow{P_1P_2} \cdot \overrightarrow{P_aP_b}^\perp$. The integral can be approximated using the midpoint rule, so we can write

$$\nabla c \approx \frac{1}{A_{1a2b}} (c_{1a} \mathbf{n}_{1a} + c_{a2} \mathbf{n}_{a2} + c_{2b} \mathbf{n}_{2b} + c_{b1} \mathbf{n}_{b1}), \quad (2.16)$$

where, \mathbf{n}_{1a} , for example, denotes the normal vector to face (P_1, P_a) with a length equal to the that of the face. c_{1a} is the average value of the concentrations at points

P_1 and P_a . Rearranging terms of (2.16), we can finally write

$$g_e = D_h \nabla c \cdot \mathbf{n}_e = \frac{(D_h \mathbf{n}_{ab} \cdot \mathbf{n}_e)(c_2 - c_1) + (D_h \mathbf{n}_{12} \cdot \mathbf{n}_e)(c_a - c_b)}{2A_{1a2b}} \quad (2.17)$$

with $c_a = (c_4 + c_5)/2$ and $c_b = (c_3 + c_6)/2$.

2.4.6 Boundary treatment for the diffusive flux

When edge e is close to the boundary, the molecule may not contain the six cells required to compute the diffusive flux. The different possible cases are treated as follows

- Cell P_3 is absent: P_b coincides with P_6 . A similar choice holds if P_4 , P_5 or P_6 do not exist.
- Cells P_3 and P_4 are both absent: the shadow volume is limited to the triangle $P_1P_aP_2$.
- Cells P_2 , P_5 and P_6 are absent, which means that edge e is located on the boundary: we distinguish two cases:
 - If the boundary is closed or a Neumann condition holds, then the viscous flux is zero.
 - Otherwise, the boundary condition is of Dirichlet type and the shadow volume is limited by points $P_1P_4P_mP_3$, where P_m is the midpoint of edge e .
- Cell P_1 is located in a corner, that is, the previous case holds and in addition, cell P_3 is absent: we distinguish also two cases, depending on the type of the boundary condition at the edge separating P_1 and P_3
 - If it is a Neumann condition or a closed boundary, the shadow volume is limited to $P_4P_mP_1$
 - If it is a Dirichlet condition, point P_3 is replaced by the midpoint of the edge separating P_1 and P_3 , where a value of the concentration is supposed to be prescribed.

2.4.7 Time integration

The time integration of (2.3) is achieved using the linearly implicit trapezoidal rule, which results in the following difference equation

$$A \frac{H_{cell}^{n+1} c^{n+1} - H_{cell}^n c^n}{\tau} = -\frac{1}{2} \sum_e H_e^{n+1} u_e^{n+1} l_e c_e^{n+1} - \frac{1}{2} \sum_e H_e^n u_e^n l_e c_e^n + \frac{1}{2} \sum_e g_e^{n+1} l_e + \frac{1}{2} \sum_e g_e^n l_e + \frac{1}{2} A H^{n+1} S^{n+1} + \frac{1}{2} A H^n S^n. \quad (2.18)$$

We have chosen to compute the increment of concentration ($c^{n+1} - c^n$) at each time step instead of the new concentration c^{n+1} itself. By slightly manipulating (2.18), we end up with the following equation, limited here for convenience to the advection terms

$$\begin{aligned} \frac{A}{\tau} H_{cell}^{n+1} (c^{n+1} - c^n) + \frac{1}{2} \sum_e H_e^{n+1} u_e^{n+1} (c_e^{n+1} - c_e^n) l_e = \\ -A \frac{H_{cell}^{n+1} - H_{cell}^n}{\tau} c^n - \sum_e \frac{H_e^{n+1} u_e^{n+1} + H_e^n u_e^n}{2} l_e c_e^n. \end{aligned} \quad (2.19)$$

Here, H_{cell} represents the total water depth at the cell centers. Since the flow solver is totally decoupled from the transport solver, the hydrodynamic coefficients H_{cell} and u_e at time levels n and $n + 1$ are both available during the computation of the solution at time level n . We propose, nevertheless, a further assumption in order to simplify the data processing described in section 2.7.2 by changing the second term of the rhs of (2.19). Our final scheme is therefore

$$\begin{aligned} \frac{A}{\tau} H_{cell}^{n+1} (c^{n+1} - c^n) + \frac{1}{2} \sum_e H_e^{n+1} u_e^{n+1} (c_e^{n+1} - c_e^n) l_e = \\ -A \frac{H_{cell}^{n+1} - H_{cell}^n}{\tau} c^n - \sum_e H_e^n u_e^n c_e^n l_e. \end{aligned} \quad (2.20)$$

2.5 General layout of the model

A general description of the structure of the transport solver is given here. As shown in the flowchart in figure 2.6, the program starts by building a number of time-independent arrays mainly related to the geometry of the grid. These arrays will allow easy access to local information once the time loop starts. Array `plist2` giving for each cell the numbers of the surrounding cells is one example of arrays built at this stage.

Then, the time marching procedure starts. For each time step, the flow coefficients are first read from the corresponding files. Then, a loop starts over faces which first constructs the local support molecule for each face. It then computes the advective and diffusive fluxes through the face function of values of variables from the molecule. After completing this local computation, it finally affects the corresponding entries in the global matrix and in the rhs term.

After the loop over faces, a loop over cells starts. This loop is basically limited here to the inclusion of the first term in the rhs of (2.19). At this stage, most of the global matrix has been built. The linear system is then solved iteratively using the BiCGStab method with an ILU preconditioner.

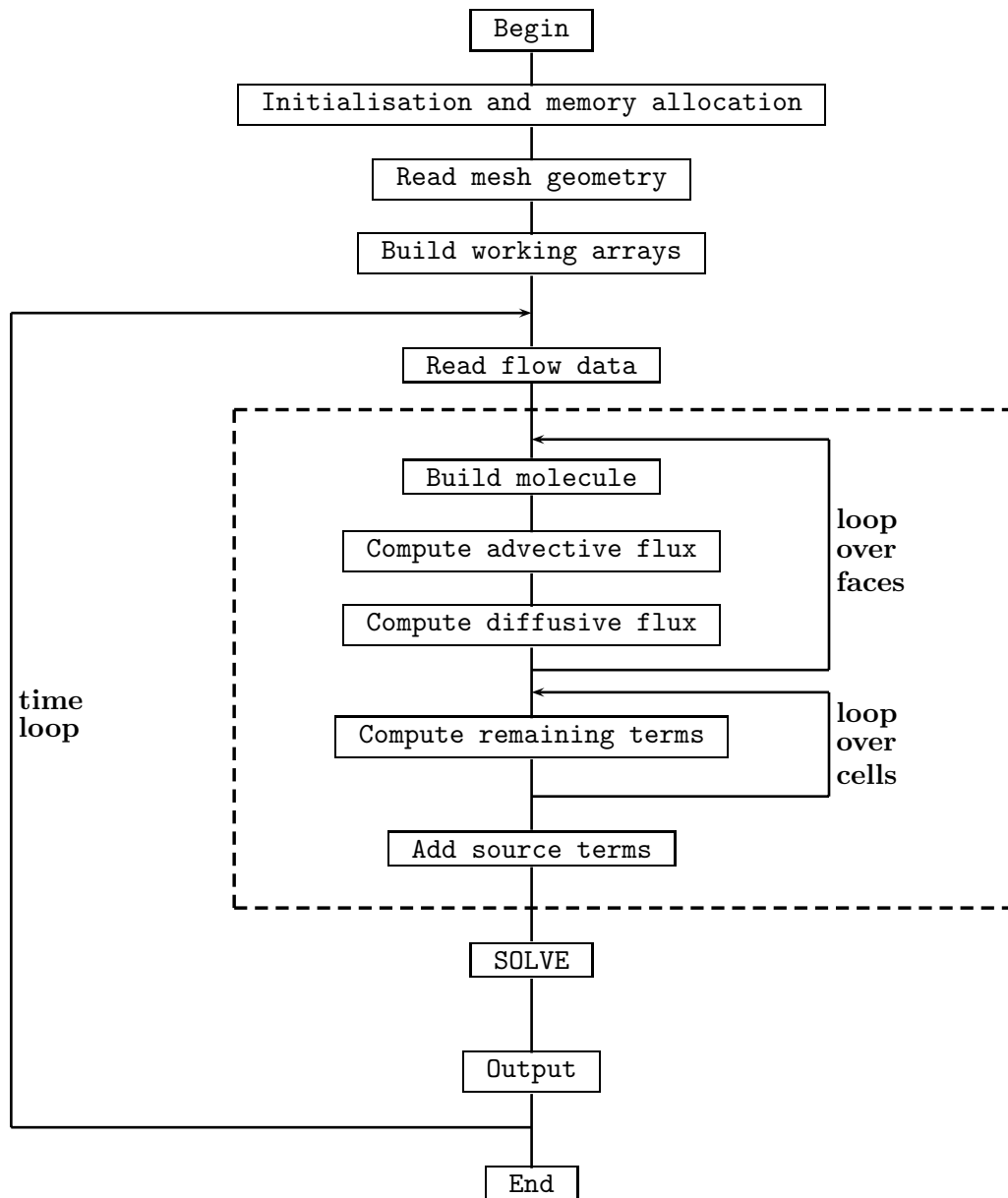


Figure 2.6: General layout of the solver algorithm

2.6 Data structure

In comparison to structured grids, when using a data structure related to unstructured grids, we have to deal with an extra complication in return for the extra flexibility gained in discretizing the physical domain. Data structure is an important issue which may affect the efficiency of the computations considerably.

The geometry of the unstructured grid contains three types of elements or grid objects as they are called in [6], namely cells, edges and nodes. In the 2-D case, the term “face” is also used where it corresponds exactly to the term edge. See figure 2.7. Different variables involved in the transport problem are related to different grid

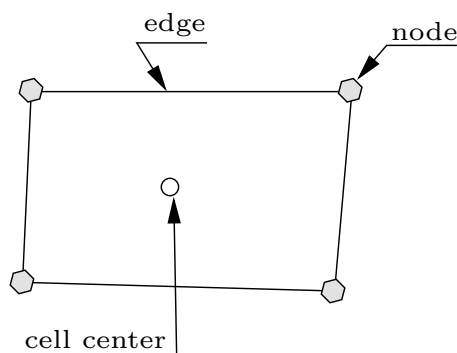


Figure 2.7: Terminology of the grid objects

objects. Table 2.1 summarizes the association of the main variables to their grid objects.

Table 2.1: Grid objects: problem variables association

grid object	problem variable
cell	concentration
	area
	barycenter x-y coordinates
	water depth
edge	normal velocity
	length
	water depth
node	x-y coordinates

The geometry of the domain is input to the solver using two main arrays. The first one contains the x and y coordinates of each node. The second array, named `plist1`, contains for each edge the numbers of the two adjacent cells and the numbers of the two nodes connected by the edge. The order in which these data are entered

is very important in this array as it affects the right sign to the fluxes. In the case of edge f in figure 2.8, the line corresponding to this face should contain in the following order

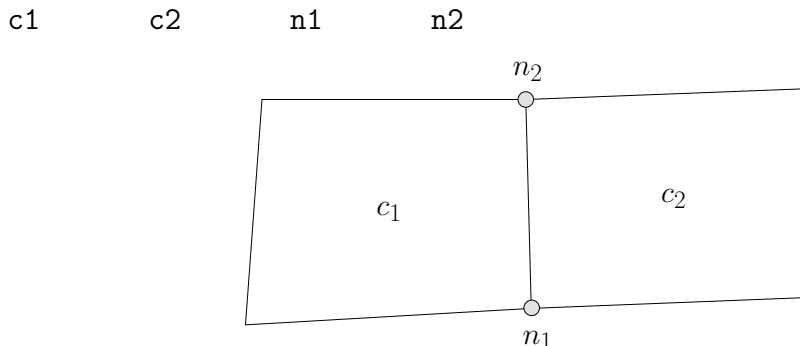


Figure 2.8: Terminology of the grid objects

There is a one-to-one correspondence between the unknown concentrations and the grid cells. However, the building procedure of the main matrix does not always loop over cells. In fact, for computing the flux, which is a major task in the adopted finite volume approach, the matrix is built by looping over edges. This is clearly a better choice since it avoids a redundant computation of the fluxes. For loops over edges, array `plist1` is straightforward to use. There are still terms in the matrix to be built using loops over cells. For this purpose, the previously mentioned internal array `plist2` is constructed within the solver, to give for each cell the numbers of the surrounding cells.

The final resulting matrix is a square matrix of an order equal to the number of cells. It has a sparse aspect since the number of non-zero entries per row is limited to the number of cells contained in the support molecule. Consequently, the matrix needs to be stored efficiently, i.e. only non-zero entries are stored. Here, we use the *compressed sparse row* (CSR) format [39]. For example, suppose cell i has a support molecule containing 5 cells numbered j_1 to j_4 in addition to cell i . The entries of the row in the full matrix corresponding to this cell therefore look like those in the matrix below

$$\begin{array}{cccccc}
 & j_1 & & j_2 & & i & & j_3 & & j_4 \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 \left(\begin{array}{cccccc}
 \cdot & & \cdot & & \cdot & & \cdot & & \cdot \\
 \cdot & & \cdot & & \cdot & & \cdot & & \cdot \\
 \dots & e_1 & \dots & e_2 & \dots & e_i & \dots & e_3 & \dots & e_4 & \dots \\
 \cdot & & \cdot & & \cdot & & \cdot & & \cdot \\
 \cdot & & \cdot & & \cdot & & \cdot & & \cdot
 \end{array} \right) i
 \end{array}$$

This full matrix is now represented in an efficient way using three arrays. First, array ia provides for each individual cell i a pointer indicating the position of the first entry related to this cell in the two other arrays ar and ja . Then, array ar contains the five entries $e_i, e_1 \dots e_5$, and finally, array ja provides the column number (which corresponds, in fact, to the cell number) of the non-zero entries in the same order

in which they are entered in array ar . One extra position in array ia is required to indicate the end of array ar .

$$\begin{array}{c}
 \mathbf{i} \\
 \downarrow \\
 ia = (\dots, ia(i), \dots)
 \end{array}$$

$$\begin{array}{c}
 \mathbf{ia(i)} \\
 \downarrow \\
 ar = (\dots, | e_i, e_3, e_4, e_1, e_2 |, \dots)
 \end{array}$$

$$\begin{array}{c}
 \mathbf{ia(i)} \\
 \downarrow \\
 ja = (\dots, | i, j_3, j_4, j_2 |, \dots)
 \end{array}$$

2.7 Positivity of the 2-D transport solver in real-life applications

Positivity is an essential objective of the researchers especially when approximating conservation laws. In the field of transport computations, it is important that the computed variable, which is often a concentration, is non-negative. Hundsdorfer et al. presented in [22] a method based upon flux limiting to ensure the positivity of a finite difference advection scheme. They showed that the demand of positivity is essentially equivalent to the demand of avoiding numerical under- and overshoots in regions of strong variation. Berzins, who dealt with a finite volume method to solve hyperbolic equations on triangular meshes, came up with modifications to standard flux limiters and possible constraints on the meshes in order to ensure positivity [4].

In the earliest works of Wilders and Fotia which form the starting point of the present work, a proof of positivity of the adopted method was presented. However, runs of real-life applications showed that negative concentrations may still occur. For this reason, we investigate and analyze the causes of the negative concentrations and propose alternatives for remediation. Also, special attention is paid to the conservative properties of the model and its input data in order to avoid the occurrence of unrealistic source and sink terms.

The investigations carried on the positivity problem are based upon runs of the main application case of this thesis; a simulation of the propagation along the Dutch coast of a dissolved pollutant originating from river Rhine. The corresponding numerical results are shown in the next chapter.

2.7.1 Positive advection

The original work by Wilders and Fotia was directed towards two main issues: presenting a positive advective scheme for tracer transport with a proof of positivity

and generalizing the type of meshes from triangular to hybrid type of meshes. The proof of positivity proposed in [54] and [45] applies for a purely advective transport equation. It is based on the main idea of building a final linear system which can be represented by a K matrix. This means that in their final discrete system (2.14), there exist a matrix function $\tilde{Q} = (\tilde{q}_{jk})$ such that the rhs of (2.14) can be written as

$$F_j(c, \mathbf{v}) = \sum_k \tilde{q}_{jk} c_k \quad \text{for every cell } j \text{ in the interior,} \quad (2.21)$$

with $\tilde{q}_{jk} = \tilde{q}_{jk}(c, \mathbf{v})$ which satisfies

$$\tilde{q}_{jk} \leq 0 \text{ for } j \neq k \quad \text{and} \quad \sum_k \tilde{q}_{jk} = 0 \text{ for all } c. \quad (2.22)$$

They state that relations (2.21) and (2.22) are strong demands implying significant properties such as: a steady state is monotonic if the input is monotonic, local minima are nondecreasing and local maxima are nonincreasing and a nonnegative input implies a nonnegative solution. Their proof can be summarized as follows. Opting for (2.10) as the advective numerical flux function, a sufficient condition for an advective K -approximation is that for each cell face e there holds

$$\frac{c_e^R - c_1}{c_2 - c_1} \geq 0, \quad (2.23)$$

and

$$c_e^L - c_1 = 0 \quad \text{or} \quad \frac{c_e^L - c_1}{c_1 - c_k} \geq 0 \text{ for at least one } k \in \{2, 3, 4\}. \quad (2.24)$$

The cells' numbering is based on figure 2.9. The proof of this statement has in

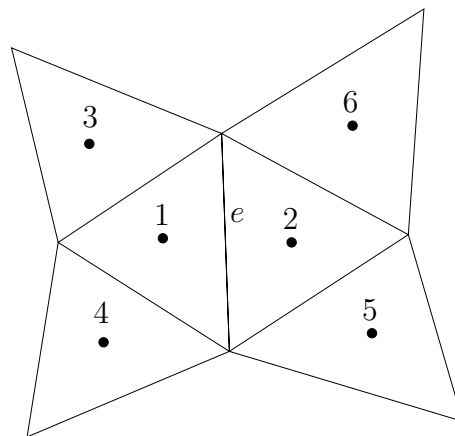


Figure 2.9: Cells' numbering for the triangular grid by Wilders and Fotia.

essence been given in [26]. Assuming a divergence free velocity field, one can write

$$\sum_e f_e(c_e^L, c_e^R) = \sum_e \{f_e(c_e^L, c_1) - f_e(c_1, c_1)\} + \sum_e \{f_e(c_e^L, c^R) - f_e(c^L, c_1)\}. \quad (2.25)$$

It follows that

$$\sum_e f_e(c_e^L, c_e^R) = \sum_e \alpha_e^L + \sum_e \alpha_e^R \quad (2.26)$$

where

$$\alpha_e^L = \begin{cases} u_e (c_1 - c_k) & \text{if } u_e \geq 0, \quad k \in \{2, 3, 4\} \\ 0 & \text{elsewhere} \end{cases}, \quad (2.27)$$

and

$$\alpha_e^R = \begin{cases} -u_e \frac{c_e^R - c_1}{c_2 - c_1} (c_1 - c_2) & \text{if } u_e < 0, \\ 0 & \text{elsewhere} \end{cases}. \quad (2.28)$$

From (2.26), (2.27) and (2.28), it follows that (2.21) and (2.22) are implied by (2.23) and (2.24). Note that (2.22) is equivalent to

$$\frac{c_e^L - c_1}{c_2 - c_1} \leq 1. \quad (2.29)$$

Considering a limiting function of the form

$$\psi = \psi(r), \quad \text{with } r = \frac{\delta_{21}}{\delta_{e1}}, \quad (2.30)$$

it follows easily from (2.13) that

$$\frac{c_e^L - c_1}{c_2 - c_1} = \frac{\psi}{r}. \quad (2.31)$$

The following geometrical requirement on the grid (called TVD triangulation in [26]) is assumed to hold

$$\mathbf{n}_{13} \cdot \mathbf{t}_{1e} \geq 0 \quad \text{and} \quad \mathbf{n}_{41} \cdot \mathbf{t}_{1e} \geq 0, \quad (2.32)$$

which basically means that the cell centers of the two secondary cells have to be located at different sides of the line through \mathbf{t}_{1e} in figure 2.4.

From (2.24), (2.29), (2.12), (2.13) and (2.31) together with the assumption (2.32) on the grid, it follows that

$$0 \leq \psi \leq r, \quad (2.33)$$

is a sufficient condition for an advective K -approximation. There exist a number of limiters which satisfy (2.33) such as the modified Van Leer limiter used in [4]. A differentiable limiter is preferred such as the $R - 1$ limiter proposed by Zijlema [57]

$$\psi = 2 \frac{(r + |r|)r}{(1 + r)^2} \quad \text{if } r \geq 0, \quad \psi = 0 \quad \text{if } r < 0. \quad (2.34)$$

According to [57], condition (2.33) can be less restrictive for regular meshes and for regular meshes, the limiter (2.34) does not present an optimal compromise. Whether

or not this condition can be relaxed for unstructured meshes remains an open question.

When using a hybrid type of grid, i.e. grids with cells of mixed forms, condition (2.32) can become more flexible provided that the shadow volume for the advective flux estimation keeps its triangular shape. In this case, a selection procedure is needed to choose three out of the available number of surrounding cells to make the reconstruction. For the particular case of a quadrilateral grid, there exist always a choice which satisfies (2.32) and one has simply to take care to make the right choice.

Comment on the practical computation of the directional gradient r

For the practical computation of r , the directional gradient defined in (2.33), one should take care of the case where $\delta_{21} = 0$. In our solver, r is computed as follows

$$r = \begin{cases} 0 & \text{if } \delta_{e1} < \varepsilon, \\ 0 & \text{if } \delta_{e1} \geq \varepsilon \text{ and } \delta_{21} < \varepsilon, \\ \frac{\delta_{21}}{\delta_{e1}} & \text{else.} \end{cases} \quad (2.35)$$

where ε is a small number. This means that the scheme reduces to first-order accuracy if the gradient of concentration is less than a given threshold. In our numerical application, we take ε equal to 10^{-8} .

2.7.2 Constancy condition and data preprocessing

When solving our advection-diffusion equation, it is important not to end up with numerical 'wiggles' in the concentration profile. If monotonicity is ensured, the numerical scheme produces no artificial extrema as time progresses [52]. A minimal monotonicity requirement is that an initially uniform scalar field remains uniform in the absence of source and sinks in the equation. This condition is called *the constancy condition* by a number of authors. In [11], a scheme which satisfies the constancy condition is said to be zeroth-order accurate. The violation of this condition expresses the fact that some purely artificial sources and sinks are generated by the discretization, and in such cases, instabilities can arise [21].

In order to satisfy the constancy condition, consistency must hold between the discretization of the flow computations and the discretization of the transport computation [32]. The following definition is given in [21] and the references therein: *A discretization of the advection equation is consistent with continuity if, given a spatially uniform scalar field as an initial datum, and a general flow field, the discretized scalar advection equation reduces to the discretized continuity equation.*

Therefore, we need to analyze the flow and transport computations in order to fulfill this mutual consistency condition.

2.7.3 Conservative flow computations

In section 2.3.2, we mentioned that flow coefficients are computed using the package WAQUA which solves the system of shallow water equations using an ADI scheme.

These computed coefficients are supposed to satisfy (2.5) up to a given accuracy. However, a check of some actual computations shows that, for a number of cells in the domain, this is not always the case. This is generally due to the fact that the maximum number of iterations of some iterative procedures is reached without obtaining the required accuracy of the solution, usually when the geometry of the bottom is highly distorted [19].

As will be described in detail in section 2.7.4, it is very important for the flow coefficients to satisfy the discrete continuity equation up to the desired accuracy. In fact, this directly effects the transport conservative properties. Since the improvement of the flow package is outside the scope of the present work, a simple correction procedure is proposed instead. It consists of computing a new total water depth denoted by \tilde{H}_{cell}^k (with the superscript k indicating the time level), as follows

$$\begin{cases} \tilde{H}_{cell}^0 = H_{cell}^0 \\ \tilde{H}_{cell}^{n+1} = \tilde{H}_{cell}^n + \frac{\tau}{A} \left(\frac{H_1^{n+1}V_1^{n+1} + H_1^nV_1^n}{2} l_1 + \frac{H_2^{n+1}V_2^{n+1} + H_2^nV_2^n}{2} l_2 \right. \\ \left. + H_3^{n+\frac{1}{2}}U_3^{n+\frac{1}{2}}l_3 + H_4^{n+\frac{1}{2}}U_4^{n+\frac{1}{2}}l_4 \right). \end{cases} \quad (2.36)$$

It can be easily verified that (2.5) is indeed exactly verified. This correction in fact affects a limited number of locations and \tilde{H}_{cell} is not expected to differ much from H_{cell} . This is verified in the numerical application section in this chapter. In the remaining part of this chapter, H_{cell} will denote the conservative \tilde{H}_{cell} .

2.7.4 Consistency

The issue of consistency between flow and transport computations was addressed by a number of authors. For example, Lin et al. working on the modeling of tracer transport in the atmosphere realized that the inconsistency between the tracer continuity equation and the underlying equation of continuity of the air can be deleterious especially for long integrations [25]. Leveque [23] formulated a condition to be satisfied by the flow parameters. This condition which fits with his numerical transport scheme is a flow continuity equation under a given required discrete form. He mentioned a number numerical flow schemes for which consistency is automatically obtained. If different schemes were to be used, further velocity projections are needed to make the velocity field divergence free. Taking care that the flow solver is consistent with the transport solver is also a technique proposed by Bonaventura et al. in [5]. Dawson in [10] proposed to add extra correction term to the discrete transport equation to overcome the non conservative velocity field and ensure local conservation. Deleersnijder also proposed in [15] to apply a velocity correction to enforce the continuity equation.

It can be concluded that two methods are generally used to overcome the problem of inconsistency: either selecting the flow numerical scheme and the transport numerical scheme in such way that they are consistent with each other, e.g. [23] and [5], or applying corrections, e.g. [10] and [15].

In our case, different approaches and numerical schemes are used to solve flow and transport problems. We propose a new simple method to ensure consistency between the two. The idea is to compute new flow parameters to be used by the transport solver in such way that consistency is ensured. In fact, we have the advantage that our original velocity field is not arbitrary and we know exactly how it was computed. According to the definition in the beginning of section 2.7.2, our scheme (2.20) has to reduce to (2.5) in case of a uniform scalar field. Taking for example a concentration $c = 1$ in (2.20), we obtain for a quadrilateral cell

$$\begin{aligned} H_{cell}^{n+1} - H_{cell}^n &= -\frac{\tau}{A} \sum_e H_{es}^n u_{es}^n l_e \\ &= -\frac{\tau}{A} (H_{s1}^n u_{s1}^n l_1 + H_{s2}^n u_{s2}^n l_2 + H_{s3}^n u_{s3}^n l_3 + H_{s4}^n u_{s4}^n l_4). \end{aligned} \quad (2.37)$$

We use the index s in the rhs to distinguish between the coefficients of WAQUA and the coefficients to be used by our solver.

We need to choose the proper water depths and velocities in such a way that (2.37) remains satisfied. In analogy (2.5) and (2.37), one can satisfy the consistency with the continuity equation if the following system of equations holds

$$\begin{cases} H_{1s}^n u_{1s}^n l_1 = \frac{H_1^{n+1} V_1^{n+1} + H_1^n V_1^n}{2} l_1 \\ H_{2s}^n u_{2s}^n l_2 = \frac{H_2^{n+1} V_2^{n+1} + H_2^n V_2^n}{2} l_2 \\ H_{3s}^n u_{3s}^n l_3 = H_3^{n+\frac{1}{2}} U_3^{n+\frac{1}{2}} l_3 \\ H_{4s}^n u_{4s}^n l_4 = H_4^{n+\frac{1}{2}} U_4^{n+\frac{1}{2}} l_4 \end{cases}. \quad (2.38)$$

One possible choice which satisfies (2.38) is the following

$$\begin{cases} H_{1s}^n = H_1^n; & u_{1s}^n = \frac{H_1^{n+1} V_1^{n+1} + H_1^n V_1^n}{2H_1^n} \\ H_{2s}^n = H_2^n; & u_{2s}^n = \frac{H_2^{n+1} V_2^{n+1} + H_2^n V_2^n}{2H_2^n} \\ H_{3s}^n = H_3^n; & u_{3s}^n = \frac{H_3^{n+\frac{1}{2}} U_3^{n+\frac{1}{2}}}{2H_3^n} \\ H_{4s}^n = H_4^n; & u_{4s}^n = \frac{H_4^{n+\frac{1}{2}} U_4^{n+\frac{1}{2}}}{2H_4^n} \end{cases}. \quad (2.39)$$

System (2.39) is the kind of correction we need to apply to the original velocities and water depths obtained from WAQUA in a preprocessing step before proceeding with the transport computation. Efficient MATLAB scripts have been written for this purpose.

2.7.5 Dry-wet procedure

Since we are dealing with transport in coastal regions, some parts of the domain change from wet to dry and vice versa following the tidal movements. From a

computational point of view, this may cause violation of the constancy condition if not considered carefully. Since the flow and transport computations are decoupled from each other, we have to make sure that the definition of the wet or dry status of a given cell is consistent between the two solvers.

The following description is reported from the user's guide of WAQUA: *The simulation accounts for tidal flats by considering grid points to be dry at depths less than a given marginal depth, $DEPTHCRIT$. When a velocity point becomes dry, it is taken out of the computation. When a water level point becomes dry, the water level point and the four surrounding velocity points are taken out of the computation. . .* (Figure 2.10 shows the relative locations of these points in a typical staggered grid element). In other words, the velocity at a given velocity point is set equal to zero as soon as:

1. the total water depth at this face is lower than the threshold
2. the total water depth at the center of one of the neighboring cells is lower than the threshold.

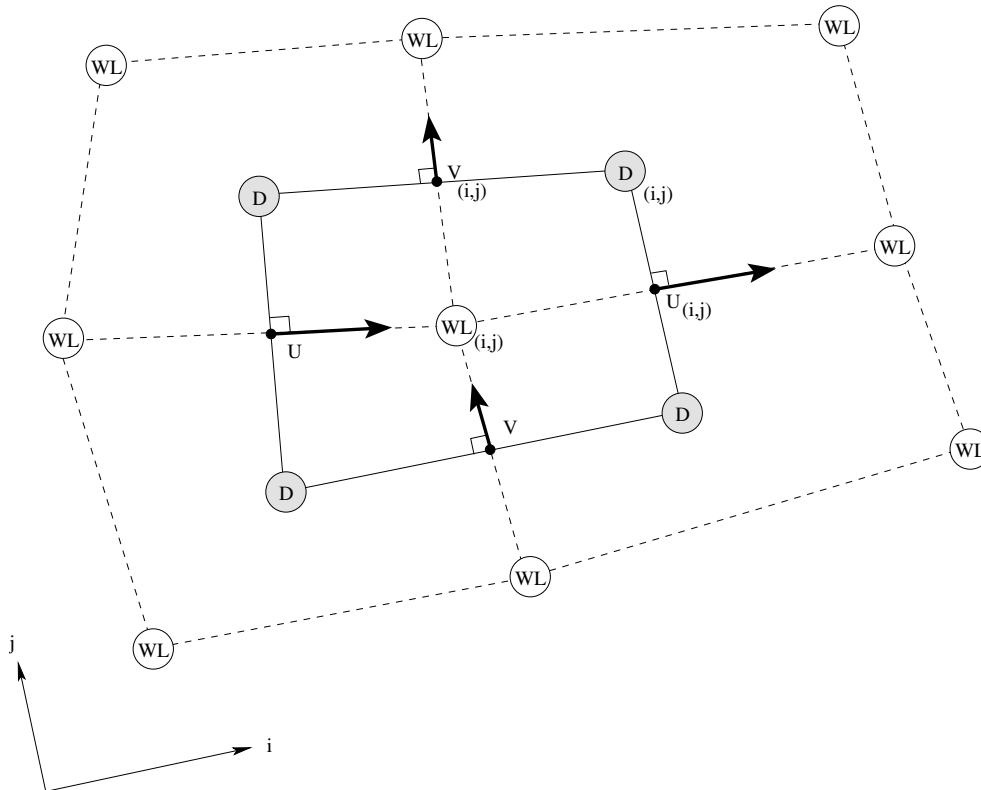


Figure 2.10: Staggered grid in WAQUA. WL: water level point; D: depth point; U, V: velocity points.

Notice that WAQUA defines dry "grid points", which may be velocity points or water level points, whereas our transport solver uses the concept of dry "cells". In order to cope with this definition, we consider in our transport solver a given cell to be dry if one of the following conditions is satisfied:

1. if the total water depth at the center of a given cell is less than a threshold equal to 0.03 m.
2. if a face separating two cells has a depth of less than 0.03 m, then the status of at least one of these two cells is changed to dry.

In order to ensure full consistency between the two definitions, we have to take care of the following additional points. First, if we apply (2.20) to a dry cell, it writes

$$H_{cell}^{n+1} c^{n+1} - H_{cell}^n c^n = 0 , \quad (2.40)$$

since the velocities at the four faces of the cells are equal to zero. Thus no special problem occurs if the cell is considered dry because its total water depth is lower than 0.03 m. However, following the second point of the above definition, the cell may be considered dry although its total water depth is higher than the threshold. In this case, H_{cell}^n is not set equal to the threshold and is therefore very likely different from H_{cell}^{n+1} . Equation (2.40) then gives

$$c^{n+1} = \frac{H_{cell}^n}{H_{cell}^{n+1}} c^n , \quad (2.41)$$

which implies a new concentration different from c^n .

The proposed correction consists in simply setting for each dry cell

$$H_{cell}^n = H_{cell}^{n+1} , \quad (2.42)$$

and therefore the concentration will remain unchanged. We expect that this artificial correction will not introduce any significant error in the overall computation since we are dealing with very low water depths, generally of the order of a few centimeters.

The second point requiring special treatment is related to cells which are immediate neighbors to dry cells. The problem also occurs when a cell is dry because only one of its faces is dry. Let's consider the case shown in figure 2.11. The arrows in this figure represent the exchanged fluxes. We assume that the cell on the left is dry because the total water depth at face e is lower than the threshold. The fluxes at the other faces of this dry cell can be different than zero which is not desirable since the fluxes coming in or going out from a dry cell should normally be equal to zero. Simply putting these fluxes equal to zero will violate the continuity equation for the neighboring cells. Therefore, the following correction is proposed. If we rewrite (2.20) for the wet cell shown in figure 2.11 in terms of fluxes, it gives

$$\frac{H_{cell}^{n+1} c^{n+1} - H_{cell}^n c^n}{\tau} = \frac{1}{A} (f_1 + f_2 + f_3 + f_4) , \quad (2.43)$$

where the f_i indicate the fluxes according to figure 2.11. This is equivalent to

$$c^{n+1} = \frac{1}{H_{cell}^{n+1}} \left\{ H_{cell}^n c^n + \frac{\tau}{A} (f_1 + f_2 + f_3 + f_4) \right\} , \quad (2.44)$$

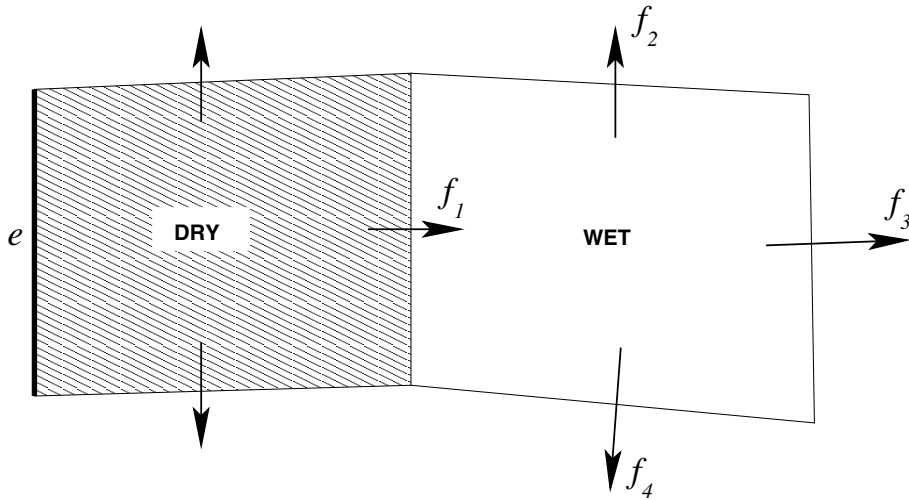


Figure 2.11: Dry-wet neighbors

or again to

$$c^{n+1} = \frac{1}{H_{cell}^{n+1}} \left\{ \left(H_{cell}^n + \frac{\tau \cdot f_1}{A \cdot c^n} \right) c^n + \frac{\tau}{A} (f_2 + f_3 + f_4) \right\}. \quad (2.45)$$

Equation (2.45) must always hold in order to preserve the constancy condition. Therefore, we propose the following correction steps

1. compute the flux f_1
2. replace H_{cell}^n by $H_{cell}^n + \frac{\tau \cdot f_1}{A \cdot c^n}$
3. set $f_1 = 0$
4. repeat this procedure for the rest of the non-zero fluxes of the dry cell

Again here, the error introduced following this correction is not expected to have a significant effect on the accuracy of the overall computation.

2.8 Improvement of the computational efficiency in the 2-D case

The previous sections of the present chapter described various aspects of our finite volume transport solver and mentioned that the transport computation is decoupled from the flow computation. The adopted flow solver was also briefly described.

Decoupling transport and flow has the advantage that the flow computations does not need to be repeated, for example when simulating different scenarios of pollutant propagation under the same flow conditions. The disadvantage is, however, the large amount of data to be computed and stored beforehand. The coefficients needed are the velocities and the water depths at the faces of the cells and the total water depths at their centers. These are time-dependent variables and therefore have to

be available for every computational time step and for the whole simulation period. This may result in a huge disk space to store the data files.

In the present section, new techniques aimed at improving the computational efficiency of the decoupled flow-transport modeling is presented. Allowing the use of different time steps for the flow and for the transport computations, reducing the necessary amount of storage and providing less expensive computations are the main objectives of these techniques.

2.8.1 Time integrated flow data

Section 2.7.4 emphasized the need for consistency between the flow and transport solvers and presented a method to produce proper flow coefficients. This method requires adoption of the same time step as used for the flow computation. We have indicated that running our solver with a time step of 1 minute for a duration of two months requires a storage capacity of more than 165 Gb.

Although it is possible nowadays to use devices capable of handling such an amount of data, it would be much preferable to reduce the required storage space. This can be achieved for example by using larger time steps are used for the transport computations. In fact, the idea presented in this section consists of computing, out of the available small time step data, some new flow coefficients at larger time steps while ensuring at the same time the consistency condition and the conservation property (2.37). The new set is in fact a time integrated flow data.

We can write (2.36) for p successive time steps of the flow model

$$\begin{aligned}
 H_{cell}^{n+1} - H_{cell}^n &= \tau/A \left\{ \frac{H_1^{n+1}V_1^{n+1} + H_1^nV_1^n}{2}l_1 + \frac{H_2^{n+1}V_2^{n+1} + H_2^nV_2^n}{2}l_2 + \right. \\
 &\quad \left. H_3^{n+\frac{1}{2}}U_3^{n+\frac{1}{2}}l_3 + H_4^{n+\frac{1}{2}}U_4^{n+\frac{1}{2}}l_4 \right\} \\
 H_{cell}^{n+2} - H_{cell}^{n+1} &= \tau/A \left\{ \frac{H_1^{n+2}V_1^{n+2} + H_1^{n+1}V_1^{n+1}}{2}l_1 + \frac{H_2^{n+2}V_2^{n+2} + H_2^{n+1}V_2^{n+1}}{2}l_2 + \right. \\
 &\quad \left. H_3^{n+\frac{3}{2}}U_3^{n+\frac{3}{2}}l_3 + H_4^{n+\frac{3}{2}}U_4^{n+\frac{3}{2}}l_4 \right\} \tag{2.46}
 \end{aligned}$$

$$\begin{aligned}
 &\cdot \\
 &\cdot \\
 &\cdot \\
 H_{cell}^{n+p} - H_{cell}^{n+p-1} &= \tau/A \left\{ \frac{1}{2}(H_1^{n+p}V_1^{n+p} + H_1^{n+p-1}V_1^{n+p-1})l_1 + \right. \\
 &\quad \frac{1}{2}(H_2^{n+p}V_2^{n+p} + H_2^{n+p-1}V_2^{n+p-1})l_2 + \\
 &\quad \left. H_3^{n+p-\frac{1}{2}}U_3^{n+p-\frac{1}{2}}l_3 + H_4^{n+p-\frac{1}{2}}U_4^{n+p-\frac{1}{2}}l_4 \right\} . \tag{2.47}
 \end{aligned}$$

Summing these equations together, we get

$$\begin{aligned}
H_{cell}^{n+p} - H_{cell}^n &= H_{cell}^{N+1} - H_{cell}^N \\
&= \frac{\tau}{A} \left\{ \left(\frac{H_1^n V_1^n}{2} + H_1^{n+1} V_1^{n+1} + \dots + H_1^{n+p-1} V_1^{n+p-1} + \frac{H_1^{n+p} V_1^{n+p}}{2} \right) l_1 \right. \\
&\quad + \left(\frac{H_2^n V_2^n}{2} + H_2^{n+1} V_2^{n+1} + \dots + H_2^{n+p-1} V_2^{n+p-1} + \frac{H_2^{n+p} V_2^{n+p}}{2} \right) l_2 \\
&\quad + \left(H_3^{n+\frac{1}{2}} U_3^{n+\frac{1}{2}} + H_3^{n+\frac{3}{2}} U_3^{n+\frac{3}{2}} + \dots + H_3^{n+p-\frac{1}{2}} U_3^{n+p-\frac{1}{2}} \right) l_3 \\
&\quad \left. + \left(H_4^{n+\frac{1}{2}} U_4^{n+\frac{1}{2}} + H_4^{n+\frac{3}{2}} U_4^{n+\frac{3}{2}} + \dots + H_4^{n+p-\frac{1}{2}} U_4^{n+p-\frac{1}{2}} \right) l_4 \right\}, \tag{2.48}
\end{aligned}$$

where N is the new counter of the transport time step. Obviously, the new transport time step T has to be a multiple of the flow time step; here, $T = p\tau$. In analogy to (2.37), we can make the following choices in order to ensure consistency

$$\left\{ \begin{array}{l}
H_{S1} = H_1^n \\
U_{S1} = \frac{1}{H_{S1}} \frac{1}{p} \left(\frac{H_1^n V_1^n}{2} + H_1^{n+1} V_1^{n+1} + \dots + H_1^{n+p-1} V_1^{n+p-1} + \frac{H_1^{n+p} V_1^{n+p}}{2} \right) \\
H_{S2} = H_2^n \\
U_{S2} = \frac{1}{H_{S2}} \frac{1}{p} \left(\frac{H_2^n V_2^n}{2} + H_2^{n+1} V_2^{n+1} + \dots + H_2^{n+p-1} V_2^{n+p-1} + \frac{H_2^{n+p} V_2^{n+p}}{2} \right) \\
H_{S3} = H_3^n \\
U_{S3} = \frac{1}{H_{S3}} \frac{1}{p} \left(H_3^{n+\frac{1}{2}} U_3^{n+\frac{1}{2}} + H_3^{n+\frac{3}{2}} U_3^{n+\frac{3}{2}} + \dots + H_3^{n+p-\frac{1}{2}} U_3^{n+p-\frac{1}{2}} \right) \\
H_{S4} = H_4^n \\
U_{S4} = \frac{1}{H_{S4}} \frac{1}{p} \left(H_4^{n+\frac{1}{2}} U_4^{n+\frac{1}{2}} + H_4^{n+\frac{3}{2}} U_4^{n+\frac{3}{2}} + \dots + H_4^{n+p-\frac{1}{2}} U_4^{n+p-\frac{1}{2}} \right).
\end{array} \right. \tag{2.49}$$

Coefficients computed in this manner satisfy the following new continuity equation written for the new time step T

$$H_{cell}^{N+1} - H_{cell}^N = -\frac{T}{A} (H_{S1} U_{S1} l_1 + H_{S2} U_{S2} l_2 + H_{S3} U_{S3} l_3 + H_{S4} U_{S4} l_4). \tag{2.50}$$

The storage space required for the new coefficients is only $1/p$ times the original space, a considerable reduction. The overall performance of this method is assessed in the numerical application section.

2.8.2 Periodic continuation

We propose in this section a second technique which may lead to less expensive computations and to further reduction of flow data storage. Water flows in coastal regions are basically dominated by the tidal effects well known by their quasi-periodical aspect. This fact may be useful in the sense that data representing a limited period

of time can be used repeatedly for the simulation of longer periods. We call this procedure “periodic continuation” [31].

The choice of the length of the period is the most sensitive one since it may considerably affect the residual velocities. These velocities are obviously of great importance in the advection process. Furthermore, the length of the tidal period may differ according to the location in the domain, making the choice more difficult.

Figure 2.12 shows the variation of the total water depth at a certain location of the domain of the Kuststrookmodel during a period of two months. This domain will be used as the major application case for the transport solver and will be described in detail in the next chapter. This figure puts into evidence the one-lunar-month periodicity. For the same domain, the half moon cycle, which corresponds to some 28 or 29 tidal periods, is proposed in [45] and this, too, looks like a rather reasonable choice. However, since such long periods of time do not present a real advantage with respect to reduction of the volume of data, we are more interested to the half-day periodicity, which we are going to investigate in this section.

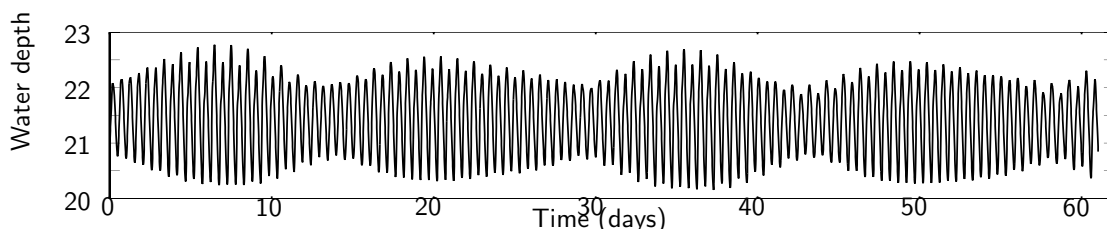


Figure 2.12: Total water depth at cell 10000

We need to know the precise duration of one period, this value being highly important for the outcome of the computations. We propose to determine it by examining the water height at a few sample locations. Figure 2.13-a shows the water elevation between two successive peaks at the sample locations. A close-up of the last portions of these curves (figure 2.13-b) shows that the duration of the periods varies for different locations roughly between 12 h 25 min and 13 h. Therefore, we assume for the moment a period length of 12h 45min, after which all the flow data files are “rebound”. In all cases, different values of the period length will be tested in the numerical application in the next chapter.

2.9 Conclusions

The present chapter is dedicated to a general description of the 2-D transport solver. The governing equation and the numerical solution procedure adopted are described in detail and a short overview is given of aspects related to the computing algorithm.

In particular, section 2.4.3 mentioned the method adopted in order to ensure the positivity of the resulting concentrations. For some practical applications, we noticed that although the conditions for positivity had been met, negative concentrations still occurred. What is more, problems in mass conservation were detected. Consequently, it was decided that further research in this direction was needed and improvements in the conservation of mass and the positivity of the solution for the transport solver were needed. These are important issues for accurate transport

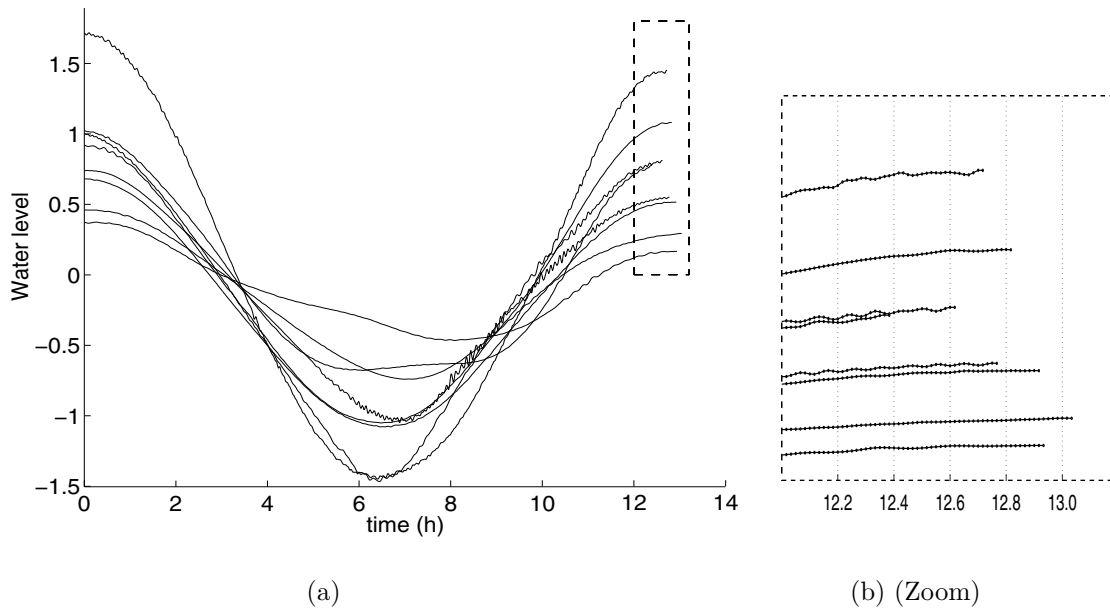


Figure 2.13: Water level elevation at some randomly chosen cells.

simulation. The main idea is to avoid the appearance of artificial source and sink terms by ensuring consistency between the flow and transport solvers. In our case, the velocity field is not arbitrary, that is, we know how it was computed. Based on this, we proposed the following preprocessing steps:

- Computed flow data have to satisfy exactly the discrete flow continuity equation up to machine accuracy. A posteriori verification whether these data do really satisfy the continuity equation is necessary as real-life applications show that this is not always the case. A correction procedure is proposed to obtain precise conservative coefficients.
- New flow coefficients are computed based on the data from the previous step. The idea is that we have a set of data which are conservative in the sense of the flow ADI scheme and from these, we create another set which satisfy **our** discrete continuity equation. In fact, conservation does not have an absolute meaning but is always related to a certain scheme. A method to compute the new coefficients is proposed in (2.39).
- A further step related to the dry-wet implementation may also be needed. Special attention must be paid to the way of implementing the dry-wet procedure for the constancy condition. Here again, matching of the transport and the flow computations is required.

Another important factor dealt with in this chapter concerns data storage. When the transport computations are decoupled from flow computations, as in our case, the storage of a very large amount of data is required. In order to minimize this amount, we show that it is possible to use larger time steps by combining the data computed with a smaller time step. It is also possible to assume periodic continuation of the flow data.

In the next chapter, the different issues mentioned above are tested and evaluated in the context of a numerical application of the 2-D solver.

Chapter 3

Numerical Results of the 2-D solver

3.1 Introduction

After the presentation in chapter 2 of our 2-D transport solver and the related issues of positivity and data storage reduction, we proceed in the present chapter with the numerical application. The selected real life application deals with one of the critical problems of the Dutch coastal zone: the pollution of the rich ecological area in the northern part of the Netherlands, called the Dutch Shallows, also known as the Waddenzee. Part of the pollution in this area originates from the Nieuwe Waterweg, the inlet of the major harbor of Rotterdam. In fact, the river Rhine, which crosses a large part of Europe, is a potential source of pollutants towards the North Sea. Due to the large ocean streams, the continuous release of pollution from the Rhine waters is transported northward in the direction of the inlets of the Waddenzee [13].

There is also a need to simulate possible environmental accidents which may result in heavy impacts on this fragile area. In this context, our model can be a valuable tool to determine the extent of the resulting contaminated area as well as the time scale of the pollution propagation.

3.2 Real-life application

The proposed numerical application considers the domain known as the *Kuststrook-model*, representing a strip of about 400×60 km of the north sea along the Dutch coast. Figure 3.1 shows the 20175 cells grid used to represent the domain. This grid also originates from the National Institute for Coastal and Marine Management (RIKZ).

In order to apply our transport solver to this domain, first we need to make available the required flow coefficients using the flow solver WAQUA. Starting the WAQUA run using some arbitrary constant initial conditions, we assume that a

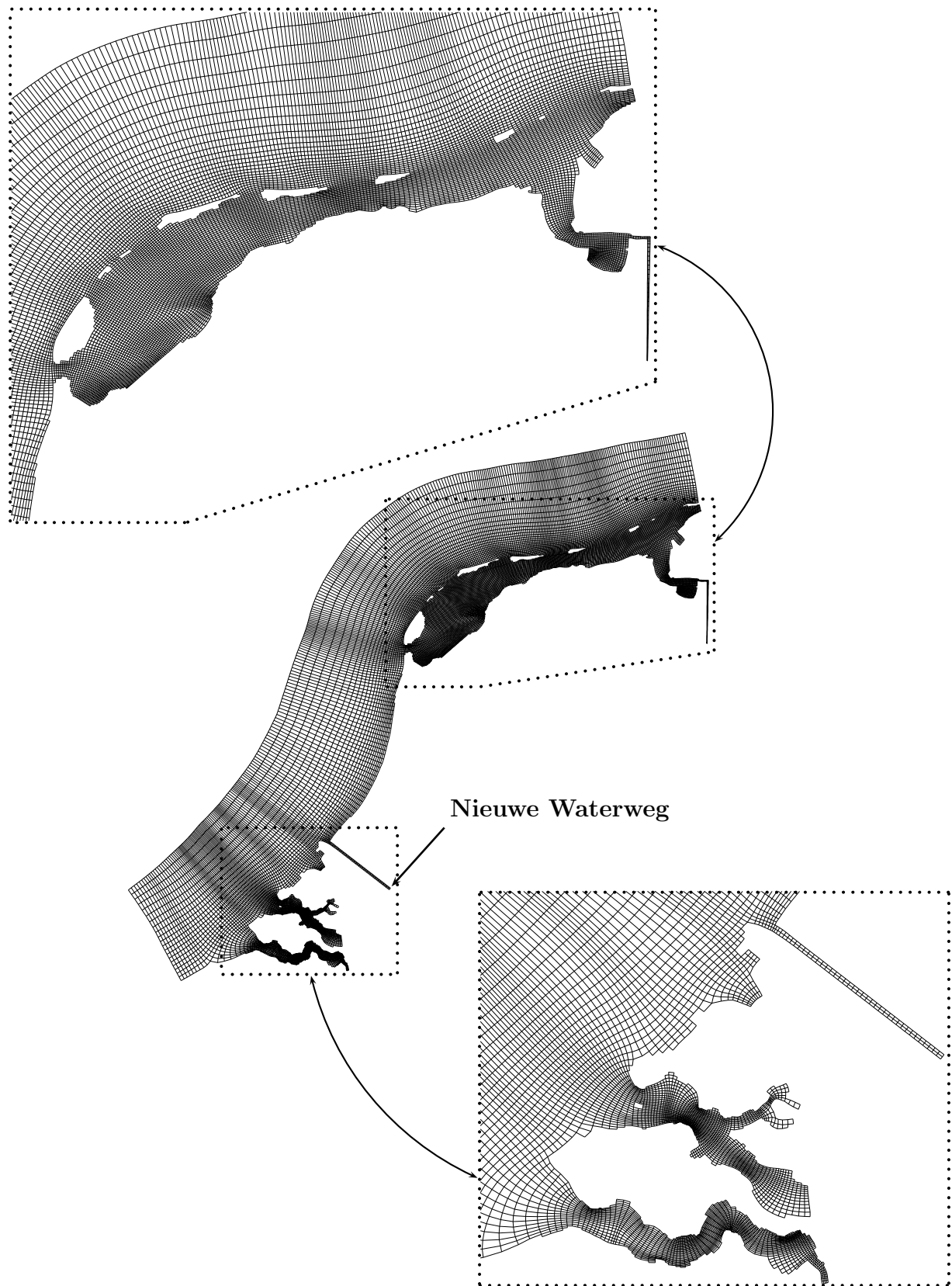


Figure 3.1: Grid of Kuststrookmodel

simulation over a period of one month is long enough in order to start producing realistic data afterwards. The boundary conditions at the open boundaries of the domain are computed based on a harmonic analysis of the tides at these boundaries. The time step in the flow computation is equal to 1 min.

WAQUA produces so-called SDS files containing the output under binary format. Special related programs can be used to extract the data from an SDS file and convert it to the required formats, notably the MATLAB .mat format. MATLAB is in fact used to process and prepare the data according to (2.39). For each time step, we need the following data files:

- file containing the values of the velocities and the total water depths at the edges of the cells. Our grid contains 41439 edges, and the size of one file is 1.4 Mb.
- file containing the values of the total water depths at the centers of the cells. The size of the file is 0.5 Mb.
- file defining the type and the values of the boundary conditions. The size of the file is 0.05 Mb.

First of all, we verify the correction on the total water depth proposed in (2.36). Our computation shows that \tilde{H}_{cell} is only very slightly different from H_{cell} . Figure 3.2 shows the maximum and minimum absolute difference throughout the domain between H_{cell} and \tilde{H}_{cell} for each time step and during one tidal period. The same pattern is basically repeated for the subsequent periods. We can see that this dif-

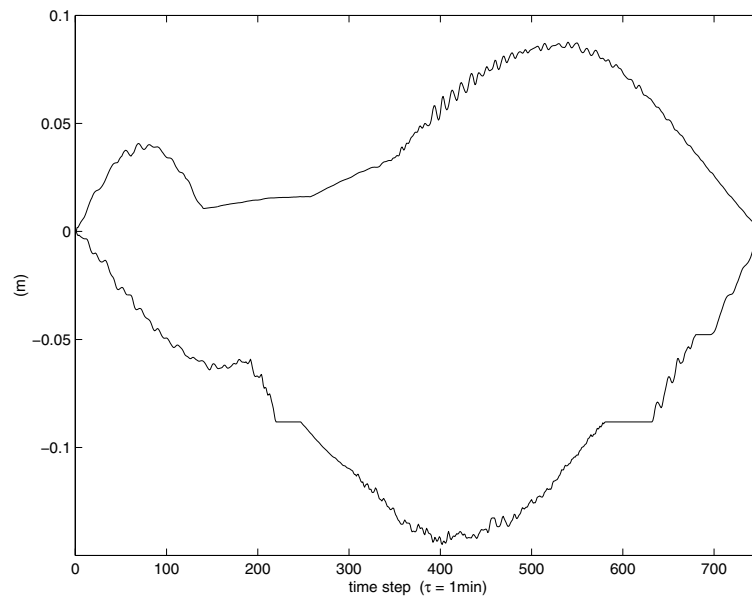


Figure 3.2: Difference between \tilde{H}_{cell} and H_{cell}

ference remains very limited and, therefore \tilde{H}_{cell} can be used as a fully conservative water depth.

Running our solver with a time step of 1 minute (the same as for the flow computation) for a duration of, say, two months, would require a storage capacity

of around 165 Gb for the input files. Extra space is also needed for the transient SDS and MATLAB files.

We proceed first with the verification of the constancy condition. We set the initial concentration equal to 1 for the whole domain as well as for the inflows to the domain through the open boundaries. No source term is added to the system. Figure 3.3 shows that the constancy condition is satisfied which ensures that no sources or sinks are introduced.

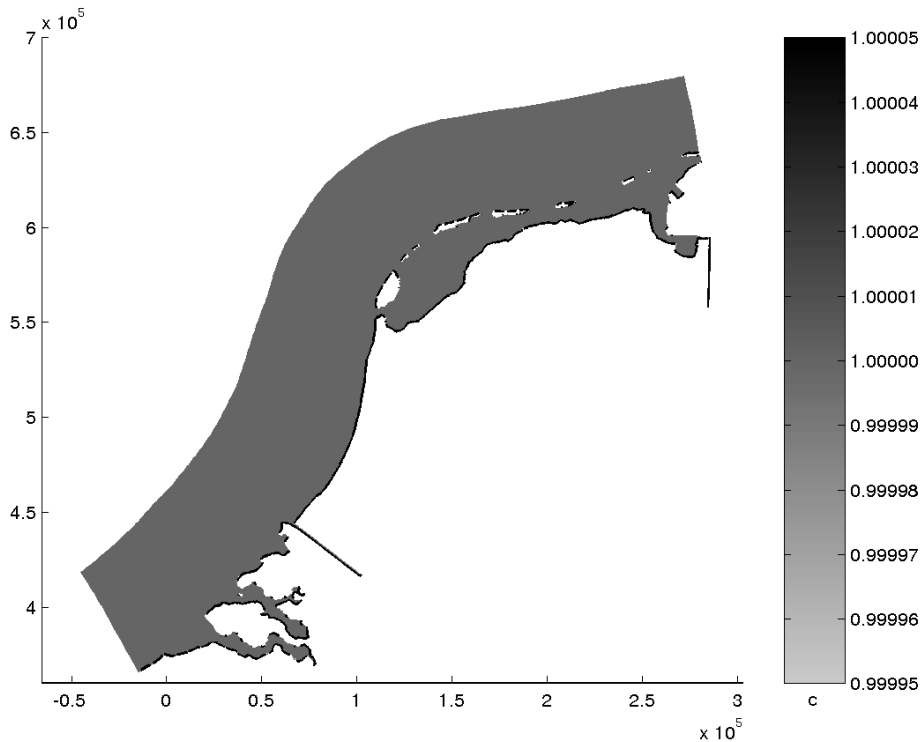


Figure 3.3: Constant solution

Then, we simulate a continuous release of a dissolved pollutant at the upstream boundary of the Nieuwe Waterweg ($x = 101200$, $y = 416500$). Based on the same time step of 1 min, figure 3.4 shows the propagation of the pollutant after 30000, 60000 and 90000 time steps (62.5 days). We can see that the contaminated area extends along the coastline towards the north and reaches the Waddenzee around 2 months after the beginning of the release. In the following, we consider the result of this full simulation as our reference solution.

Checking for negative values of concentrations, figure 3.5 shows the lowest computed concentration throughout the domain for all time steps. This figure shows that except for some fluctuations between some extremely small negative values, the lowest concentration is not below zero.

The result of the simulation is compared to the result obtained from the transport module of WAQUA. The situation after 90000 time steps is shown in figure 3.6. It is clear from the two figures that the pollutant propagation is very similar in both simulations. The result of WAQUA, however, shows negative concentrations in more

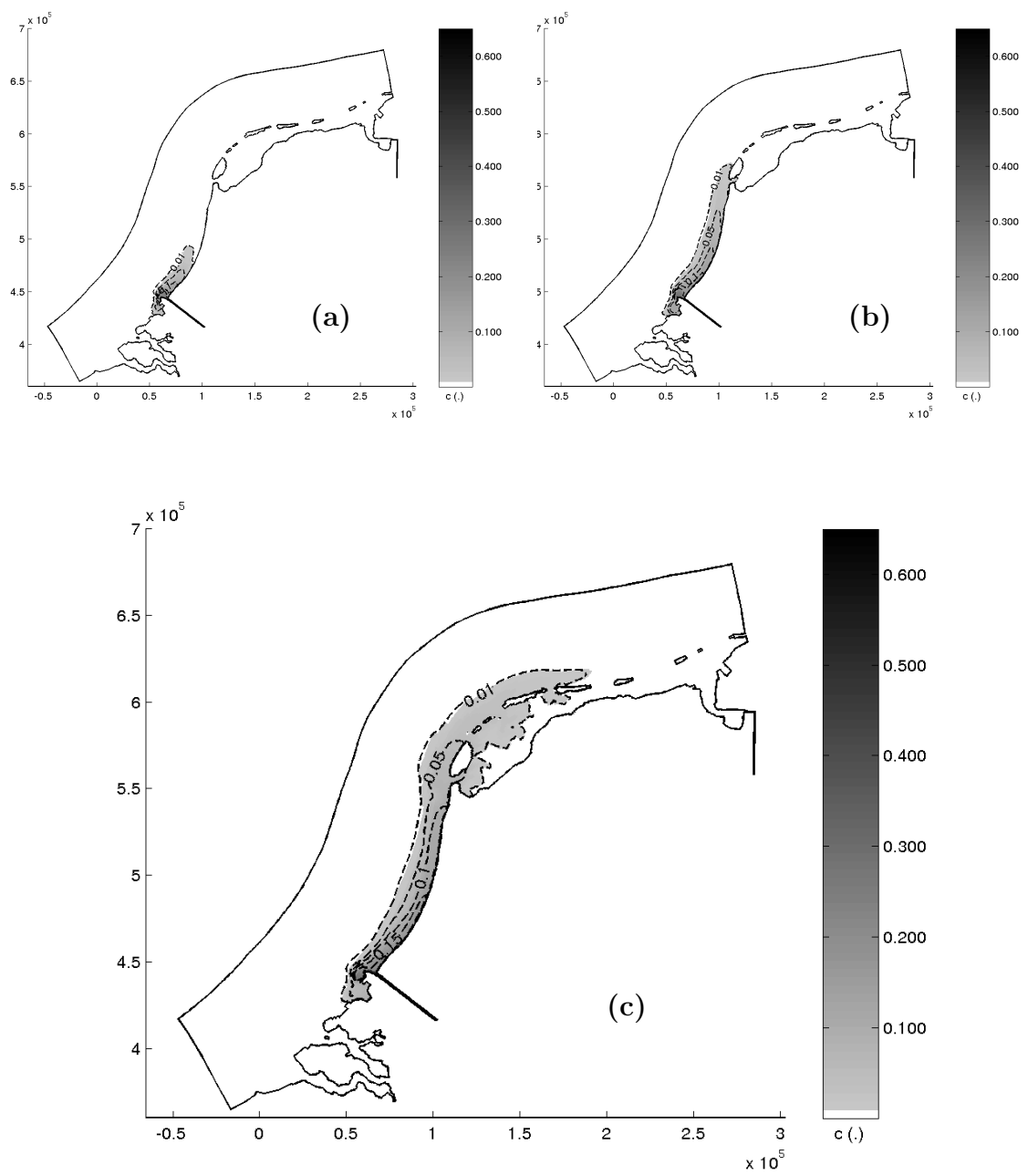


Figure 3.4: Spreading of the tracer after 30000 (a), 60000 (b) and 90000 (c) time steps. ($\tau = 1min$)

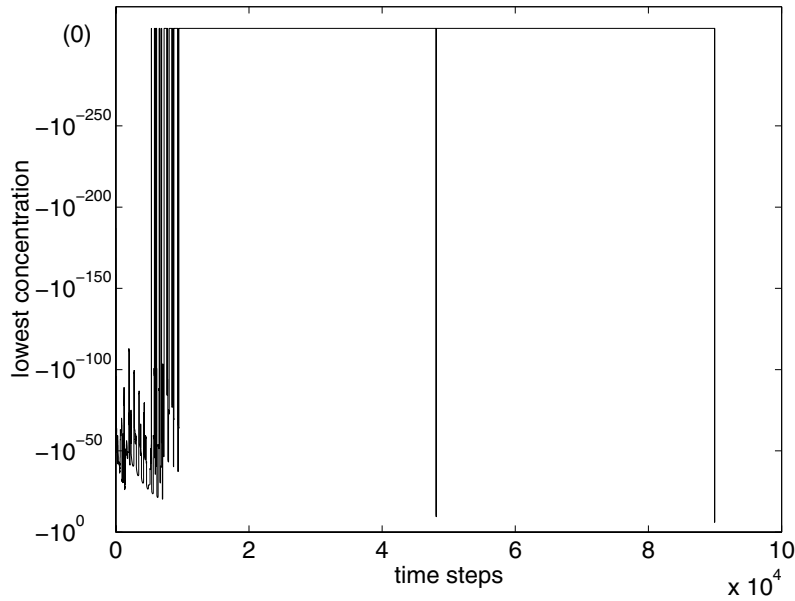


Figure 3.5: Lowest concentration ($\tau = 1$ min)

than 2000 locations, the lowest being of the order of -10^{-4} . Table 3.1 presents a more detailed comparison between the two. We can read in this table the minimum and maximum concentration at a given time step, the number of cells having a negative concentration, the total mass of pollutant, the square root of the sum through the domain of the squared differences of the concentrations, and the sum of the absolute differences.

Table 3.1: Comparison between the results of WAQUA and the reference solution at time step 90000.

	Reference solution	WAQUA
Minimum concentration	0	-0.00043
Cells with negative concentration	0	2100
Maximum concentration	0.561	0.606
Total mass of pollutant	$3.63 \cdot 10^9$	$3.66 \cdot 10^9$
$\sqrt{\sum (C_{ref} - C_{WAQUA})^2}$	-	1.085
$\sum C_{ref} - C_{WAQUA} $	-	50.073

* C_{ref} : concentration from the reference solution, $\tau=1$ min

3.3 Time integrated flow data

Here, we evaluate the efficiency of the technique of using time integrated flow data presented in section 2.8.1 through its application to the Kuststrookmodel.

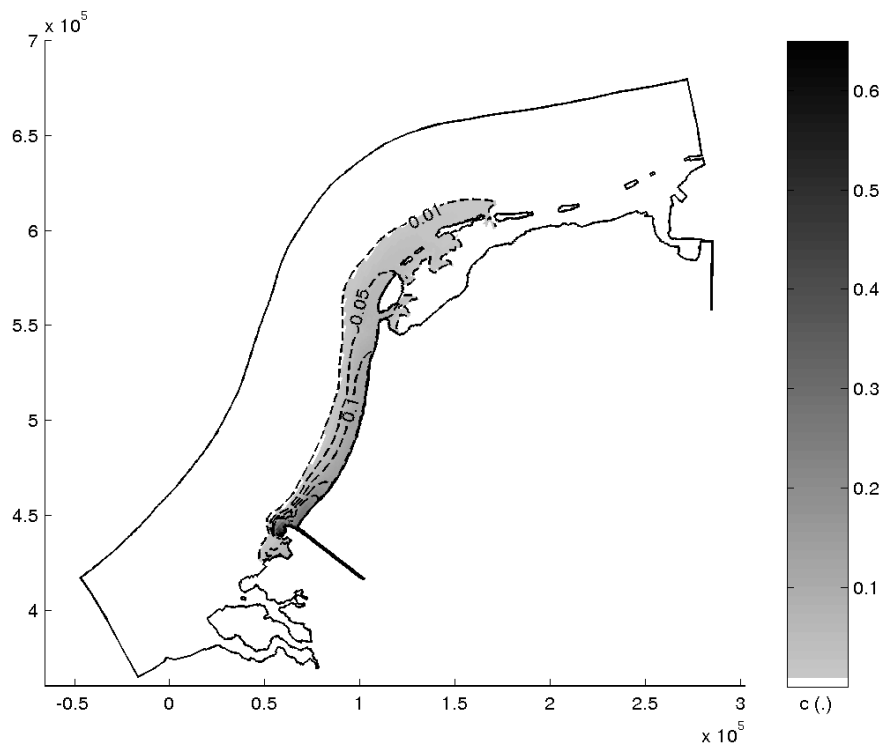


Figure 3.6: Situation after 62.5 days (WAQUA simulation, $\tau=1$ min)

First, we process the available 1 minute data set from WAQUA according to (2.49) to produce a new set with time intervals of 15 minutes. Thus, the required storage space is reduced to 11 Gb. Next, we check for the constancy condition for this new time step, which proves to be satisfactory. Finally, the same experiment of pollutant release in the river Rhine is simulated. A comparison between figure 3.7, showing the displacement of the pollutant after 62.5 days, and figure 3.4, showing the same displacement obtained however with the original 1 minute time step data, we can see that the extent of the contaminated region is very similar for both time steps. Table 3.2 also shows very similar figures, comparing the results obtained in both cases. We can therefore conclude that the use of integrated time steps is an efficient way to render the computations faster and less cumbersome.

Figure 3.8 shows the average Courant number (averaged over the whole domain) and the maximum Courant number resulting from the computations with time steps of 1 minute and 15 minutes. There is a clear similarity between the two plots if we disregard the difference in the order of magnitude (in fact the difference in order of magnitude is a factor 15). When using the 15 minute time step, we notice that occasionally the Courant number reaches some high values. This fact is observed only during the first 1000 time steps of the simulation, which counts in total 6000 time steps.

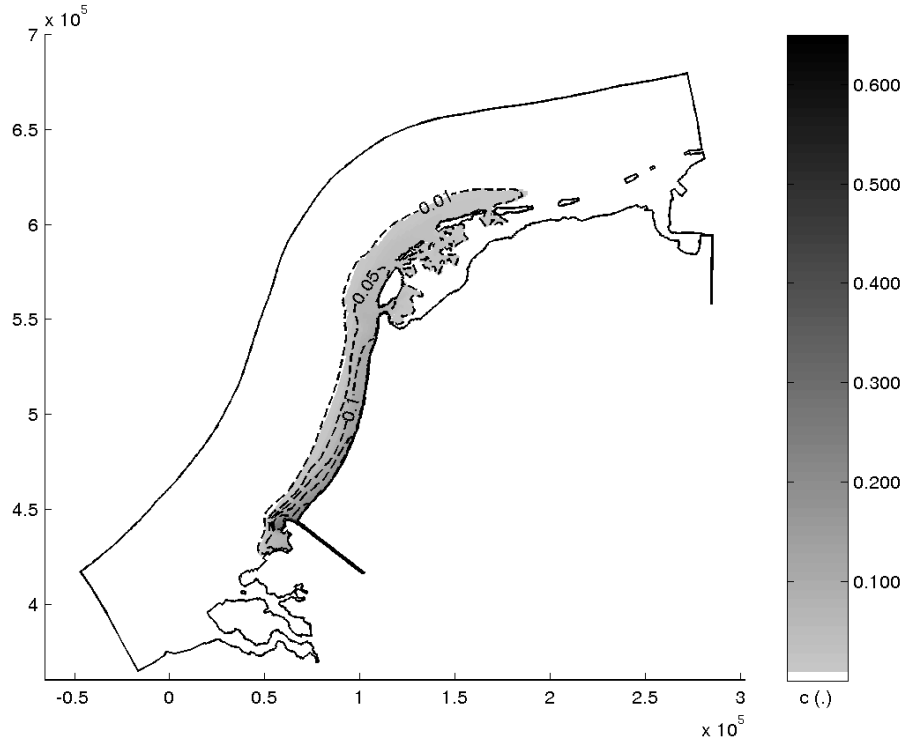
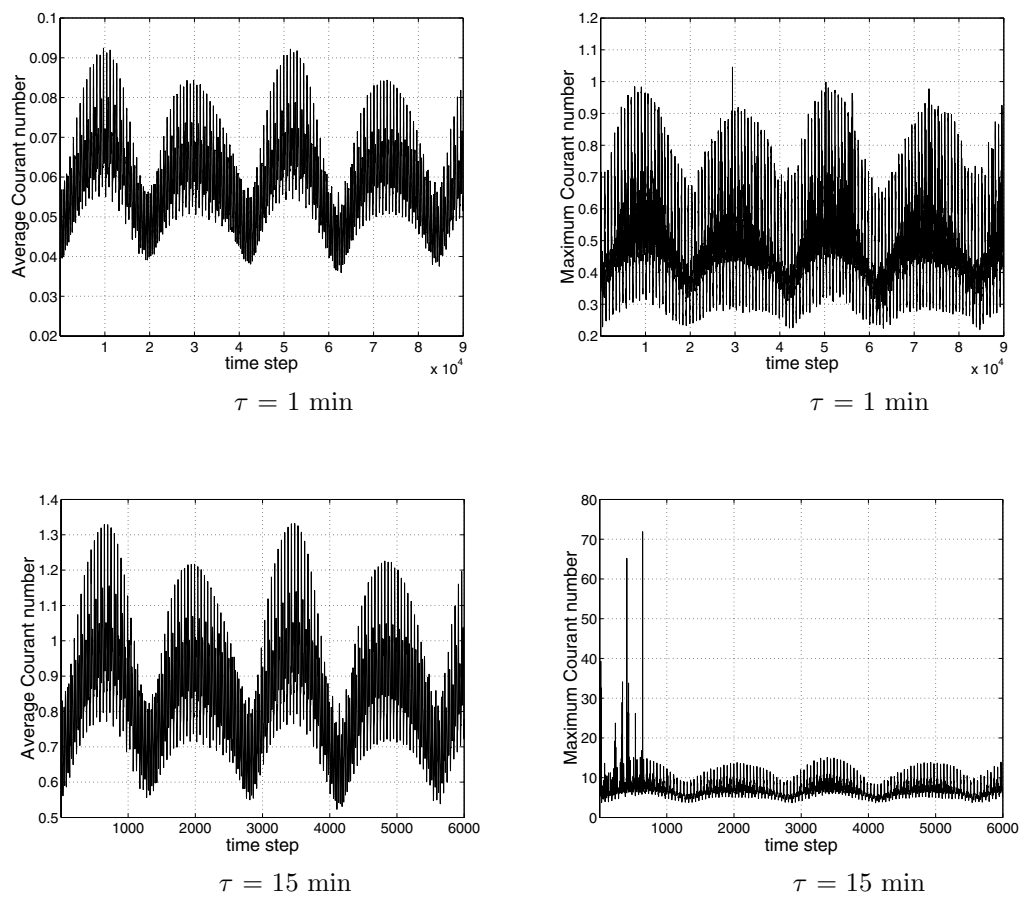


Figure 3.7: Situation after 62.5 days (continuous simulation $\tau = 15$ min)

Table 3.2: Comparison of the results of the time integrated flow data.

	$\tau = 1$ min	$\tau = 15$ min
	(Reference solution)	
Minimum concentration	0	-1.910^{-8}
Cells with negative concentration	0	1
Maximum concentration	0.5612	0.5612
Total mass of pollutant	$3.63 \cdot 10^9$	$3.62 \cdot 10^9$
$\sqrt{\sum (C_{ref} - C_{\tau=15})^2}$	-	0.589
$\sum C_{ref} - C_{\tau=15} $	-	21.63

* C_{ref} : concentration from the reference solution, $\tau=1$ min

**Figure 3.8:** Courant number

3.4 Periodic continuation

Assuming that our data has certain periodical aspects, we study whether the periodical use of a “representative” set of data can lead to acceptable results. If so, considerable reduction of stored data could be achieved.

Using a time step of 1 minute, we proceed with our model to the check of the constancy condition. The data from WAQUA is preprocessed and stored according to (2.39) for one tidal period. The computations show that the concentration profile in the domain remains unchanged until the end of the first tidal period. As soon as the second tidal period is started, this property is lost. This is due to the violation of the conservation property between the last and the first set of data while proceeding to the periodic continuation.

We proceed nevertheless with the simulation of the continuous release of a dissolved pollutant at the same location as previously. Based on the time step of 1 min and a tidal period length of 12h 45min, figure 3.9 shows the propagation of the pollutant after 62.5 days. Compared to figure 3.4, we notice that the polluted area extends further towards the north-east. However, the concentration distribution along the western coast is quite similar.

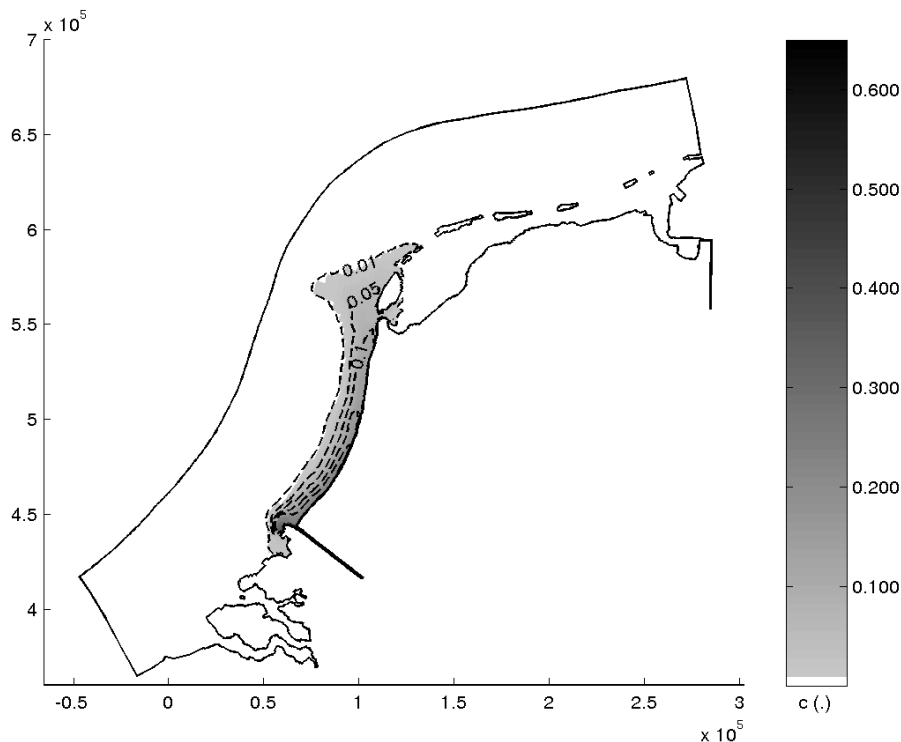


Figure 3.9: Situation after 62.5 days ($\tau = 1$ min)

Examining the solution at this date shows that some cells have a negative concentration. The largest negative values in the absolute sense are of the order of 10^{-4} . Figure 3.10 shows that starting from around the time step 33700, there is a regular decrease in the lowest value of the concentration. We can notice in figure 3.10-a

that this decrease takes place in a stepwise way in which the duration between the jumps corresponds exactly to the adopted tidal period of 765 time steps. This may indicate that the unique non-conservative computational step taking place at the end of each tidal period is the underlying cause of these negative values.

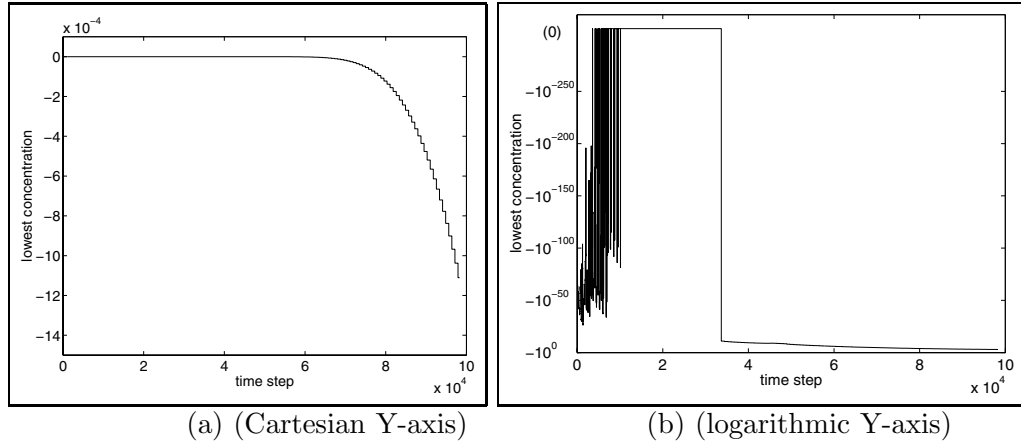


Figure 3.10: Lowest concentration ($\tau = 1$ min)

3.5 Combination of periodic continuation and time integrated flow data

In these section, we apply both techniques of time integrated flow data and periodic continuation. The same period duration is used, that of 12h 45min, i.e. 51 time steps of 15 min. The computation shows that the constancy condition is also satisfied for the entire first tidal period but this property is again lost as soon as the second tidal period is started.

The same experiment of pollutant release in the river Rhine is repeated using the 15 minute time step and the result is shown in figure 3.11. There is a clear similarity between figure 3.9 and figure 3.11, confirmed by table 3.3. This proves once again the efficiency of using the time integrated flow data.

Table 3.3: Results using periodic continuation and time integrated flow data

Time step	1 min	15 min
Minimum concentration	-0.00047	-0.00041
Cells with negative concentration	48	51
Maximum concentration	0.59106	0.59111
Total mass of pollutant	$3.538 \cdot 10^9$	$3.539 \cdot 10^9$
$\sqrt{\sum (C_{dt=1} - C_{dt=15})^2}$	-	0.214
$\sum C_{dt=1} - C_{dt=15} $	-	6.0

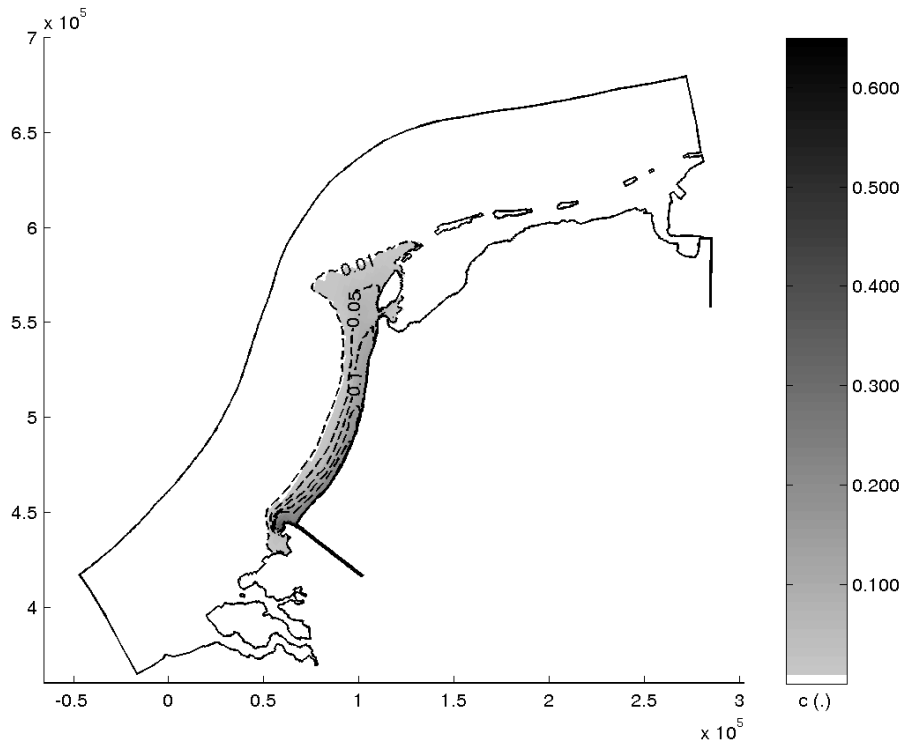


Figure 3.11: Situation after 62.5 days ($\tau = 15$ min)

Trying to investigate whether we can obtain results closer to the reference solution presented in section 3.2, we propose to compare the results obtained using different tidal period durations. All these computations are achieved using the 15 min time step. Based upon durations of 12h, 12h 15min, 12h 30min, 12h 45min, 13h and 13h 15min, figure 3.5 shows that the shore line is affected by higher concentrations if we consider longer tidal periods, whereas with shorter periods, the pollutant has a tendency to move away from the Waddenzee toward the west.

Although the period of 13h 15min seems to give a plot most resembling to the reference solution, table 3.4 shows that this is not really the case. In fact, the sum of the squared differences as well as the sum of the absolute differences are the smallest for a period length of 12h 45min. This is also the period whose total mass of pollutant at the end of the simulation is nearest to the continuous solution.

Although no clear answer can be given as to which is the best period length, the results obtained are still of interest to evaluate how close these results are to the reference solution. In fact, any further judgment fully depends upon the accuracy required.

3.6 Conclusions

For the numerical application of our 2-D solver, we chose the scenario of the propagation of pollution, originating from the river Rhine along the Dutch coast. Using the preprocessed output of WAQUA, it was first checked that our model satisfies the

Table 3.4: Comparative table of different tidal period lengths

	12:00'	12:15'	12:30'	12:45'	13:00'	13:15'
Total mass	$3.24 \cdot 10^9$	$3.48 \cdot 10^9$	$3.31 \cdot 10^9$	$3.53 \cdot 10^9$	$3.43 \cdot 10^9$	$3.38 \cdot 10^9$
Highest C	0.529	0.560	0.552	0.591	0.678	0.678
Lowest C	$-2.9 \cdot 10^{-6}$	$-7.3 \cdot 10^{-6}$	$-6.2 \cdot 10^{-5}$	$-4.2 \cdot 10^{-4}$	$-4.7 \cdot 10^{-3}$	$-1.5 \cdot 10^{-2}$
Negative C	29	43	55	51	52	59
$\sqrt{\sum (C - C_{ref})^2}$ *	3.8	2.9	2.1	1.6	2.3	3.7
$\sum C - C_{ref} $ *	229.1	176.1	142.1	110.4	129.8	191.0

* C_{ref} : concentration from the reference solution

constancy condition. Next, a full simulation of a tracer release from the Nieuwe Waterweg was carried out with a time step of 1 minute and a duration of two months. The results were compared to those from the transport solver of WAQUA, showing good correspondence. The result of this long term simulation was adopted as our reference solution.

The numerical experiments show that the use of time integrated flow data is a very efficient way to reduce the amount of input data. Using periodic continuation, the constancy condition is satisfied only during the first tidal period. However, the unique non-conservative computational step at the beginning of the second period affects this property. A comparative study was conducted in order to evaluate the results obtained using different tidal period lengths. We noticed that this parameter is an important factor that may affect the solution considerably.

Up to this stage, we have assumed the overall phenomena of the transport process to be two-dimensional and have adopted a depth-averaged approach. Although there are a number of factors which support this hypothesis, it is still sometimes claimed that three-dimensional aspect should be considered. For these reason, the next two chapters of this thesis are dedicated to building and testing a 3-D version of the transport solver in order to provide arguments to answer this open question.

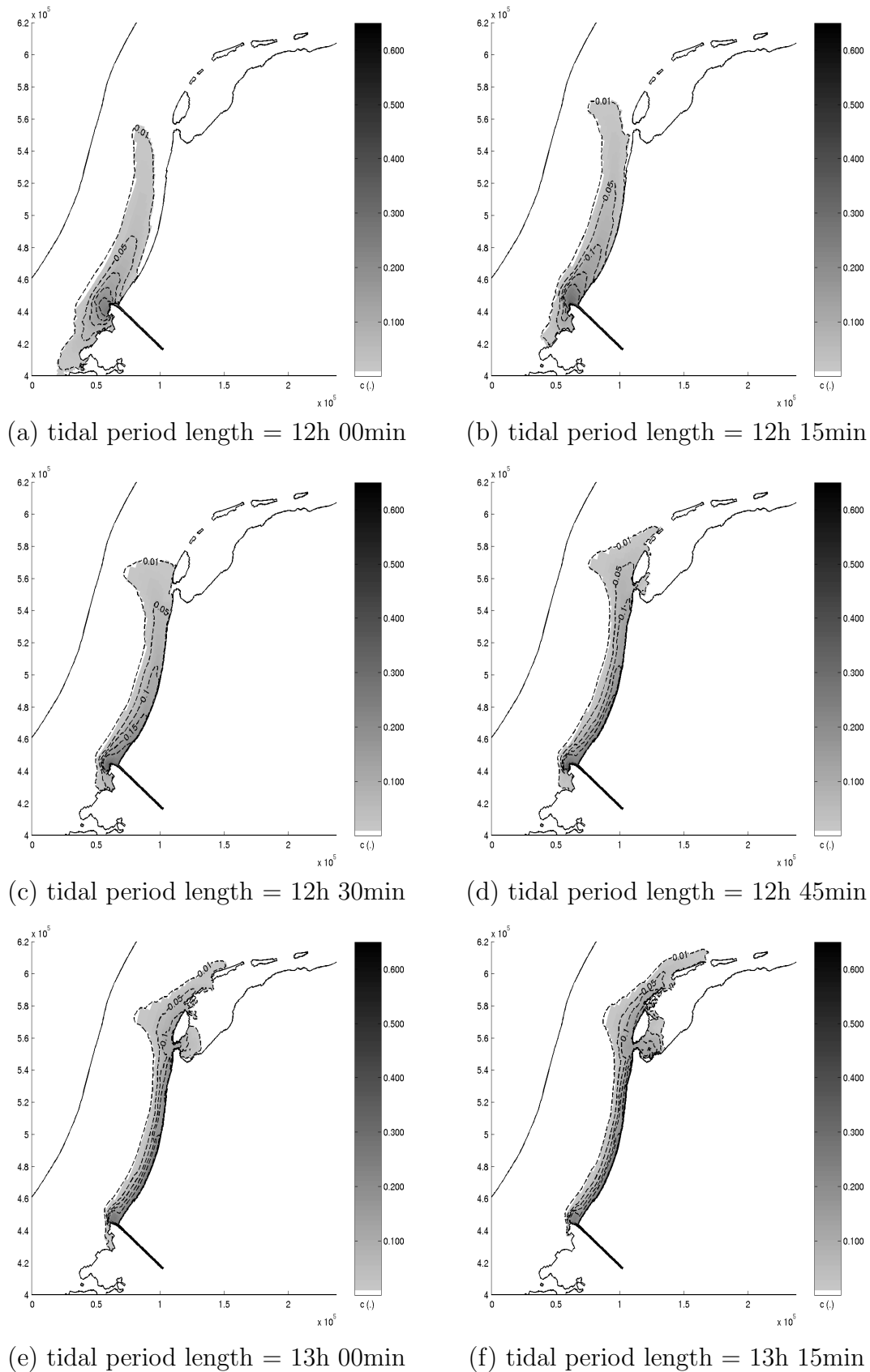


Figure 3.12: Simulations using different tidal period lengths.

Chapter 4

Development of a 3-D transport solver

4.1 Introduction

Predicting the concentration of a particular contaminant in a water body is a major objective of many environmental studies. It is an important step which allows for the evaluation and design of any protection plans. Numerical models are by far the most powerful tools that can be used for this purpose. Especially convenient for large scale systems, numerical models help in reducing the use of water quality control techniques such as setting up monitoring programs, which can be very laborious. At least, these can be limited to acquiring data for testing and validating the numerical models.

Previous chapters of this thesis dealt with the 2-D depth-averaged transport solver. The governing equations and the implementation details of the finite volume solver were presented and the constructed model was used to simulate the propagation of contaminants along the Dutch coast. In general, depth-averaged models resolve differences in the horizontal directions only and are commonly used for relatively shallow water bodies. Although they remain adequate for many applications, there are still situations where the vertical distribution of the concentration becomes significant and the use of 3-D models becomes necessary. The limits of applicability of the 2-D and the 3-D models cannot be clearly defined, and there will always remain a significant area of uncertainty about the applicability of each model.

In this chapter, we intend to build a new 3-D transport solver based upon the experience gained with the 2-D solver. The same combination of finite volume method, semi-implicit time stepping and horizontal unstructured grid is adopted. In the vertical direction, the grid is structured and consists of the desired number of layers. The numerical application of the 3-D version is presented in the next chapter.

4.2 Governing equation

For convenience, we start by recalling the governing equation of the movement of a tracer in three dimensions. This equation is expressed by the advection-diffusion equation which, for negligible density changes in the ambient conditions, has the form

$$\frac{\partial c}{\partial t} + \frac{\partial uc}{\partial x} + \frac{\partial vc}{\partial y} + \frac{\partial wc}{\partial z} = \frac{\partial}{\partial x}(D_h \frac{\partial c}{\partial x}) + \frac{\partial}{\partial y}(D_h \frac{\partial c}{\partial y}) + \frac{\partial}{\partial z}(D_v \frac{\partial c}{\partial z}) + S \quad (4.1)$$

With this formulation, we assume that the diffusion tensor reduces to two scalar values, D_h and D_v , which are the diffusion coefficients in the horizontal and vertical directions respectively.

4.3 3-D grid

The choice of the type of grids to be used is an important step in the modeling process. For a three-dimensional grid, we need to make choices for the horizontal direction and for the vertical direction. In the horizontal direction, the use of unstructured grids presents, as for the 2-D model, the advantage of a great flexibility in the discretization of domains with complex geometries and consequently this choice is adopted also for the 3-D case. For the discretization of the vertical direction, a structured type is preferred. The physical domain is divided into a fixed number of layers. This same technique is also used in TRIWAQ which presents the advantage of allowing the use of TRIWAQ output flow parameters in our numerical application with a minimum of interpolations. The geometry in the vertical direction is therefore straightforward and the use of unstructured grids can not be justified. Thus, the horizontal geometry of the grid remains unchanged whereas the layer interfaces are allowed to move in the vertical direction. This type of discretization is convenient to extend any 2-D grid to the third dimension.

Consequently, a given cell has the same x and y vertex coordinates as the cells located in the same column, as shown in figure 4.1. This makes it possible to identify the cells in the computational domain by extending the 2-D numbering; any cell is fully identified by giving the 2-D identifier together with the layer number. The numbering of layers is such that layer 1 is the topmost layer and layer K is the lowest for a vertical discretization consisting of K layers. See figure 4.2. In the vertical direction, a layer with index k is located between layer interfaces k and $k - 1$, the interface $k - 1$ being the top one.

Similarly to the 2-D case, the unit element or volume of the discrete system is called a cell. However, in the 3-D case, a distinction needs to be made between faces and edges. The term face now designates the vertical interface area separating two cells, whereas the term edge designates the line connecting two nodes as represented by figure 4.3. In a quadrilateral 3-D grid, an edge can be shared between up to four cells.

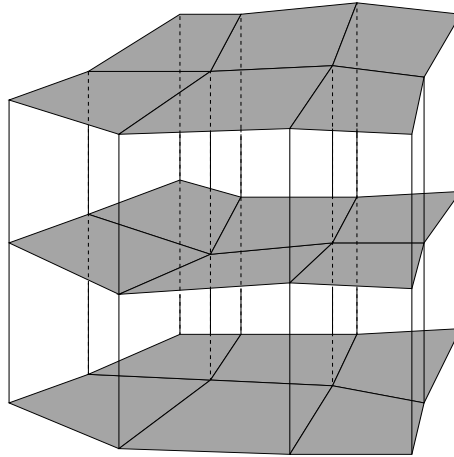


Figure 4.1: 3-D grid

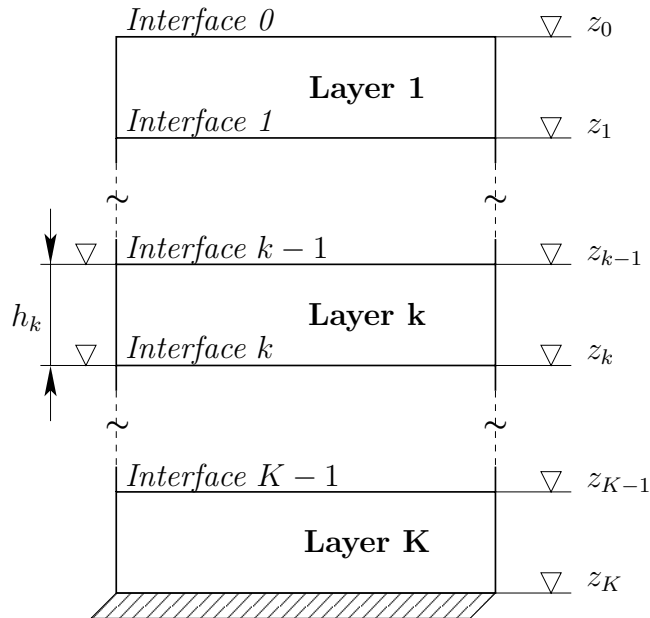


Figure 4.2: Layer depth and layer elevation

4.4 Layer-averaged transport equation

The transport equation (4.1) applies for every unit element or cell of the adopted spatial discretization. In order to derive the discrete formulation, we first make the assumption that the vertical distribution within one cell is not significant anymore and therefore the transport equation can be integrated in the vertical direction. For this purpose, we define the layer averaged concentration of a given cell from layer k to be

$$c_k = \frac{1}{h_k} \int_{z_k}^{z_{k-1}} c \, dz, \quad (4.2)$$

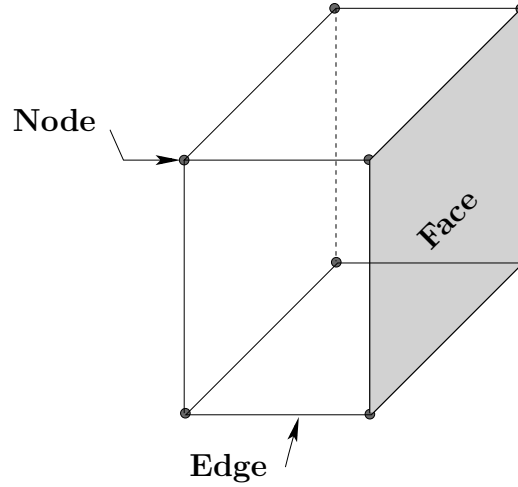


Figure 4.3: Definition of grid elements

where h_k is the thickness of the cell defined as the difference between the elevations of the central points of the top and bottom faces (corresponding to the water level point)

$$h_k = z_{k-1}(wl) - z_k(wl) . \quad (4.3)$$

For the finite volume approach, we assume that c_k is constant within each cell and is represented by the value at the cell center.

Then, we go through the terms of (4.1) one by one and compute the average over the layer depth.

Time derivative term

Let us consider again a cell from layer k . The Leibniz rule is used to rewrite the integral of the time derivative between the bottom and top interfaces of the cell.

$$\int_{z_k}^{z_{k-1}} \frac{\partial c}{\partial t} dz = \frac{\partial}{\partial t} \int_{z_k}^{z_{k-1}} c dz - \left[c \frac{\partial z}{\partial t} \right]_{z_k}^{z_{k-1}} = \frac{\partial h_k c_k}{\partial t} - \left[c \frac{\partial z}{\partial t} \right]_{z_k}^{z_{k-1}} . \quad (4.4)$$

Horizontal advection term

Similarly, the Leibniz rule applied to the horizontal advection term in the x direction gives

$$\int_{z_k}^{z_{k-1}} \frac{\partial u c}{\partial x} dz = \frac{\partial}{\partial x} \int_{z_k}^{z_{k-1}} u c dz - \left[u c \frac{\partial z}{\partial x} \right]_{z_k}^{z_{k-1}} .$$

One can easily check that the first term of the rhs can be rewritten as

$$\frac{\partial}{\partial x} \int_{z_k}^{z_{k-1}} u c dz = \frac{\partial h_k u_k c_k}{\partial x} + \frac{\partial}{\partial x} \int_{z_k}^{z_{k-1}} (u - u_k)(c - c_k) dz .$$

A similar expression will hold for the horizontal advection term in the y direction if u and v are interchanged. Therefore, we can write

$$\int_{z_k}^{z_{k-1}} \left(\frac{\partial uc}{\partial x} + \frac{\partial vc}{\partial y} \right) dz = \nabla \cdot (h_k \mathbf{u}_k c_k) - [\mathbf{u} \cdot \nabla z c]_{z_k}^{z_{k-1}} + Disp, \quad (4.5)$$

where the term $Disp$ represents the quantity

$$\nabla \cdot \left[\int_{z_k}^{z_{k-1}} (u - u_k)(c - c_k) dz \right]_{z_k}^{z_{k-1}},$$

which may be regarded as a dispersion term. $\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$ is the velocity vector and $\mathbf{u}_k = \begin{pmatrix} u_k \\ v_k \end{pmatrix}$ is the layer averaged velocity vector.

Vertical advection term

This term is integrated in a straightforward manner to give

$$\int_{z_k}^{z_{k-1}} \frac{\partial wc}{\partial z} dz = [w c]_{z_k}^{z_{k-1}}. \quad (4.6)$$

Horizontal diffusion term

In order to compute the integral of this term, the Leibniz rule is applied twice. The x direction is considered first

$$\begin{aligned} \int_{z_k}^{z_{k-1}} \frac{\partial}{\partial x} \left(D_h \frac{\partial c}{\partial x} \right) dz &= \frac{\partial}{\partial x} \int_{z_k}^{z_{k-1}} D_h \frac{\partial c}{\partial x} dz - \left[D_h \frac{\partial c}{\partial x} \frac{\partial z}{\partial x} \right]_{z_k}^{z_{k-1}} \\ &= \frac{\partial}{\partial x} \left(D_h \frac{\partial}{\partial x} \left(\int_{z_k}^{z_{k-1}} c dz \right) - \left[D_h c \frac{\partial z}{\partial x} \right]_{z_k}^{z_{k-1}} \right) - \left[D_h \frac{\partial c}{\partial x} \frac{\partial z}{\partial x} \right]_{z_k}^{z_{k-1}} \\ &= \frac{\partial}{\partial x} \left(D_h \frac{\partial h_k c_k}{\partial x} \right) - \left[\frac{\partial}{\partial x} \left(D_h c \frac{\partial z}{\partial x} \right) \right]_{z_k}^{z_{k-1}} - \left[D_h \frac{\partial c}{\partial x} \frac{\partial z}{\partial x} \right]_{z_k}^{z_{k-1}}. \end{aligned} \quad (4.7)$$

A similar equation can be written for the y direction to end up with the following expression for the horizontal diffusion term

$$\begin{aligned} \int_{z_k}^{z_{k-1}} \left(\frac{\partial}{\partial x} \left(D_h \frac{\partial c}{\partial x} \right) + \frac{\partial}{\partial y} \left(D_h \frac{\partial c}{\partial y} \right) \right) dz = \\ \nabla \cdot (D_h \nabla (h_k c_k)) - [\nabla \cdot (D_h c \nabla z)]_{z_k}^{z_{k-1}} + [D_h \nabla c \cdot \nabla z]_{z_k}^{z_{k-1}}. \end{aligned} \quad (4.8)$$

Vertical diffusion term

$$\int_{z_k}^{z_{k-1}} \frac{\partial}{\partial z} \left(D_v \frac{\partial c}{\partial z} \right) dz = \left[D_v \frac{\partial c}{\partial z} \right]_{z_k}^{z_{k-1}}. \quad (4.9)$$

Source term

$$\int_{z_k}^{z_{k-1}} S dz = S_k h_k. \quad (4.10)$$

Putting all the terms back together again, the layer integrated transport equation can now be written as follows

$$\begin{aligned} \frac{\partial h_k c_k}{\partial t} + \nabla \cdot (h_k \mathbf{u}_k c_k) + \left[(w - \mathbf{u} \cdot \nabla z - \frac{\partial z}{\partial t}) c \right]_{z_k}^{z_{k-1}} + Disp = \\ \nabla \cdot (D_h \nabla (h_k c_k)) + \nabla \cdot [D_h c \nabla z]_{z_k}^{z_{k-1}} + [D_h \nabla c \cdot \nabla z]_{z_k}^{z_{k-1}} + \left[D_v \frac{\partial c}{\partial z} \right]_{z_k}^{z_{k-1}}. \end{aligned} \quad (4.11)$$

One can define ω as the vertical velocity of the flow with respect to the moving interface. ω and w are then related by the following expression

$$\omega = w - \mathbf{u} \cdot \nabla z - \frac{\partial z}{\partial t}. \quad (4.12)$$

Adopting a similar notation as in the 2-D case, i.e. $\mathbf{f}_k = h_k \mathbf{u}_k c_k$ and $\mathbf{g}_k = D_h \nabla (h_k c_k)$, and substituting in (4.11), this latter equation can be rewritten as follows

$$\begin{aligned} \frac{\partial h_k c_k}{\partial t} + \nabla \cdot \mathbf{f}_k + [\omega c]_{z_k}^{z_{k-1}} + Disp = \\ \nabla \cdot \mathbf{g}_k + \nabla \cdot [D_h c \nabla z]_{z_k}^{z_{k-1}} + [D_h \nabla c \cdot \nabla z]_{z_k}^{z_{k-1}} + \left[D_v \frac{\partial c}{\partial z} \right]_{z_k}^{z_{k-1}} + S_k h_k. \end{aligned} \quad (4.13)$$

This is the differential form of the three-dimensional advection-diffusion equation that we propose to solve.

4.5 3-D finite volume formulation

The next step towards the finite volume implementation consists in writing the area integral form of (4.13) and defining the numerical approximation of each term. Similarly to the previous section, we go through all the terms of the equation one by one. When applicable, Green's formula is applied to transform the area integral of divergence terms to integrals along the boundaries.

The terminology used here for the different terms of the layer averaged equation may not correspond exactly to terminology of the previous section since some reshuffling has taken place.

Time derivative term

$$\int_A \frac{\partial h_k c_k}{\partial t} dA = A \frac{\partial h_k c_k}{\partial t} \approx A \frac{dhc}{dt} \quad (4.14)$$

Horizontal advection term

$$\int_A \nabla \cdot \mathbf{f}_k dA = \sum_e \int_{\sigma_e} \mathbf{f}_k \cdot \mathbf{n}_e d\sigma_e = \sum_e (f_{e,k} l_e) \quad (4.15)$$

Index e indicates that the value of the variable is to be evaluated at the faces. The approximation of the numerical advective flux $f_{e,k}$ is dealt with in the next section.

Vertical advection term

$$\int_A [\omega c]_{z_k}^{z_{k-1}} dA = A [\omega c]_{z_k}^{z_{k-1}} \quad (4.16)$$

Horizontal diffusion term

$$\int_A \nabla \cdot \mathbf{g}_k dA = \sum_e \int_{\sigma_e} \mathbf{g}_k \cdot \mathbf{n}_e d\sigma_e = \sum_e (g_{e,k} l_e) \quad (4.17)$$

Vertical diffusion term

$$\int_A \left[D_v \frac{\partial c}{\partial z} \right]_{z_k}^{z_{k-1}} dA = A D_v \left(\left. \frac{\partial c}{\partial z} \right|_{z_{k-1}} - \left. \frac{\partial c}{\partial z} \right|_{z_k} \right) \quad (4.18)$$

Source term

$$\int_A S_k h_k dA = A S_k h_k \quad (4.19)$$

Remaining terms

$$\int_A \nabla \cdot [D_h c \nabla z]_{z_k}^{z_{k-1}} dA = \sum_e \int_{\sigma_e} [D_h c \nabla z]_{z_k}^{z_{k-1}} \cdot \mathbf{n}_e d\sigma_e \quad (4.20)$$

$$\int_A [D_h \nabla c \cdot \nabla z]_{z_k}^{z_{k-1}} dA \approx A [D_h \nabla c \cdot \nabla z]_{z_k}^{z_{k-1}} \quad (4.21)$$

For the time being, the area integral of the term $Disp$ in (4.13) will simply be assumed to be part of the diffusion represented by the coefficients D_h and D_v .

4.6 Flux approximation

The approximation of fluxes is a major step in a finite volume method and therefore needs to be implemented efficiently. In the 3-D case, we need to distinguish between fluxes at vertical interfaces and at horizontal interfaces. The objective is to avoid redundancy in the computations. Therefore, fluxes through vertical interfaces are computed using loops over faces and horizontal ones using loops over cells.

On the other hand, a better approximation of fluxes can be obtained by first examining the locations of different flow variables and using expressions which minimize the use of interpolated values. As mentioned previously, in our case flow data are obtained using the software package TRIWAQ. More details about these data are given in the next section.

4.6.1 Available data from TRIWAQ

TRIWAQ, the 3-D version of WAQUA presented earlier in chapter 2, solves the shallow water equations in three dimensions. It is therefore generally used to calculate the vertical distribution of the flow parameters, notably for accuracy constraints. It has a similar input structure as WAQUA except for the description of the layers. The mathematical details of TRIWAQ can be found in [18].

Modeled systems are represented in TRIWAQ by quadrilateral unit elements forming rectilinear, curvilinear or spherical grids. We are interested in curvilinear grids which have the advantage of giving a better geometrical fit to domains with complex shapes than rectilinear grids. Spherical grids are adapted only to very large systems where the effect of the quasi-spherical shape of the earth becomes important.

Grids used by TRIWAQ are staggered, that is, different variables are located in different locations. The velocity vectors are computed at the center of the (vertical) faces and only the magnitude of the velocity vectors is given, the direction being always normal to the faces. Layer elevations are very important to ensure a mass conservative computation. In TRIWAQ, these are defined at every layer interface at five locations: the central point of the layer interface and the midpoints of the edges as shown in figure 4.4. Water depths at cell centers and at faces are computed by subtracting the corresponding elevations.

Finally, for comparison purposes, we also use the transport solver of TRIWAQ later on. Similar to our solver, concentrations are located at the centroids of cells in TRIWAQ.

4.6.2 Horizontal advection flux

Upwind methods have the advantage of adequately accounting for the direction of information in the domain. Here, the Enquist-Osher function is used to approximate the numerical advective flux f_e . For a given face e from layer k , this equation writes

$$f_e(c_{e,k}^L, c_{e,k}^R) = \frac{u_{e,k} + |u_{e,k}|}{2} h_{e,k} c_{e,k}^L + \frac{u_{e,k} - |u_{e,k}|}{2} h_{e,k} c_{e,k}^R. \quad (4.22)$$

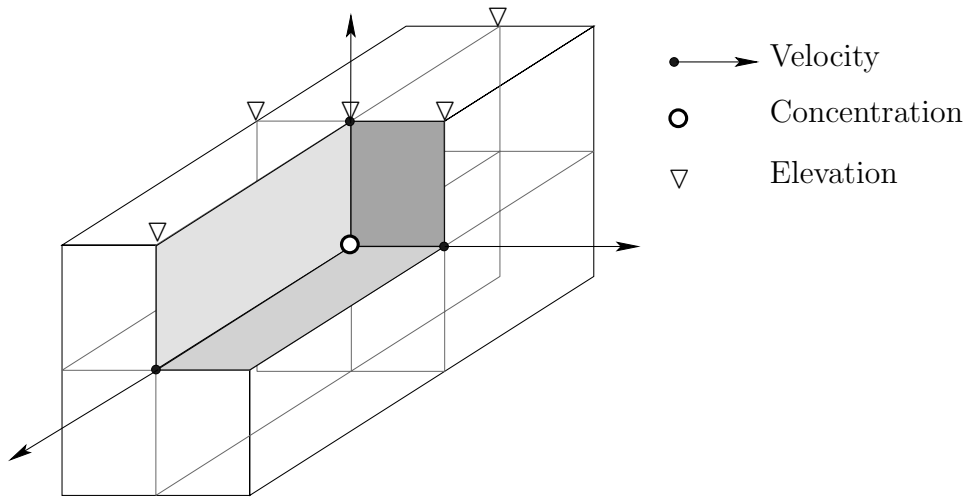


Figure 4.4: Location of the calculation points of different variables in a 3-D grid volume.

In this equation, $c_{e,k}^L$ and $c_{e,k}^R$ are the values of the concentration at the central point of face e . $u_{e,k}$ is the velocity magnitude, and $h_{e,k}$ is the depth or thickness of layer k , both at the same point.

Higher-order accuracy of the scheme is obtained through reconstruction of the values of the concentration at the center of the faces. For a given face, this value is computed based upon the concentration at the cells forming the support molecule around the face. The support molecule is similar to the one in figure 2.4 and is formed by the two cells sharing the face concerned (primary cells), and also the cells sharing faces with these two primary cells (secondary cells). All the cells forming the molecule belong to the same layer as the face.

With the adoption of this formulation, the computation of the advective flux in three dimensions is equivalent to repeating the 2-D procedure as many times as the number of layers. The corresponding programming task is therefore made easier; a do-loop is built over the main routine computing the 2-D flux to repeat the same instructions for each layer without having to modify the lower level routines.

4.6.3 Vertical advection term

The vertical advection term involves values of the concentration at the center of the horizontal interfaces between cells

$$A [\omega c]_{z_k}^{z_{k-1}} = A (\omega_{k-1} c(z_{k-1}) - \omega_k c(z_k)) . \quad (4.23)$$

These values are not readily available since the concentration values are located at the centroids of the unit volumes. A second-order approximation of this value can be obtained by averaging the concentration of the top and bottom cells sharing the interface.

$$c(z_{k-1}) = \frac{h_k c_{k-1} + h_{k-1} c_k}{h_{k-1} + h_k} \quad \text{and} \quad c(z_k) = \frac{h_{k+1} c_k + h_k c_{k+1}}{h_k + h_{k+1}} . \quad (4.24)$$

Vertical velocities, however, are computed and thus available precisely at the requested location. This discretization results in the final following expression

$$A [\omega c]_{z_k}^{z_{k-1}} = A \left\{ \omega_{k-1} \frac{h_k}{h_{k-1} + h_k} c_{k-1} + \left(\omega_{k-1} \frac{h_{k-1}}{h_{k-1} + h_{k+1}} - \omega_k \frac{h_{k+1}}{h_k + h_{k+1}} \right) c_k - \omega_k \frac{h_k}{h_k + h_{k+1}} c_{k+1} \right\} \quad (4.25)$$

The vertical velocity at the top-most horizontal interface is equal to zero. The same applies to the bottom one. Therefore, the corresponding discrete forms for the vertical advection term are adjusted and have the following forms

$$A [\omega c]_{z_1}^{z_0} = -\frac{A \omega_1}{h_1 + h_2} \{ h_2 c_1 + h_1 c_2 \} \quad (4.26)$$

and

$$A [\omega c]_{z_K}^{z_{K-1}} = \frac{A \omega_{K-1}}{h_{K-1} + h_K} \{ h_K c_{K-1} + h_{K-1} c_K \} \quad (4.27)$$

4.6.4 Horizontal diffusion flux

For the discretization of the diffusive flux, the gradient $\nabla(h_k c_k)$ needs to be evaluated along vertical cell interfaces. Of the available methods suitable for unstructured grids, we use the one based on the divergence theorem.

First, there is a need to define an area around the face in which the divergence theorem will be applied. We propose to use the shadow volume presented in figure 2.5, again similarly to the 2-D case. The numerical horizontal diffusive flux through a given face in layer k is approximated by

$$g_{e,k} = D_h \nabla c_k \cdot \mathbf{n}_e = \frac{(D_h \mathbf{n}_{ab} \cdot \mathbf{n}_e)(c_{2,k} - c_{1,k}) + (D_h \mathbf{n}_{12} \cdot \mathbf{n}_e)(c_{a,k} - c_{b,k})}{2A_{1a2b}}. \quad (4.28)$$

In this equation, only the values of the concentration are layer dependent. All normal vectors \mathbf{n}_* as well as the area A_{1a2b} depend only on the grid horizontal geometry, which is the same for all layers.

4.6.5 Vertical diffusive flux

The vertical diffusive term in (4.18) requires computation of the gradient of the concentration in the vertical direction at the center of the horizontal cell interfaces. Considering the notation in figure 4.5, this gradient is approximated using Taylor series expansion. We can write that

$$c_{k-1} = c(z_{k-1}) + \frac{1}{2} h_{k-1} \left. \frac{\partial c}{\partial z} \right|_{z_{k-1}} + \frac{1}{3!} h_{k-1}^2 \left. \frac{\partial^2 c}{\partial z^2} \right|_{z_{k-1}} + \mathcal{O}(h^3) \quad (4.29)$$

$$c_k = c(z_{k-1}) - \frac{1}{2} h_k \left. \frac{\partial c}{\partial z} \right|_{z_{k-1}} + \frac{1}{3!} h_k^2 \left. \frac{\partial^2 c}{\partial z^2} \right|_{z_{k-1}} + \mathcal{O}(h^3), \quad (4.30)$$

where c_k represents the concentration at the center of the cell forming the layer k , and $c(z_{k-1})$ is the concentration at the center of interface $k - 1$.

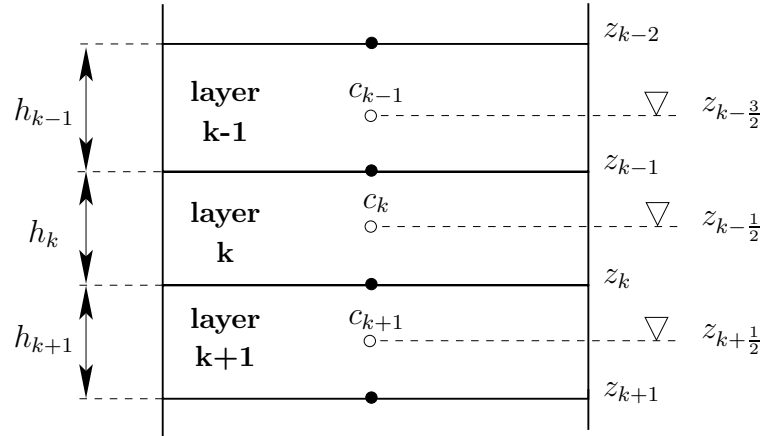


Figure 4.5: Variables used for the computation of the vertical diffusive flux

Provided that there are no abrupt changes in the vertical spacing, subtracting these two equations provides a second-order approximation of the required partial derivative

$$\left. \frac{\partial c}{\partial z} \right|_{z_{k-1}} \approx 2 \frac{c_{k-1} - c_k}{h_{k-1} + h_k}. \quad (4.31)$$

Writing a similar expression for the partial derivative at interface k , we can now write the approximation of the vertical diffusive flux as

$$A D_v \left(\left. \frac{\partial c}{\partial z} \right|_{z_{k-1}} - \left. \frac{\partial c}{\partial z} \right|_{z_k} \right) = 2 A D_v \left(\frac{c_{k-1} - c_k}{h_{k-1} + h_k} - \frac{c_k - c_{k+1}}{h_k + h_{k+1}} \right). \quad (4.32)$$

All the terms present in this expression are available at the required locations; both the concentrations and the layer thickness are found at the cell centers.

4.6.6 Remaining terms

First, we start with the term in (4.20). The integral in this term is approximated using the mid-point rule

$$\sum_e \int_{\sigma_e} [D_h c \nabla z]_{z_k}^{z_{k-1}} \cdot \mathbf{n}_e d\sigma_e = \sum_e [D_h c_e \nabla z_e]_{z_k}^{z_{k-1}} \cdot \mathbf{n}_e \quad (4.33)$$

Recall that \mathbf{n}_e is the normal vector to face e with a magnitude equal to the length of the face. Adopting the notation presented in figure 4.6, we can develop the flux through face e as

$$D_h \{c_e(z_{k-1}) \nabla z_{e,k-1} - c_e(z_k) \nabla z_{e,k}\} \cdot \mathbf{n}_e, \quad (4.34)$$

where $z_{e,k}$ is the elevation of the midpoint of the bottom edge of face e . The concentration $c_e(z_k)$ can be taken as an average of $c_{e,k}$ and $c_{e,k+1}$, leading to the following expression

$$D_h \left\{ \frac{c_{e,k-1} + c_{e,k}}{2} \nabla z_{e,k-1} - \frac{c_{e,k} + c_{e,k+1}}{2} \nabla z_{e,k} \right\} \cdot \mathbf{n}_e. \quad (4.35)$$

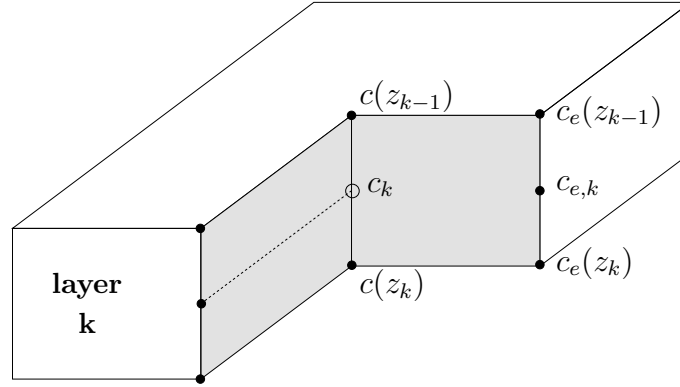


Figure 4.6: Convention for the notation of the concentration at different locations

If we consider the primary cells 1 and 2 sharing face e (as shown for example in figure 2.5), the concentration $c_{e,k}$ can again be computed as an average of the two known neighboring concentrations $c_k(1)$ and $c_k(2)$. Therefore, (4.35) becomes

$$D_h \left\{ \frac{\nabla z_{e,k-1}}{2} \frac{c_{k-1}(1) + c_{k-1}(2)}{2} + \frac{\nabla z_{e,k-1} - \nabla z_{e,k}}{2} \frac{c_k(1) + c_k(2)}{2} - \frac{\nabla z_{e,k}}{2} \frac{c_{k+1}(1) + c_{k+1}(2)}{2} \right\} \cdot \mathbf{n}_e. \quad (4.36)$$

In this expression, all the required concentrations are located at known computational points. What remains, then, is to estimate the gradient terms at the top and bottom edges of face e . We again use the Green-Gauss technique over the shadow area ($P_1P_aP_2P_b$) presented in figure 2.5. For the gradient $\nabla z_{e,k}$, we can write

$$\begin{aligned} \nabla z_{e,k} &= \frac{1}{A_{1a2b}} \int_{A_{1a2b}} \nabla z_{e,k} dA = \frac{1}{A_{1a2b}} \int_{\partial A_{1a2b}} z_{e,k} \mathbf{n} d\partial A \\ &\approx \frac{1}{A_{1a2b}} (z_{1a,k} \mathbf{n}_{1a} + z_{a2,k} \mathbf{n}_{a2} + z_{b1,k} \mathbf{n}_{b1} + z_{2b,k} \mathbf{n}_{2b}), \end{aligned}$$

where $\mathbf{n}_{..}$ are the normal vectors to the boundaries of area ($P_1P_aP_2P_b$) defined in a similar way as in equation (2.16). z values are interpolated as follows

$$z_{1a,k} = \frac{z_{1,k} + z_{a,k}}{2}; \quad z_{a2,k} = \frac{z_{a,k} + z_{2,k}}{2}; \quad z_{b1,k} = \frac{z_{b,k} + z_{1,k}}{2}; \quad z_{2b,k} = \frac{z_{2,k} + z_{b,k}}{2} \quad (4.37)$$

$z_{1,k}$ and $z_{2,k}$ are readily available from the TRIWAQ output. However, $z_{a,k}$ and $z_{b,k}$ need to be interpolated from neighboring values

$$z_{a,k} = \frac{z_{4,k} + z_{5,k}}{2}; \quad z_{b,k} = \frac{z_{3,k} + z_{6,k}}{2}. \quad (4.38)$$

$\nabla z_{e,k}$ can now be written as follows

$$\nabla z_{e,k} = \frac{1}{2 A_{1a2b}} \{ (z_{2,k} - z_{1,k}) \mathbf{n}_{ab} + (z_{a,k} - z_{b,k}) \mathbf{n}_{12} \}. \quad (4.39)$$

A similar expression can be written for $\nabla z_{e,k-1}$, which makes it possible to compute all the terms in (4.36). The final remark with respect to this term is that it is face-related and therefore its computation will be conducted by looping over faces.

Term (4.21) is the last term in the discrete transport equation which needs to be approximated. This term involves the gradient of the concentration c and that of the elevation z at the horizontal cell interfaces. The Green-Gauss technique is used for their approximation based upon the shadow volume shown in figure 4.7. The concentrations at the horizontal interface (at points 1, 2, 3, 4 and 5 in figure

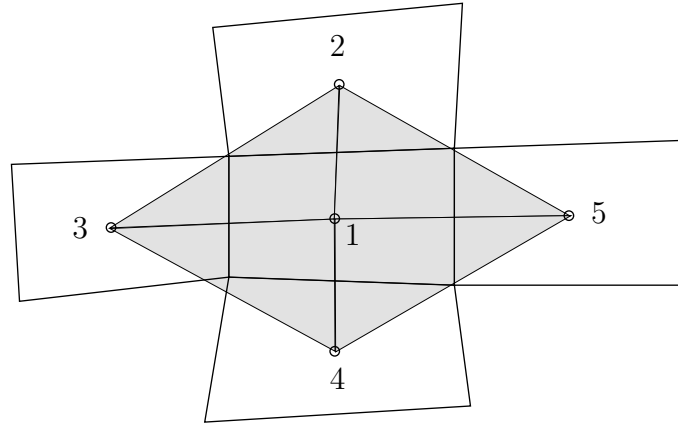


Figure 4.7: Shadow volume for the computation of $[\nabla c \nabla z]_{z_k}^{z_{k-1}}$

4.7) are interpolated from the known values on top and underneath each point. The computation of this flux is achieved by looping over cells.

4.7 The semi discrete system

All the fluxes through vertical and horizontal faces involved in our discrete transport equation are now expressed under a discrete formulation. For a given cell, figure 4.8 shows the 3-D support molecule or stencil resulting from the adopted discretization of the different terms. The cell itself is shown in gray and the different signs in this figure indicate the term through which the cells of the molecule are involved in the final equation.

Taking together the equations of the whole grid, we end up with the semi-discrete system of the form

$$L \frac{dhc}{dt} = Pc. \quad (4.40)$$

The matrix L is a diagonal matrix which contains the values of the areas of the cells. Matrix P is a sparse matrix whose entries are independent of the concentration c except for the entries resulting from the horizontal advection. This dependency on c results from the limiting function ψ used during the reconstruction procedure.

Looking again at the 3-D molecule of the given cell in figure 4.8, to each cell in this figure containing at least one sign, there corresponds a non-zero element in the row of the matrix of that cell. This means that each row of P has at most 23 elements per row, which confirms the highly sparse character of this matrix.

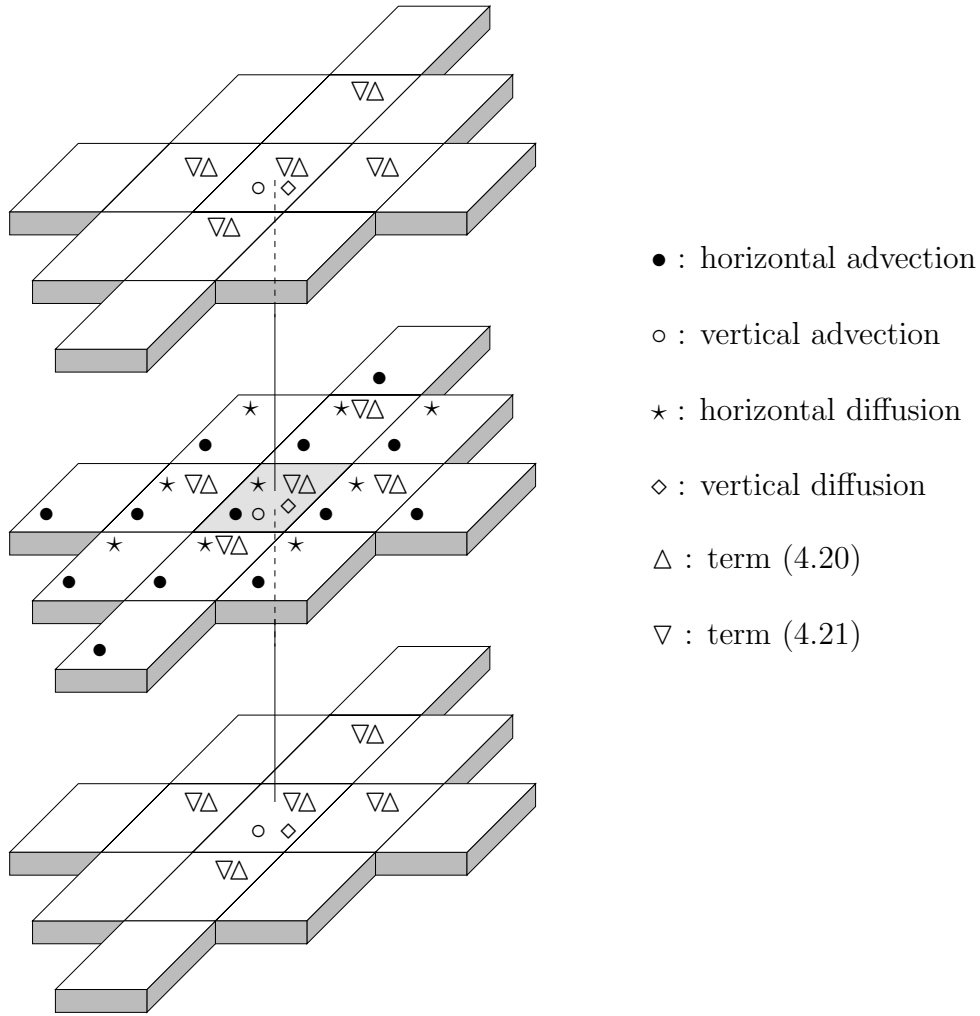


Figure 4.8: 3-D support molecule

4.8 Time integration

Implicit methods have the advantage of overcoming the stability constraint of the explicit schemes and allowing adjustment of the computational time step to the underlying physics. The trapezoidal rule is used here in the increment or delta formulation for the time integration of the semi-discrete system (4.40). First, the rhs term is approximated as

$$\begin{aligned} L \frac{dhc}{dt} &= L \frac{h^{n+1} c^{n+1} - h^n c^n}{\tau} \\ &= L h^{n+1} \frac{\delta^n}{\tau} + \frac{h^{n+1} - h^n}{\tau} c^n \quad \text{with} \quad \delta^n = c^{n+1} - c^n. \end{aligned}$$

Then, we set

$$J = \frac{\partial Pc}{\partial c}, \quad (4.41)$$

the Jacobian of the matrix (Pc) , and

$$M = A h^{n+1}, \quad (4.42)$$

the diagonal matrix containing the cell areas multiplied with the total water depths at cell centers at time step $n + 1$, and

$$\Delta H = h^{n+1} - h^n, \quad (4.43)$$

the diagonal matrix containing the change in water depth between time steps n and $n + 1$. The linearly implicit trapezoidal rule then gives

$$\left(\frac{1}{\tau} M - \frac{1}{2} J^n \right) \delta^n = (Pc)^n + \frac{1}{\tau} \Delta H c^n. \quad (4.44)$$

Both matrix M and matrix ΔH depend on time through the water depth terms. Within each separate time step, all these terms (including the ones at time step $n + 1$) are known beforehand from the flow computation and are therefore considered as constant. As for matrix P , its entries also depend on time (but through the concentration terms), as mentioned previously. We assume these terms to be independent by freezing them to the previous time step as the time marching procedure advances.

4.9 General layout of the model

The flowchart shown in figure 4.9 summarizes the general layout of the 3-D model. The model first constructs a number of time-independent arrays mainly related to geometry and object indexing. The time loop starts with reading the flow data from files. The flux computation starts with a loop over faces: the support molecule is constructed for each face and the real flux computation is then done at the local level of the support molecule; this procedure finishes by affecting the coefficients of the local molecule in their proper position in the global matrix. The loop over layers is nested within the loop over faces.

The second part of the time loop consists of looping over cells and layers to compute the vertical advection and diffusion terms as well as the remaining terms of (4.13).

The last step consists of adding the source terms and solving the resulting system.

4.10 Data structure

Data structure in the 3-D solver is even more important (in comparison with the 2-D case) because of the large amount of data/arrays involved and the added complexity of the 3-D problem. In the 2-D case, we have spent much effort on implementing an efficient data structure and here we use the optimization results and experience gained. The starting point for the implementation of the 3-D solver is the 2-D version itself. Thus we benefit not only from the existing data structure but also from several other practical aspects of the pre-built model.

The extension from 2-D to 3-D is made possible especially by the type of 3-D grid we are using. In fact, the structured character of the grid in the vertical direction implies that all the layers have the same geometry in an $x - y$ plane; looked at from top, all the nodes are exactly superimposed. This means that the

horizontal geometry is layer independent, and therefore it is not necessary to add a third dimension too many arrays involved in the computation.

4.10.1 Numbering of grid objects

Assume we have a 2-D grid with the corresponding numbering of cells, faces, boundary faces and nodes. For the 3-D case, the same 2-D numbering is adopted for the cells and faces in every individual layer. A second index, the layer number, is required in order to fully identify each cell and face in the domain.

For the nodes, however, there is no need for a layer index. Nodes are basically used only for computing the length of faces, which is a layer-independent quantity.

4.10.2 Array indexing in 3-D

Arrays used in the model are related to different variables or objects and have therefore different lengths. Thus, each type of arrays needs adequate treatment in order to extend it to the third dimension. On the other hand, for efficient use of computer memory, the same variable array may appear in different dimensional form depending on the subroutine using this array. Generally, work arrays are used in the high level subroutines, that is, all the integer arrays being used are stored in one unique one-dimensional work array and similarly for the real arrays. This allows for easy memory allocation and deallocation, leading to more efficient use of computer memory. In lower-level subroutines, two-dimensional arrays are used. The correspondence between the two indexing methods can be done automatically through an adequate declaration of array sizes at the beginning of the subroutines. Figure 4.10 gives the correspondence between the indices of a given cell i in both cases. In this figure, $ncell$ is the total number of cells in the domain.

One of the other advantages of the structured aspect of the grid in the vertical direction is that the connectivity lists defining the relative positions of different grid objects are basically not affected. This means that this list is built only once for one layer and used repeatedly for the other layers.

4.10.3 Matrix storage

As mentioned in section 4.7, the final matrix to be constructed has a highly sparse aspect. The CSR format presented in chapter 2 is also used here to ensure an efficient use of the computer memory during the matrix storage.

In the 3-D case, the support molecule of a given cell involves cells from three layers, except for the top and bottom layers, where only two layers are involved. See figure 4.8. This leads to a new definition of arrays ia , ja and ar , the three one-dimensional arrays representing the sparse matrix. Array ia is defined in a straightforward manner and now contains $(ncell * K + 1)$ elements, with K being the number of layers. Elements are ordered in such way that the first $ncell$ positions are reserved for layer 1 followed by $ncell$ positions of layer 2, etc. Arrays ja and ar need more work for extension to 3-D, however. To give some details, let us consider cell number (i, k) presented in figure 4.10. In a one-dimensional array, this cell has

a unique index $j = (k - 1) * ncell + i$. The part of array ar and ja reserved for cell j are divided in three sub-arrays of exactly same length, each corresponding each to a different layer: first, we get entries from cells in the same layer as cell j , then those from the layer on top, and finally those from the layer underneath that element, see figure 4.11. Entries in the three sub-arrays are arranged in such way that they correspond to superimposed elements, for example the m^{th} element in the second sub-array corresponds to the cell located on top of the one in the m^{th} position in the first sub-array, and the m^{th} element in the third array is located underneath that same element. For the top-most and bottom layers, only two sub arrays are reserved. Although in the present discretization schemes of the different flux terms some of the entries are known to be zeros, arranging data in this way presents at least two advantages: first, it allows for future testing of different discretizations without having to introduce major changes in the code, and second, it is more convenient for debugging purposes. The length of arrays ar and ja is therefore almost $3K$ times larger than their original length in the 2-D case.

4.11 Boundary conditions

Some aspects related to boundary treatment were already presented in section 4.6 dealing with flux computation. A general summary is given in the present section.

Per individual layer, horizontal fluxes are dealt with, as in the 2-D case. For advective fluxes, a zero-flux condition applies for closed boundaries. For open boundaries, either a Dirichlet or a simple upwind condition is adopted, depending on the velocity direction. For the diffusive fluxes, the boundary treatment consists in building the shadow volume using the only surrounding cells available. Details on construction of this shadow volume for different possible situations are given in section 2.4.6.

In the vertical direction, advection fluxes as well as diffusive fluxes are equal to zero at the topmost and bottom horizontal interfaces. In the present version of the code, terms (4.20) and (4.21) are set to zero in the top and bottom layers.

4.12 Conclusions

In this chapter, the main building steps of the new 3-D transport solver are described. To summarize, the model solves the layer averaged advection-diffusion equation over an unstructured grid in the horizontal direction and a structured grid in the vertical direction. The shape of the unit volumes is such that the vertices of the volumes which belong to a same column have the same time-independent $x - y$ coordinates whereas the vertical coordinates change in time. In order to proceed with the finite volume formulation, the transport equation is first averaged over each layer. The Leibniz rule is the basic tool used to average the different terms of the equation. The unknown concentration, computed at the center of each volume, represents therefore the average value over the volume.

The estimation of different horizontal and vertical fluxes of the finite volume

method is explained in detail. In the vertical direction, second-order central differencing, rather than upwind, is applied for most terms.

The model uses a linearly implicit method for the time integration and the resulting overall linear system is solved using a BICGStab method with an ILU preconditioning.

For the 3-D model, the data structure is even more important than in the 2-D case because of the very large amount of interfering parameters. In particular, details are given describing the way array indexing and matrix storage are implemented.

The numerical application of the 3-D solver follows in the next chapter. A hypothetical test case as well as a three-dimensional simulation of the Dutch coastal model are used to test and evaluate the performance of the new model.

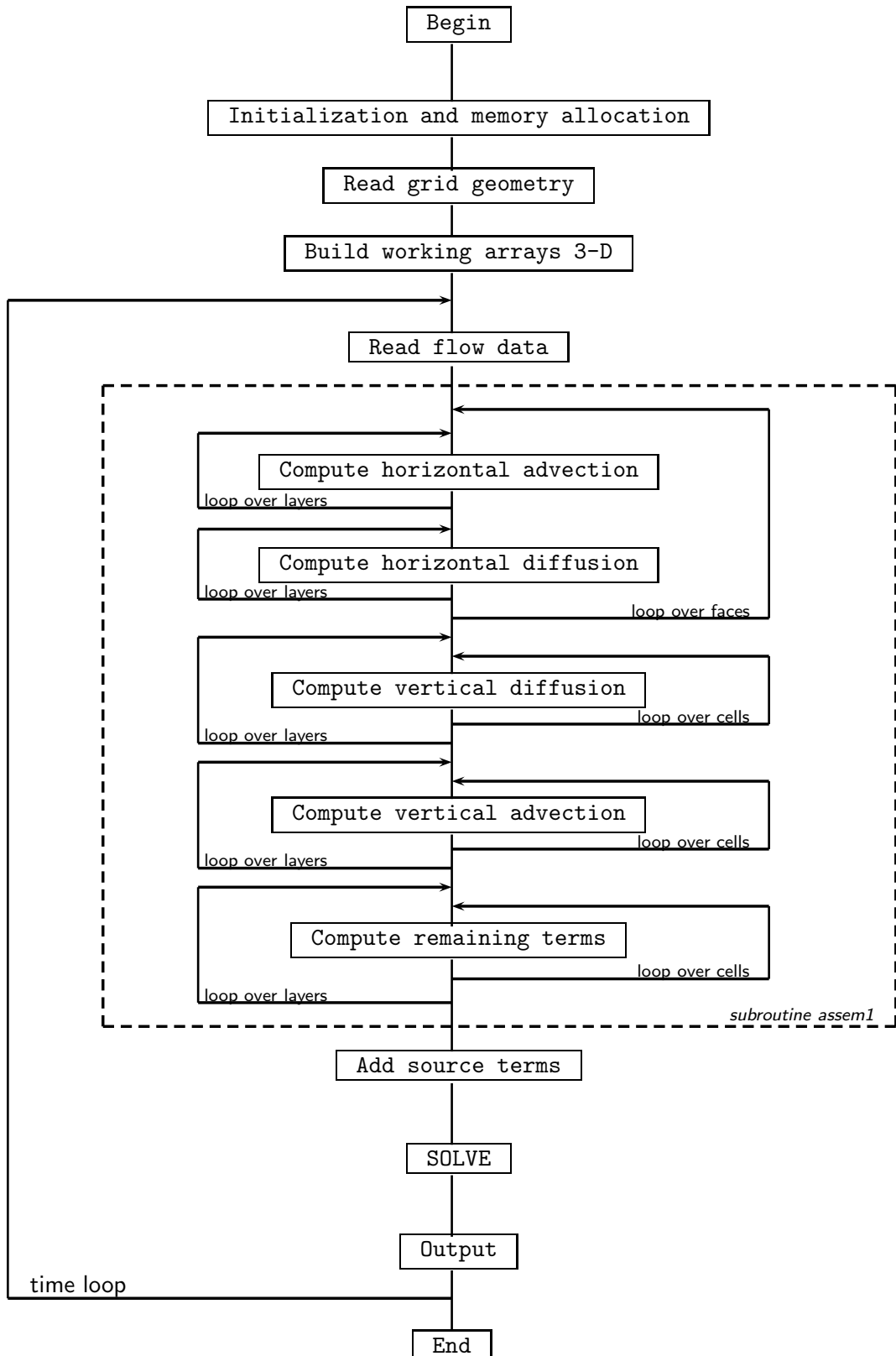


Figure 4.9: General layout of the solver algorithm

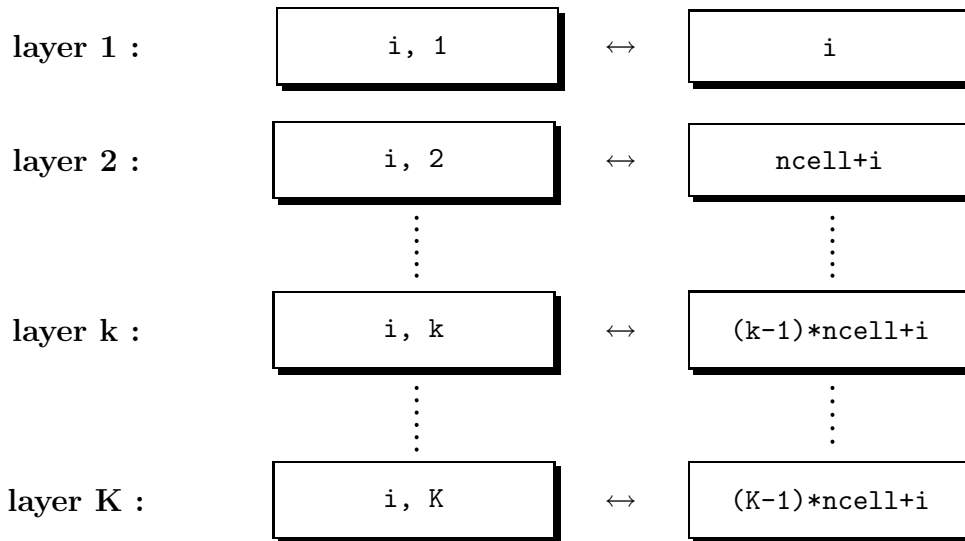


Figure 4.10: Correspondence between one- and two-dimensional array indexing

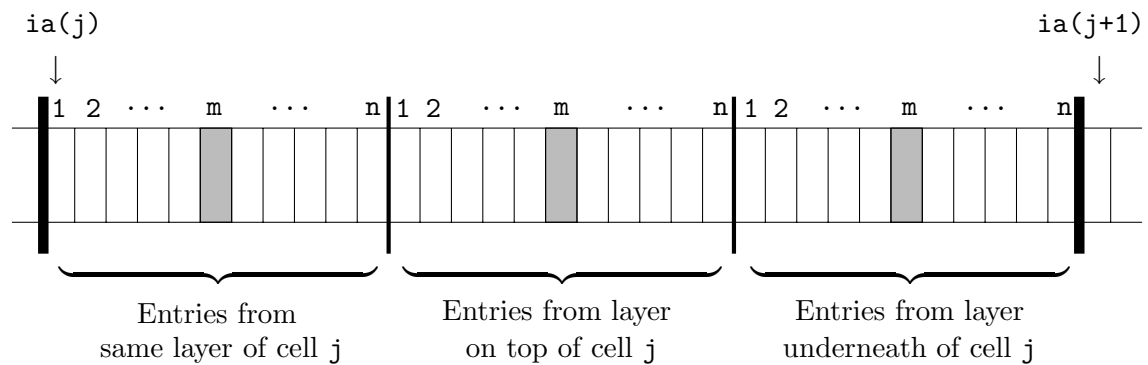


Figure 4.11: Structure of arrays ja and ar in the 3-D case

Chapter 5

Numerical results of the 3-D solver

5.1 Introduction

This chapter is devoted to the application of the constructed 3-D transport solver to practical cases. A preliminary step to the transport computation consists of obtaining the required flow parameters by solving the flow problem using the three-dimensional version of WAQUA named TRIWAQ. The raw output data from TRIWAQ needs then to be preprocessed in order to provide conservative data for the transport computation. Preprocessing detail are presented in the next section. Two numerical applications are carried out: first a hypothetical test case consisting of simulation of transport scenarios for a stretch of river and second, again the Dutch coast problem is considered but now in its 3-D version. The performance of the newly built model is evaluated based on these two cases.

5.2 Flow data preprocessing

Before starting with the numerical results of the 3-D solver, it is necessary to mention the intermediate step of data preprocessing. It is clear that the use of raw flow data will inevitably lead to the occurrence of unrealistic source and sink terms. The constancy condition defined in section 2.7.2 therefore needs to be satisfied, and for this purpose, a preprocessing step is required. The data preprocessing technique used for the 2-D case needs to be reconsidered to make it suitable for the 3-D case.

If we set a uniform concentration in the transport equation, we need to retrieve exactly the flow continuity equation in order to ensure that the constancy condition is met. Doing so, horizontal and vertical diffusion terms as well as term (4.21) disappear immediately since they are a function of the gradient of concentration. Term (4.20) does not cancel, and for now, we propose to leave out this term from the discrete transport equation before proceeding with data preprocessing. All calculations made including this term and presented later in this chapter showed that it has a very limited impact on the results.

Now, writing the transport equation for a uniform concentration gives the condition which needs to be satisfied by the flow coefficients:

$$\frac{A}{\tau}(H_{cell}^{n+1} - H_{cell}^n) = - \sum_e H_e^n U_e^n l_e - A [\omega^n]_{z_k}^{z_{k-1}}. \quad (5.1)$$

We need to compute coefficients which satisfy equation (5.1) but prefer to have them at a larger time step than the one used for the flow computations, i.e.:

$$\frac{A}{T}(H_{cell}^{N+1} - H_{cell}^N) = - \sum_e H_e^N U_e^N l_e - A [\omega^N]_{z_k}^{z_{k-1}}, \quad (5.2)$$

where N is a multiple of n .

We can use the same technique of time integrated flow data described in section 2.8.1 and which consists in combining data at small time steps to generate new data at larger time steps. The procedure can be made easier using some special output variables from TRIWAQ called “time integrals”. These are *WPINT*, *DISUNT*, *DISVNT*, *UPINT*, and *VPINT*, defined as follows:

$$\begin{aligned} DISUNT_k^{t+T} &= \int_t^{t+T} U_k H_k dt & DISVNT_k^{t+T} &= \int_t^{t+T} V_k H_k dt \\ UPINT_k^{t+T} &= \int_t^{t+T} U_k dt & VPINT_k^{t+T} &= \int_t^{t+T} V_k dt \\ WPINT_k^{t+T} &= \int_t^{t+T} \omega_k dt & & \end{aligned}, \quad (5.3)$$

where T is such that the time level $N + 1$ in (5.4) corresponds to $t + T$.

For a given element, these variables satisfy the following mass conservation equation:

$$\frac{A}{T}(H_{cell}^{N+1} - H_{cell}^N) = \sum_e \frac{DISUNT^{N+1}}{T} l_e + \sum_{top,bottom} A \frac{WPINT^{N+1}}{T}. \quad (5.4)$$

In this equation, *DISUNT* may also be *DISVNT* depending on the direction of the face. By making a correspondence between (5.2) and (5.4), the following choice for the flow parameters appears to ensure the required conservation property:

$$H_e^N = \frac{DISUNT^{N+1}}{UPINT}; \quad U_e^N = \frac{UPINT^{N+1}}{T}; \quad \omega^N = \frac{WPINT^{N+1}}{T}; \quad (5.5)$$

For the particular case where *UPINT* is equal to zero, the value of H_e^N is simply taken equal to the elevation difference $z_{k-1} - z_k$.

Computations using these coefficients led to satisfactory results, as will be shown later in this chapter. Occasionally, however, the discrete mass conservation (5.4) is not accurately satisfied, especially near regions where the bathymetry presents a

highly distorted geometry. To overcome this, we propose to apply a correction on H_{cell}^{N+1} but using a slightly different method than in the 2-D case described by (2.36). Here, for the computation of the solution at time level $N + 1$, the model reads from input data the required velocities and elevations at time step N and uses these values to compute H_{cell}^{N+1} according to (5.2) instead of reading H_{cell}^{N+1} from the input. In the next time level $N + 2$, H_{cell}^{N+1} does not have the value previously computed but it is read again from the input files and it is with respect to this latter point that the procedure differs from the two 2-D case. This method ensures that (5.2) is satisfied up to machine accuracy everywhere in the domain. However, this also means that the water depth H_{cell} may occasionally have a discontinuous profile in time. Our computations showed that this has basically no effect on the accuracy of the results.

5.3 Test case

The 3-D solver was first applied to a test case in order to facilitate the implementation of the model. The physical domain of the chosen test case consisted of a stretch of river 500 m long and 100 m wide with a bed presenting some deformations and irregularities as shown in figure 5.1. The idea is to generate the 3-D motion of the flow and subsequently of the tracer.

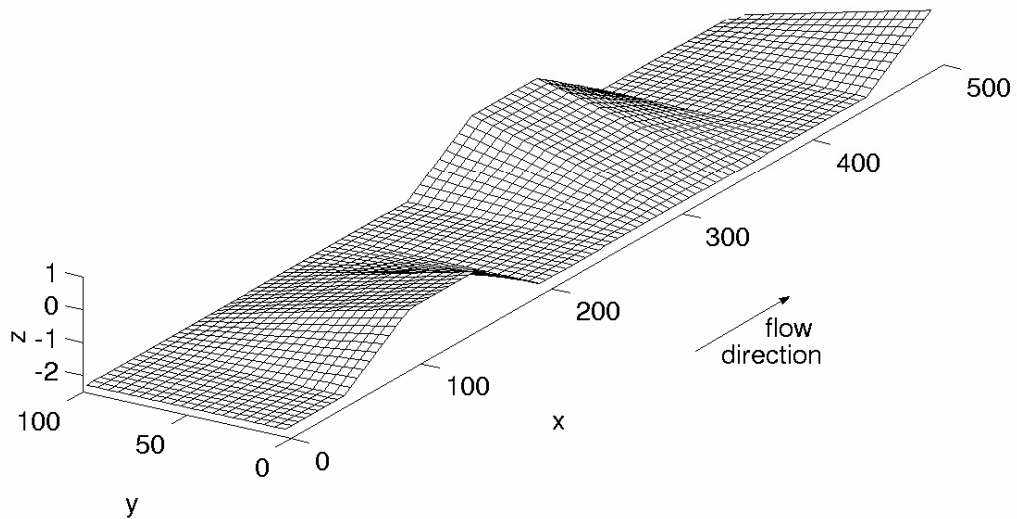


Figure 5.1: Grid of the test case

5.3.1 Grid

The domain of the test case was covered with a rectilinear grid of 100 points in the x-direction and 20 points in the y-direction. The space step was 5 m in both directions. In the vertical direction, the domain was subdivided in 10 layers of equal thickness.

5.3.2 Hydrodynamic coefficients

The flow simulation using TRIWAQ was carried out for a duration of 3 hours 20 minutes with a time step of 0.01 minute. The initial conditions consisted of a uniform water level equal to 1 m (water level above an imaginary plane of reference) and a uniform velocity equal to 0.2 m/s in the main flow direction. The boundary conditions consisted of imposing a constant discharge of 5.9 m³/s at the upstream boundary and a periodical water level at the downstream boundary according to the function shown in figure 5.2. The bottom friction was estimated using Chezy formula with a coefficient equal to 70.

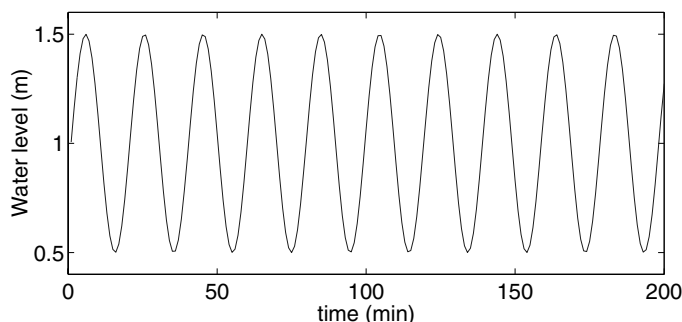


Figure 5.2: Water level at the downstream boundary

Figure 5.3 shows a snapshot of the resulting velocity field for the topmost 5 layers after 30 minutes from the start of the flow computation.

The following coefficients had to be stored at the required time steps in order to be used by the transport solver:

- horizontal and vertical velocities
- layer elevation at the velocity points and water level points. (See figure 4.4)
- type and value of the boundary conditions

5.3.3 Numerical results for the test case

The numerical experiments started with a check for the constancy condition. Using a computational time step of 5 s, the run for the constancy condition was performed successfully. The concentration remained perfectly equal to the chosen constant value, in our case $c = 1$, up to machine accuracy. This is mainly because the correction applied on H_{cell}^{N+1} is computed within the solver.

Before continuing with further simulations, we carried out a check related to term (4.20). It consists in running the constant solution problem while including this term. Figures 5.4 and 5.5 show the result and the analysis of the simulation of the constant solution in this case. Figure 5.4 corresponds to a time of 40 min and shows the concentration distribution. Figure 5.5 shows some statistics of these results: (a) shows the maximum and minimum values through the whole domain as a function of time, (b), (c) and (d) show the mean value, the number of cells where the error exceeds 1% and the standard deviation, respectively, all functions of time.

The main conclusion is that the concentration remained almost unchanged and that therefore the adopted approximation of omitting term (4.20) is legitimate.

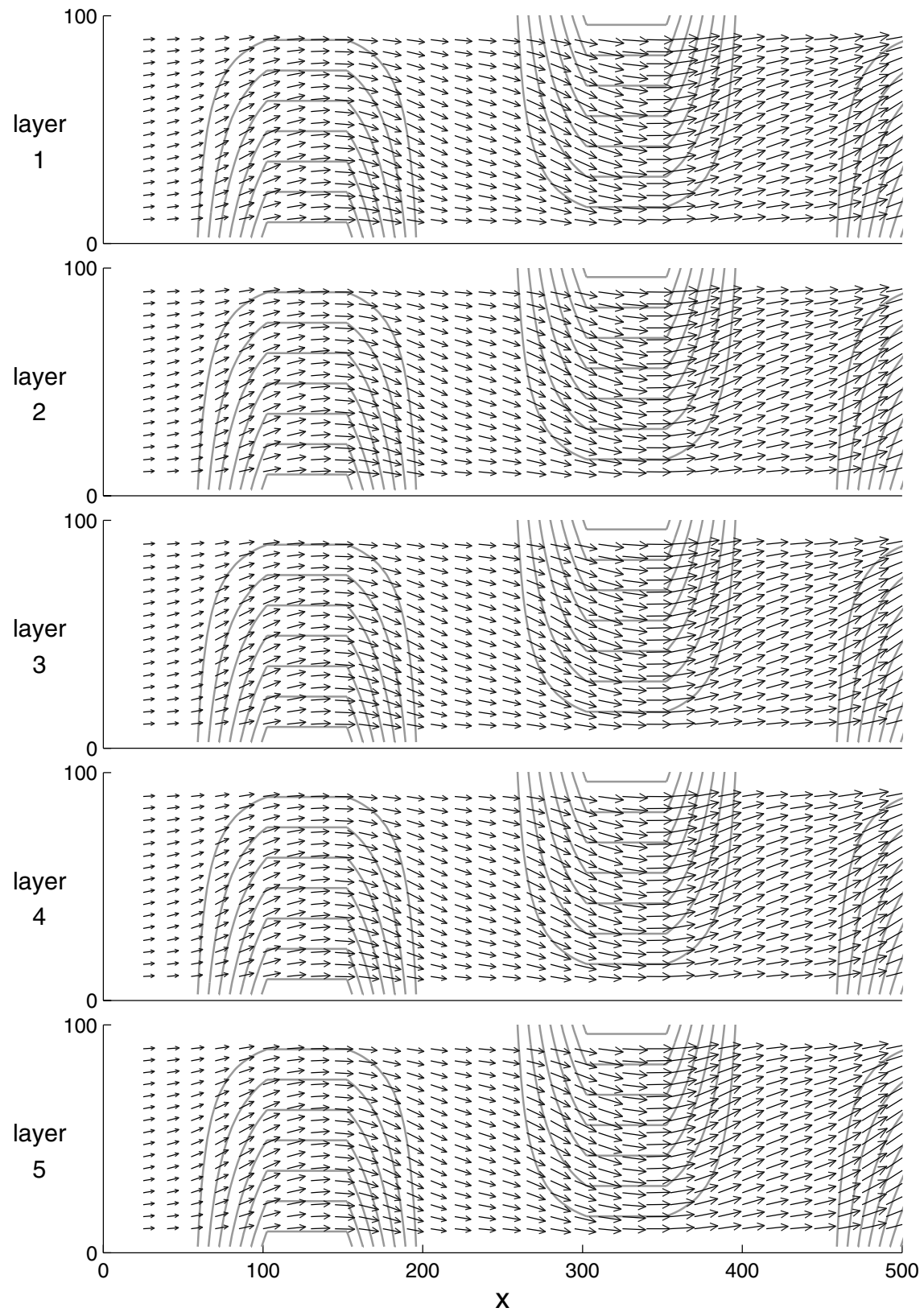


Figure 5.3: Velocity field at the topmost 5 layers at time 30 minutes

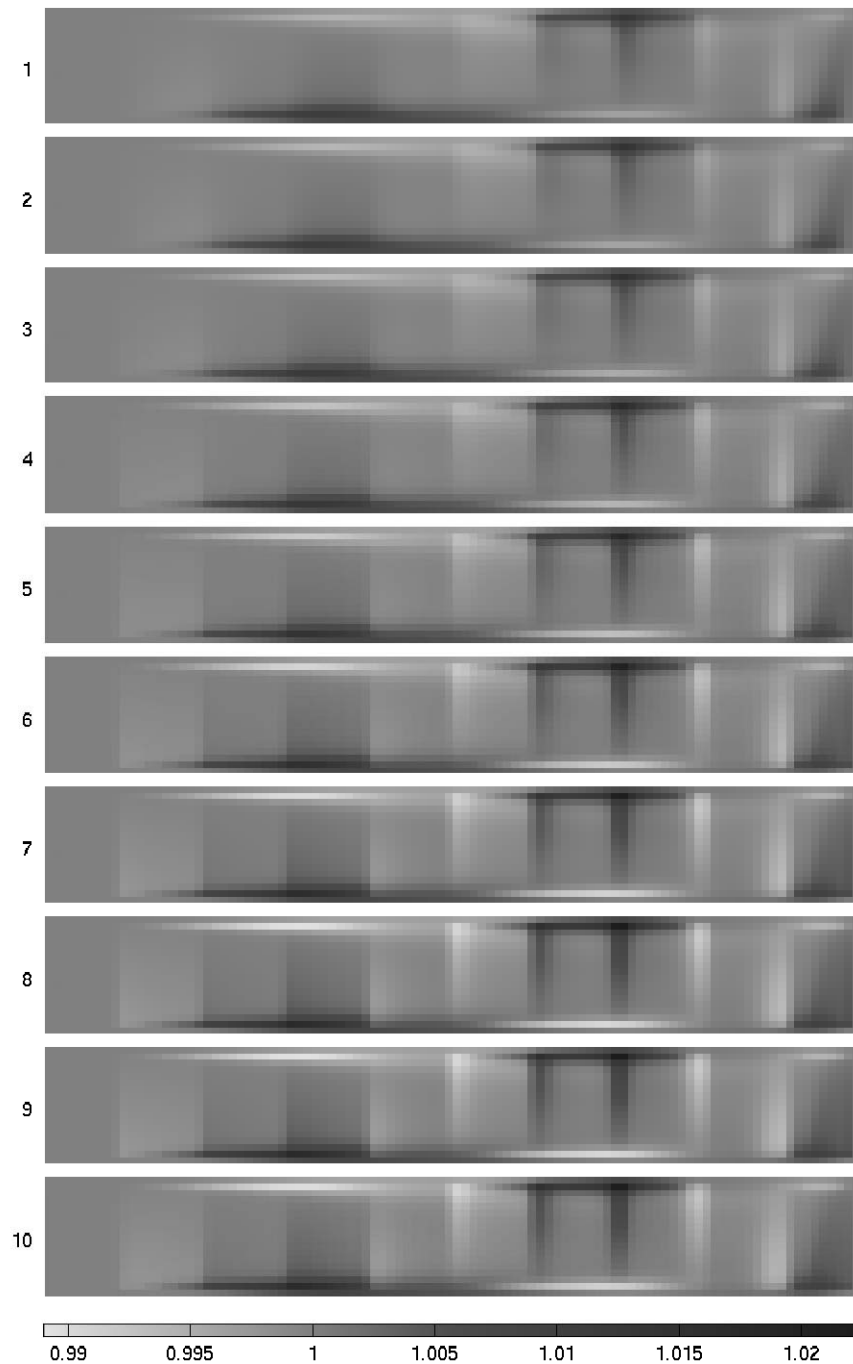


Figure 5.4: Constant solution after 40 min

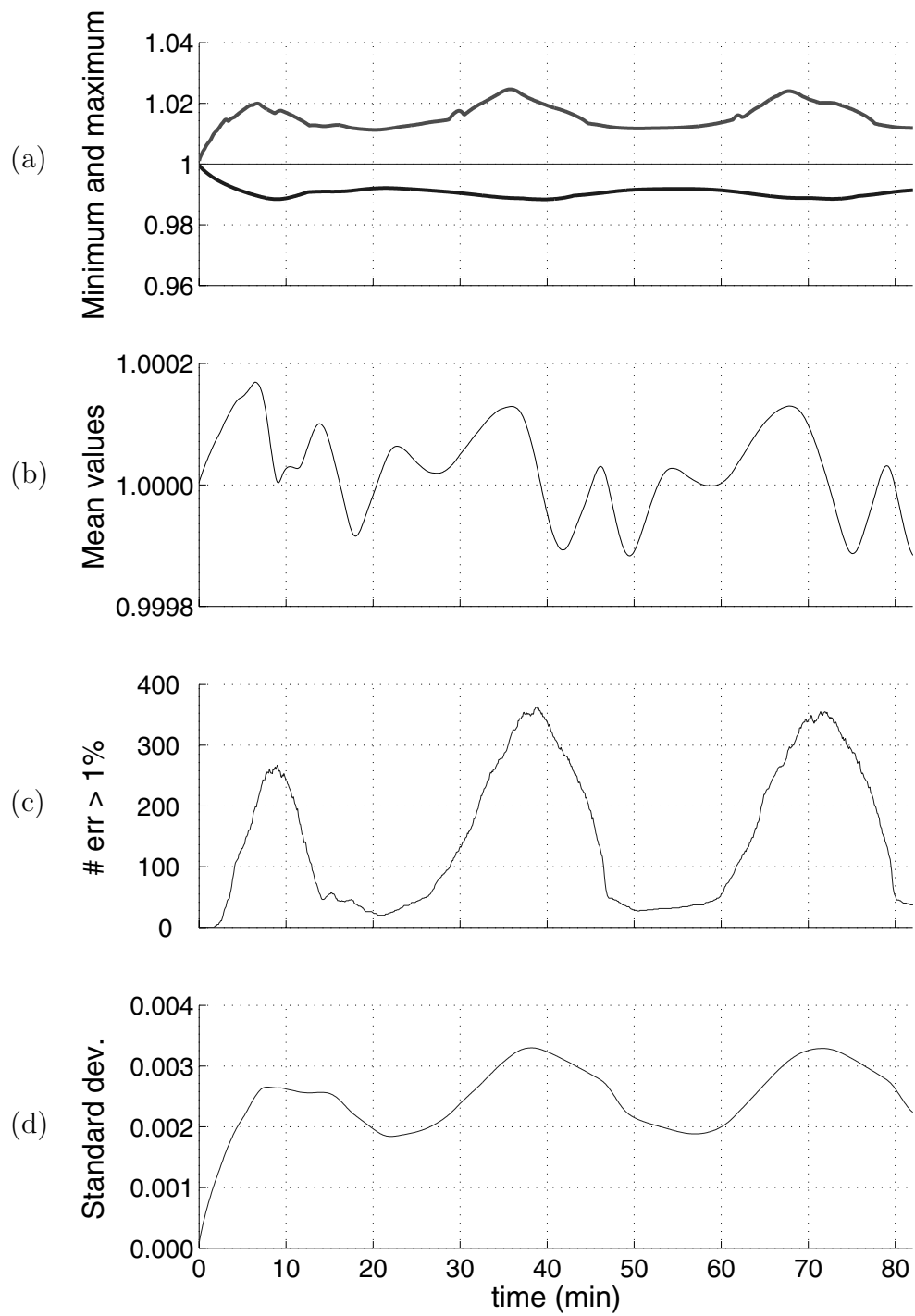


Figure 5.5: Layer depth and layer elevation

A continuous release of a miscible pollutant was then simulated. The release point was located at the upstream boundary of the stretch, halfway between the right and left banks and was assumed to take place at the water surface (layer 1). The computation started with a concentration equal to zero throughout the domain. The diffusion coefficients were estimated at $0.01 \text{ m}^2/\text{s}$ for the horizontal direction and $0.001 \text{ m}^2/\text{s}$ for the vertical direction. The computational time step was 5 s.

Figure 5.6 shows for each of the 10 layers a top view of the horizontal concentration profile. Three contour lines represent concentration 0.05, 0.5 and 1 (when applied). Figure 5.7 shows a 3-D representation of the iso-value $c=0.25$ viewed from different angles. Both figures show a 3-D aspect of the tracer transport. The result of the 2-D simulation of the same problem is presented in figure 5.9 for comparison.

Figure 5.8 shows the total mass of tracer present in the domain, computed based on the concentration profile obtained. The dashed straight line represents the total mass released, which, in our case, is linear in time corresponding to a constant release. This figure shows a very good conservation of mass during the first 15 minutes of the simulation. After that, the solute starts leaving the domain from the right boundary and of course the two curves start deviating from each other. In this computation, no negative concentrations were registered.

For comparison purposes, figure 5.10 shows the results of the simulation of the same scenario using TRIWAQ transport solver. Compared to figure 5.6, TRIWAQ produces a slightly less diffusive pattern in the horizontal direction. However, it has the disadvantage of producing large negative concentrations in many locations.

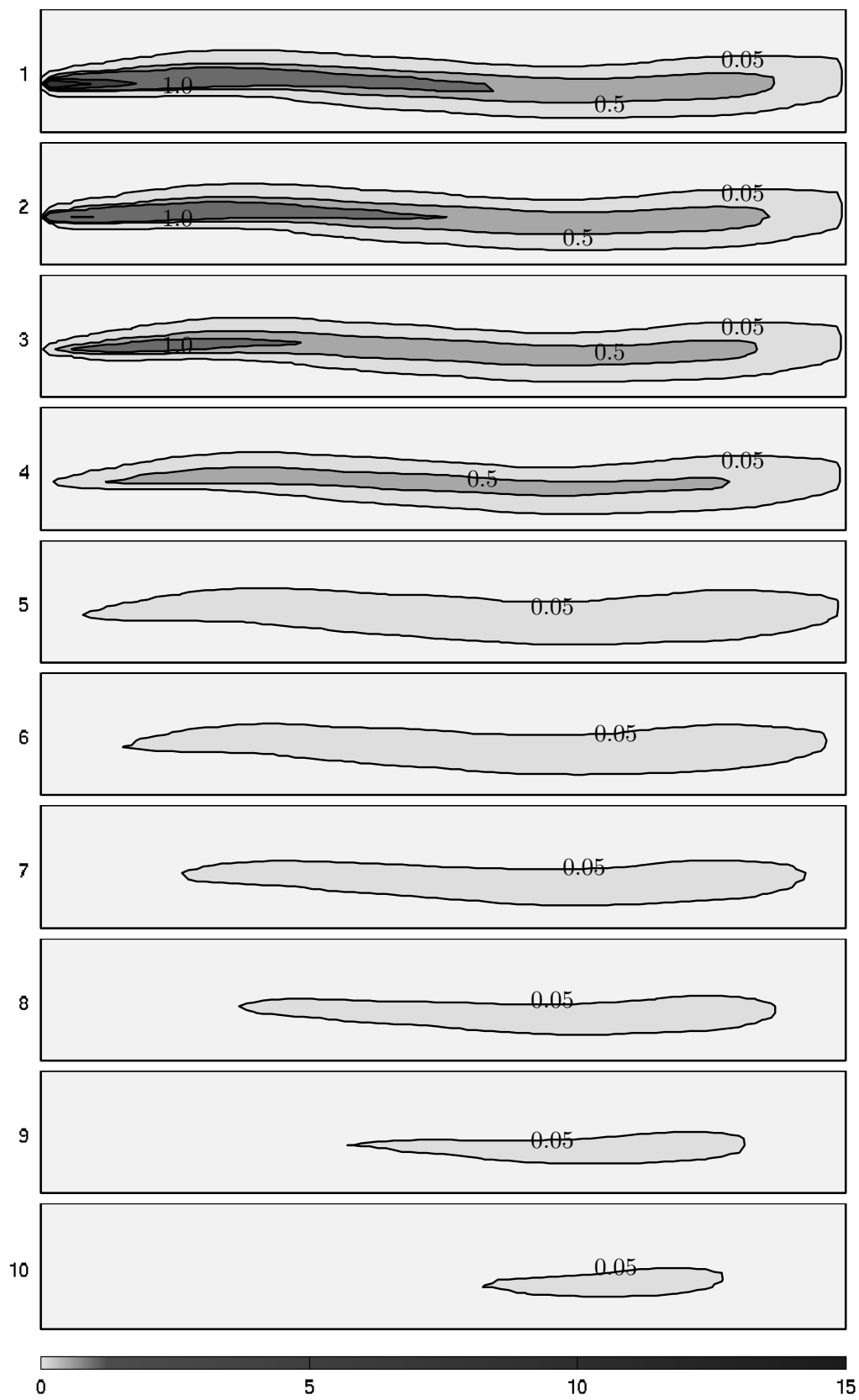
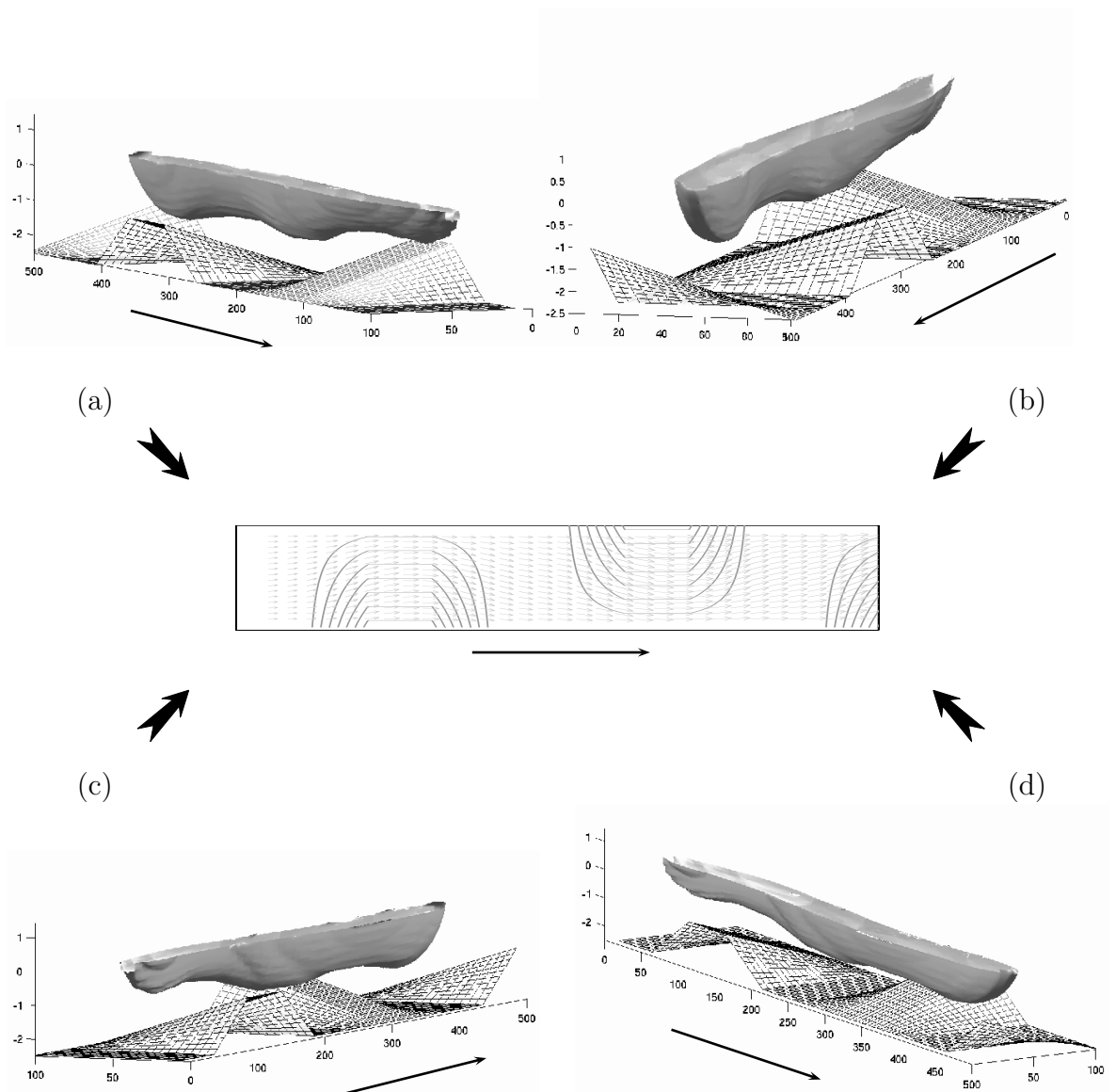


Figure 5.6: Top view of the concentration at different layers after 16 min



NB:

- The long thin arrows indicate the main flow direction in each view
- The bold arrows show the viewer position in each plot

Figure 5.7: 3-D representation of the iso-value 0.25 after 16 min viewed from different angles

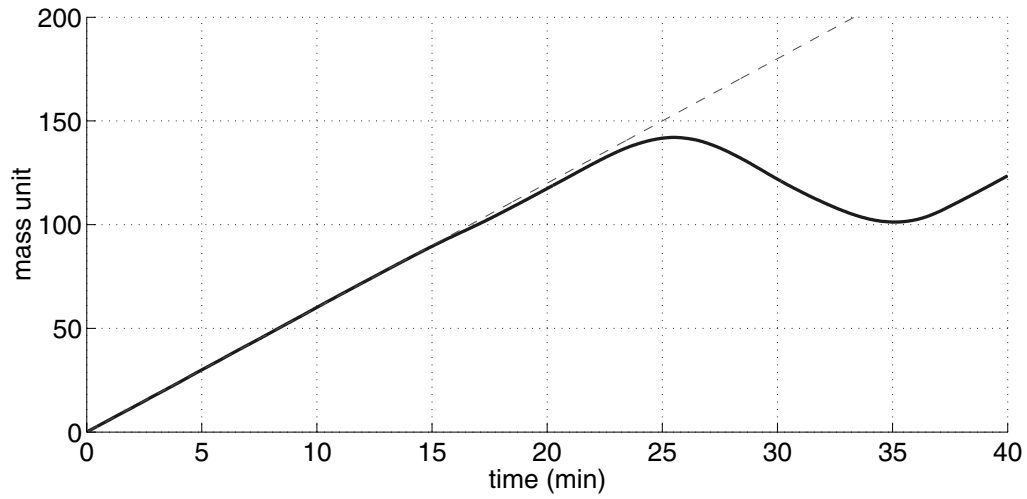


Figure 5.8: Total mass in the computational domain

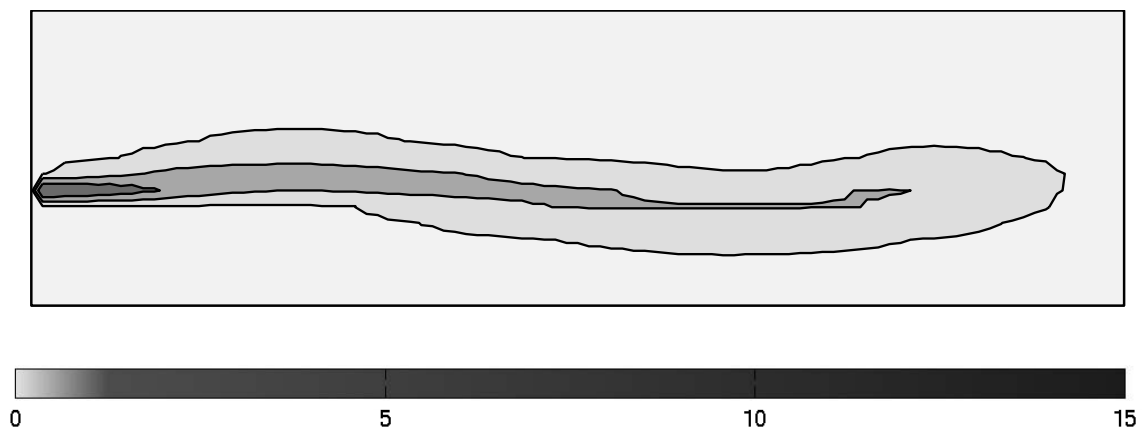


Figure 5.9: 2-D simulation of the test case. Situation after 16 min.

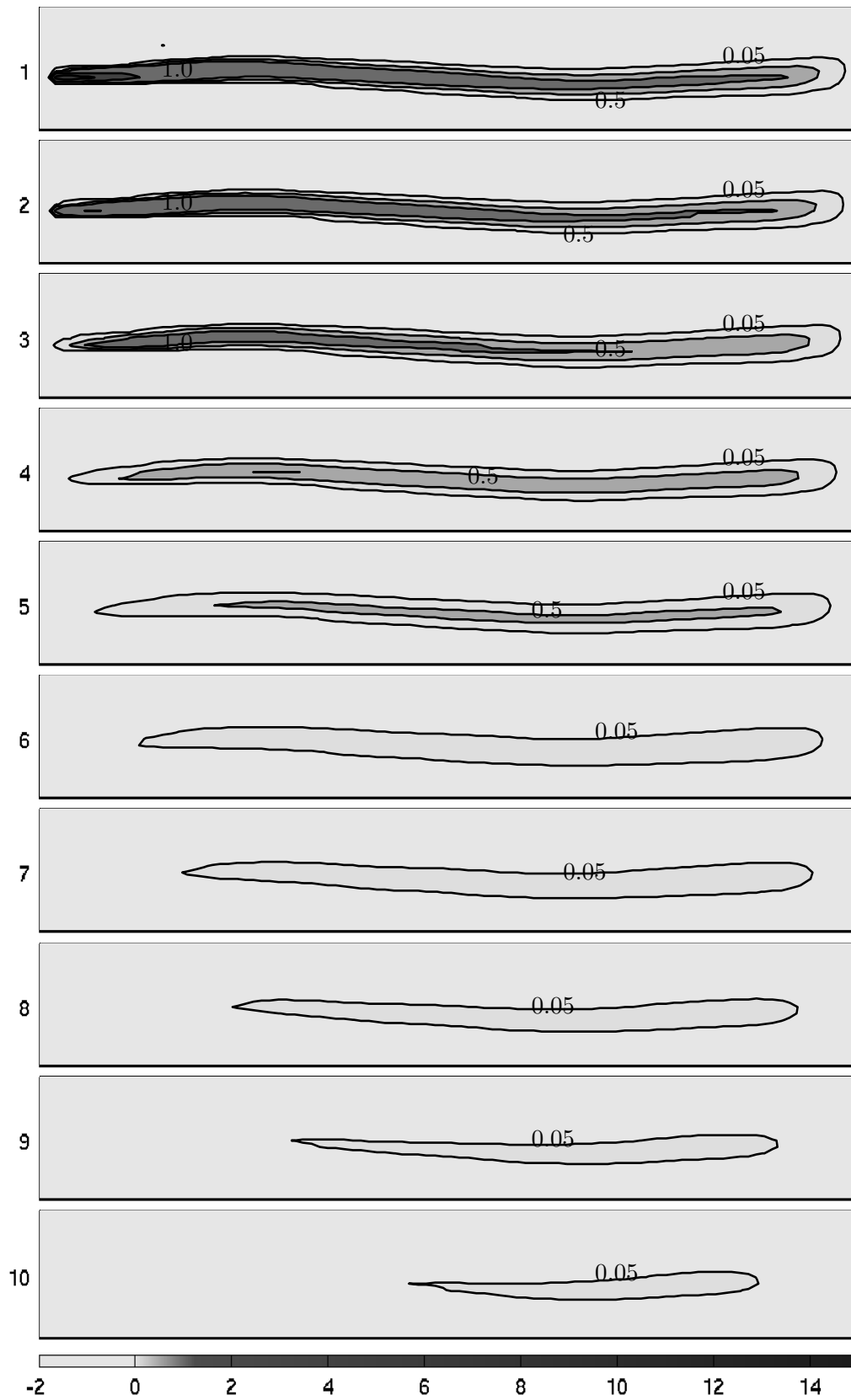


Figure 5.10: TRIWAQ simulation of the concentration at the different layers

The sensitivity analysis carried out for this test case revealed that the value of the vertical diffusion coefficient plays an important role in the final vertical distribution of the concentration. Figure 5.11, obtained using $D_v = 0.01 \text{ m}^2/\text{s}$ instead of the value 0.001 used previously, shows a result differing widely from that in figure 5.6. These two values of D_v were also used to produce time history plots at cell number 170. The location of this cell is marked in figure 5.12-c. In figures 5.12-a and 5.12-b, the topmost curve corresponds to the concentration in layer 1, the next to layer 2, etc, which was of course to be expected since the release takes place in the top layer. The vertical mixing is clearly larger for the higher D_v value whereas in the other case, the low mixing results in a concentration which remains high at the top layer. For some environmental applications, this fact might be important and it is therefore highly recommended to use a good estimation of this coefficient.

Figures 5.13, 5.14 and 5.15 show time history plots of the averages of the concentrations over the 10 layers for both values of vertical diffusion and $D_h = 0.01 \text{ m}^2/\text{s}$ together with the corresponding result of a 2-D run using $D_h = 0.01 \text{ m}^2/\text{s}$ as well. These figures correspond to three partial cross-sections of the canal depicted in figure 5.12-c. Two general conclusions can be drawn from these plots: first, there is a good match between the 3-D results, which means that the quantity of pollutant located in a given column of cells is almost equal in both cases. Only the vertical distribution may differ. Second, the 2-D concentrations are in most cases somewhat higher than the 3-D values in the area located around the middle of the canal. Closer to the banks, the situation is reversed which indicates that the tracer transport was less diffusive in the 2-D case. This led us to believe that for the simulation of a same scenario using the 2-D and 3-D model, different values for the horizontal diffusion coefficient have to be chosen. Just for a first impression of this issue for the present test case, we carried out three runs of the 2-D model using different values of D_h , namely, 0.1, 0.05 and the original $0.01 \text{ m}^2/\text{s}$ value. A comparison of the time history plots obtained with the plots of the 3-D runs (using $D_h = 0.01 \text{ m}^2/\text{s}$) shows that for a large number of cells, a value of D_h such that $0.05 < D_h < 0.1$ gave a result closer to the 3-D case.

Here, it is important to mention that basically the variation of the horizontal diffusion in the flow solver does not affect the velocity field produced. Previous runs using values 0.001, 0.01 and $0.1 \text{ m}^2/\text{s}$ (chosen deliberately of different orders of magnitude) only led to a maximum relative variation of velocities and water levels of the order of 10^{-4} . This means that the same flow data set can indeed be used for several transport simulations in which only horizontal diffusion varies.

Going back to the analysis of the vertical diffusion, we need to mention that up to here, only constant profiles have been taken for the vertical diffusion. We repeated the same experiment but with the two parabolic profiles depicted in figure 5.17. This kind of profile is preferred by some authors, such those of [43], for some practical applications. These two profiles were chosen in such way that areas under the curves correspond to those same areas for the constant vertical diffusion 0.01 and $0.001 \text{ m}^2/\text{s}$. Figure 5.18 and 5.19 show respectively for these two values the resulting spreading for both parabolic profiles. Compared to figures 5.6 and 5.11, we notice that the shape of the diffusion profile does not have much effect on the final distribution and that the order of magnitude of D_v is rather the predominant

factor.

A third different type of models often used in practical applications is the algebraic model described in appendix A. In this type of models, the vertical diffusion has a spatial varying parabolic profile. The result obtained in this case is depicted in figure 5.20. The spreading is rather similar to the previous experiments with the constant and parabolic profiles and with D_v of the order of $0.01 \text{ m}^2/\text{s}$.

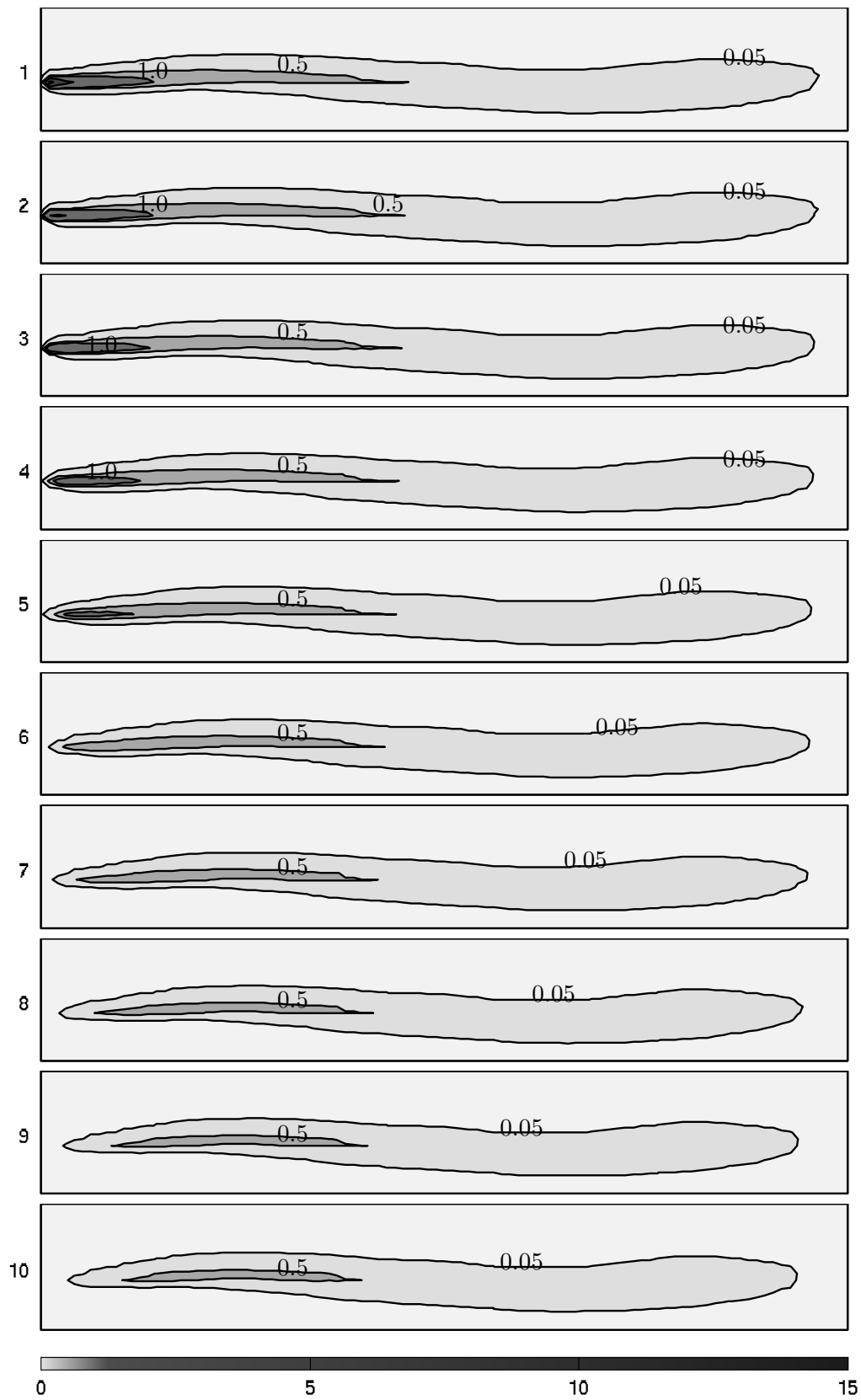


Figure 5.11: Top view of the concentration at different layers after 16 min ($D_v = 0.01 \text{ m}^2/\text{s}$).

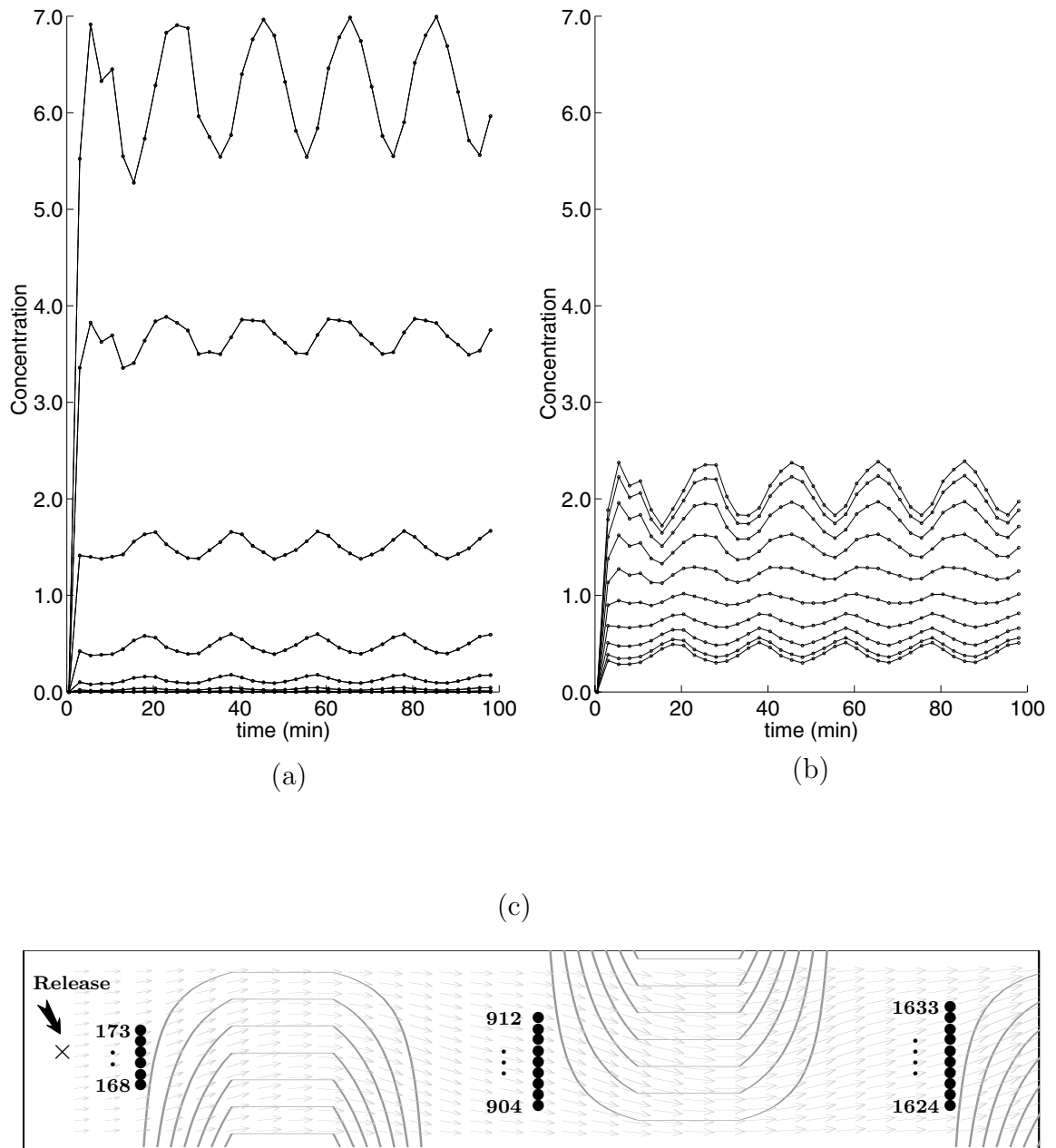


Figure 5.12: Time history of the concentration at cell 170, with (a) $D_v = 0.001 \text{ m}^2/\text{s}$ and (b) $D_v = 0.01 \text{ m}^2/\text{s}$. (c) shows the location of the time history plots.

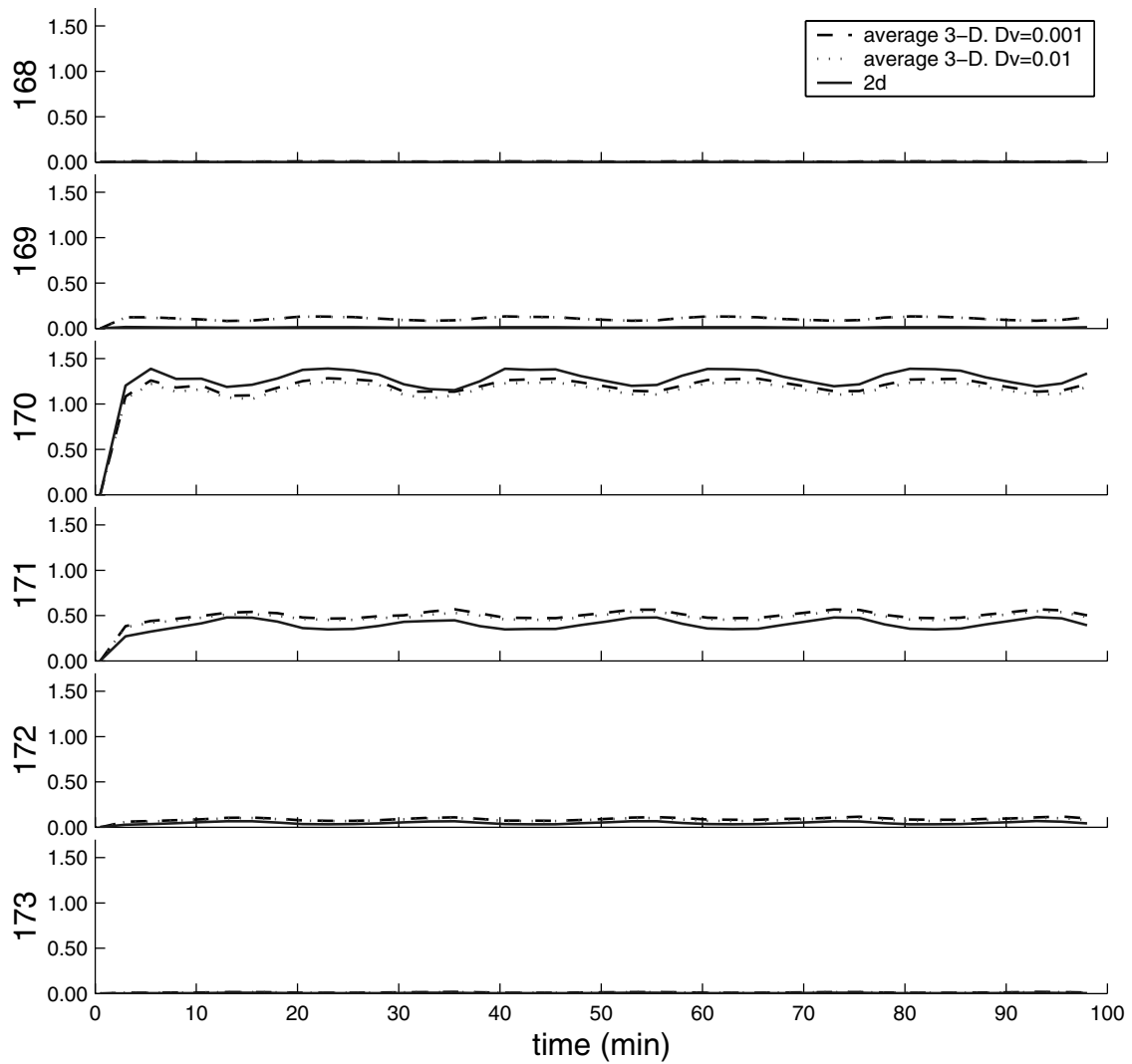


Figure 5.13: Time history plots at the cross-section formed by cells 168 to 173. See locations in figure 5.12-c

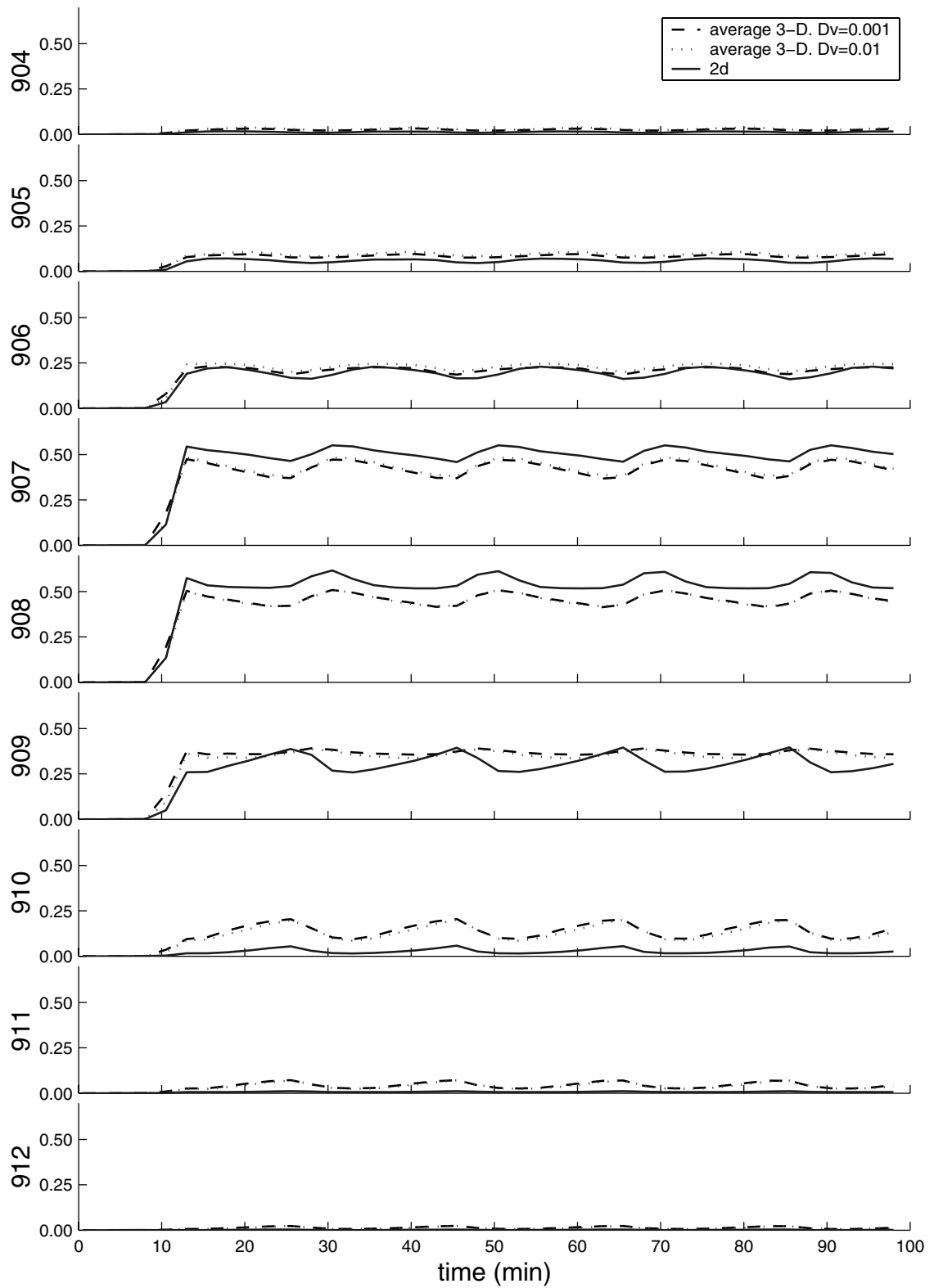


Figure 5.14: Time history plots at the cross-section formed by cells 904 to 912. See locations in figure 5.12-c

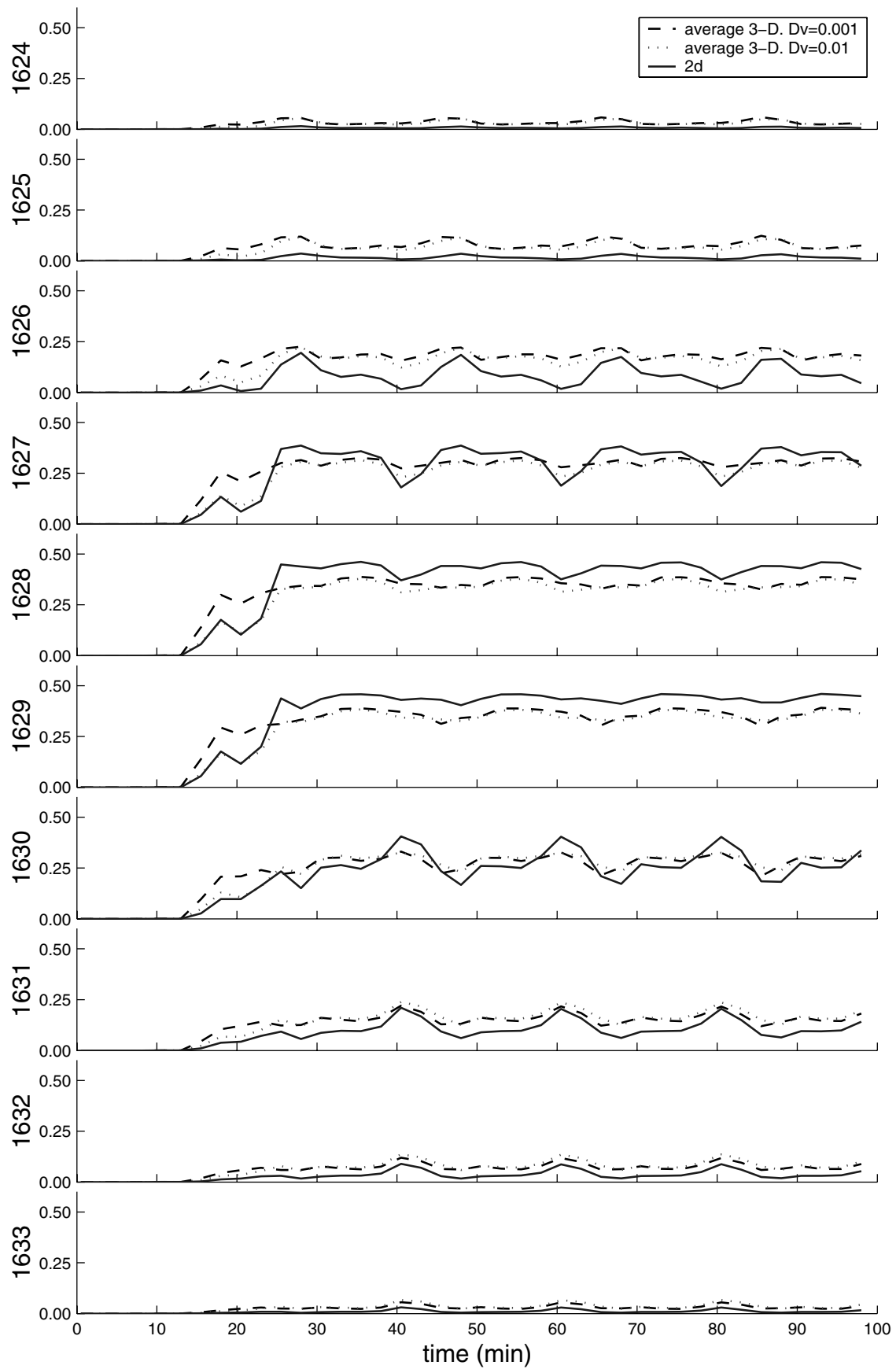


Figure 5.15: Time history plots at the cross-section formed by cells 1624 to 1633. See locations in figure 5.12-c

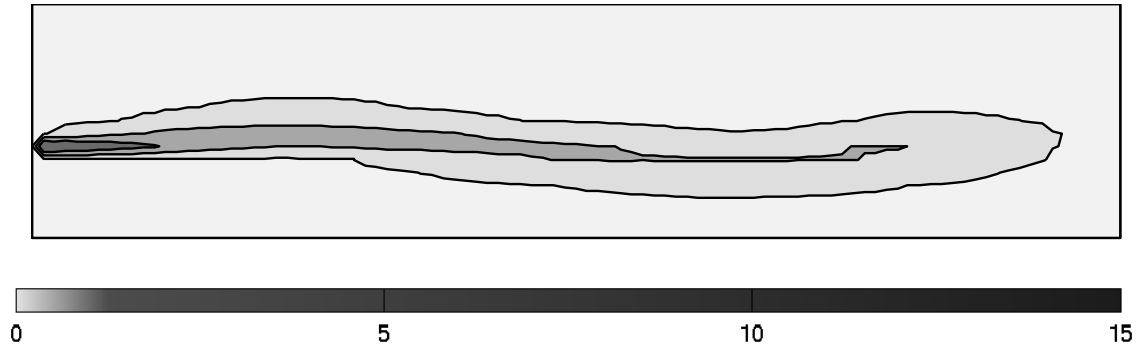


Figure 5.16: Spreading of the pollutant resulting from a 2-D simulation

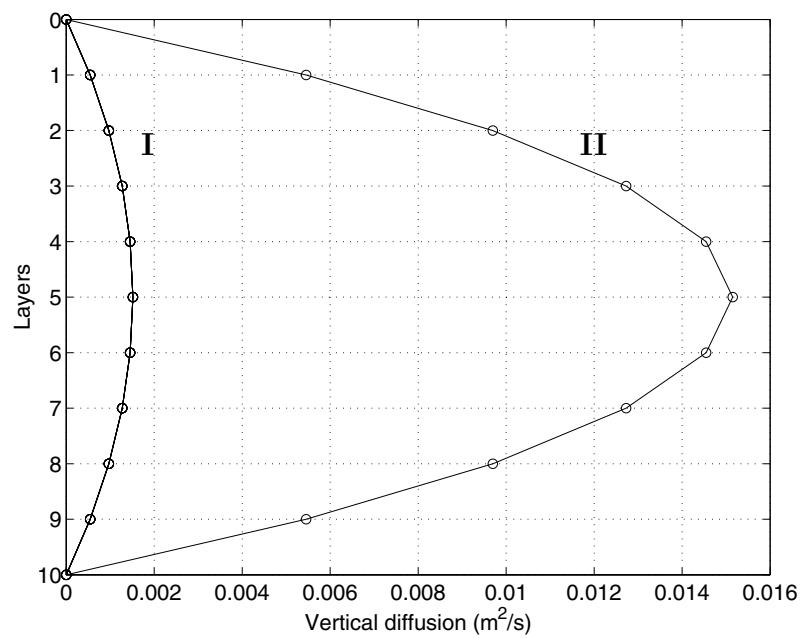


Figure 5.17: Parabolic vertical diffusion profiles.

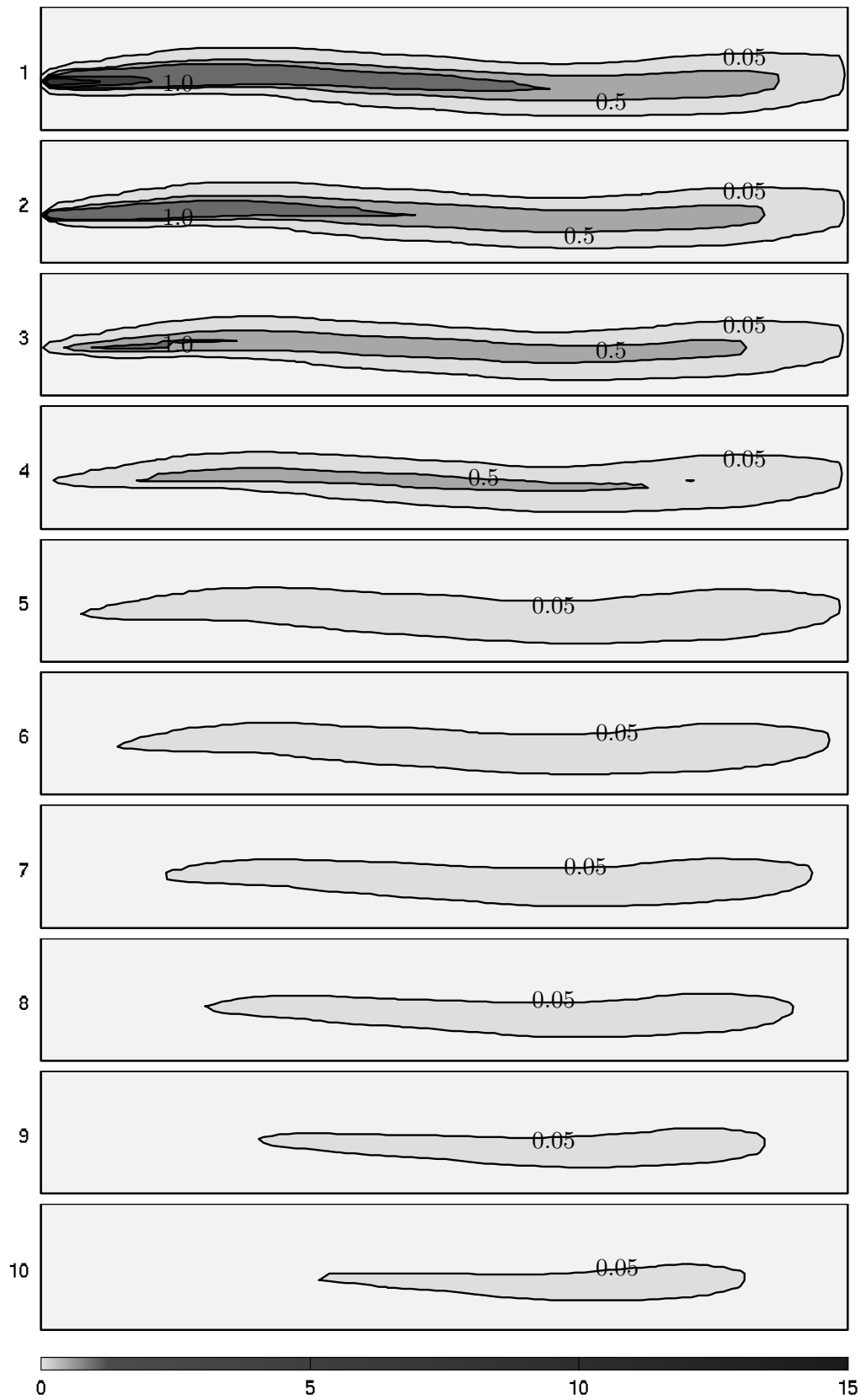


Figure 5.18: Top view of the concentration at different layers after 16 min using parabolic vertical diffusion (profile I).

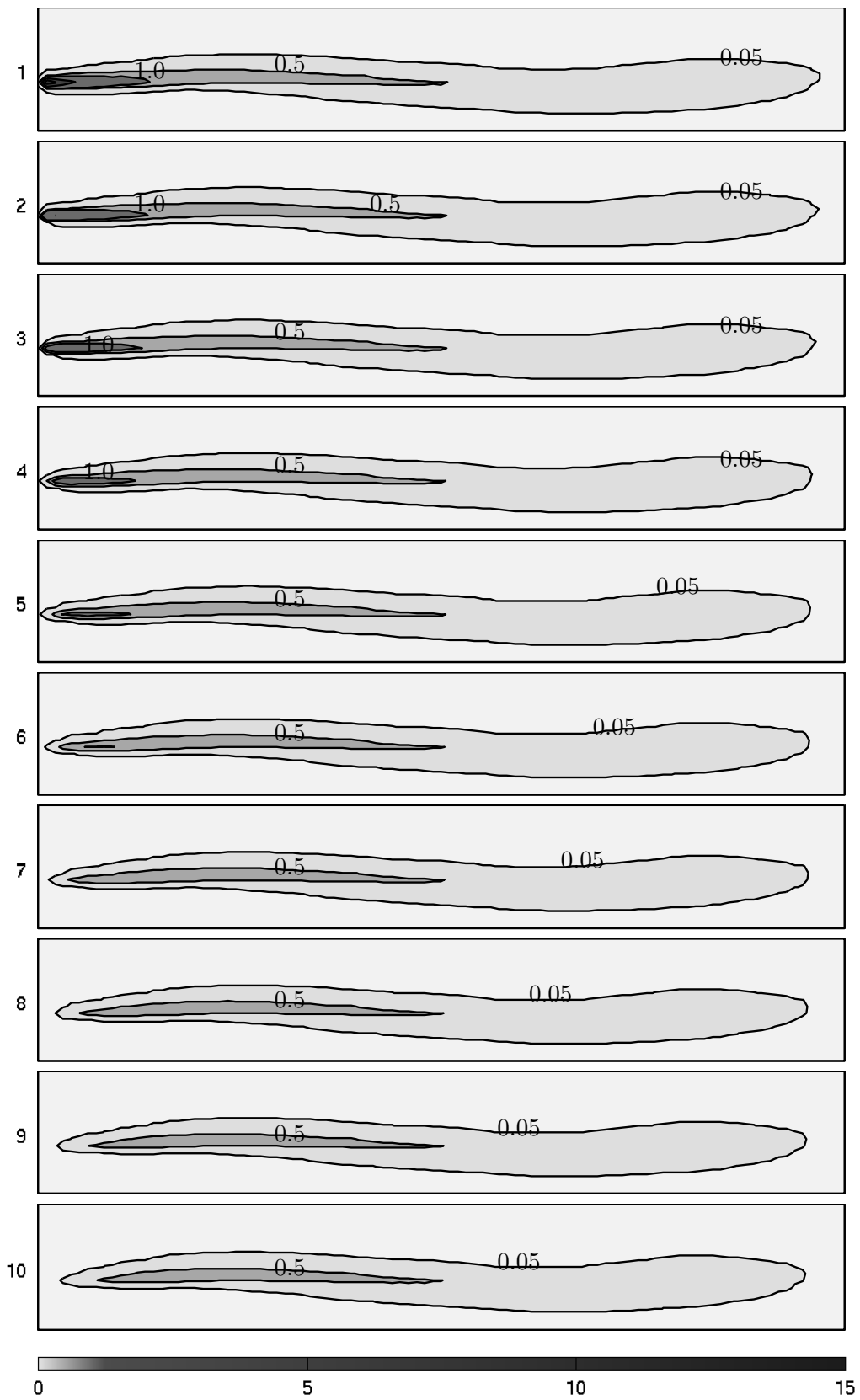


Figure 5.19: Top view of the concentration at different layers after 16 min using parabolic vertical diffusion (profile II).

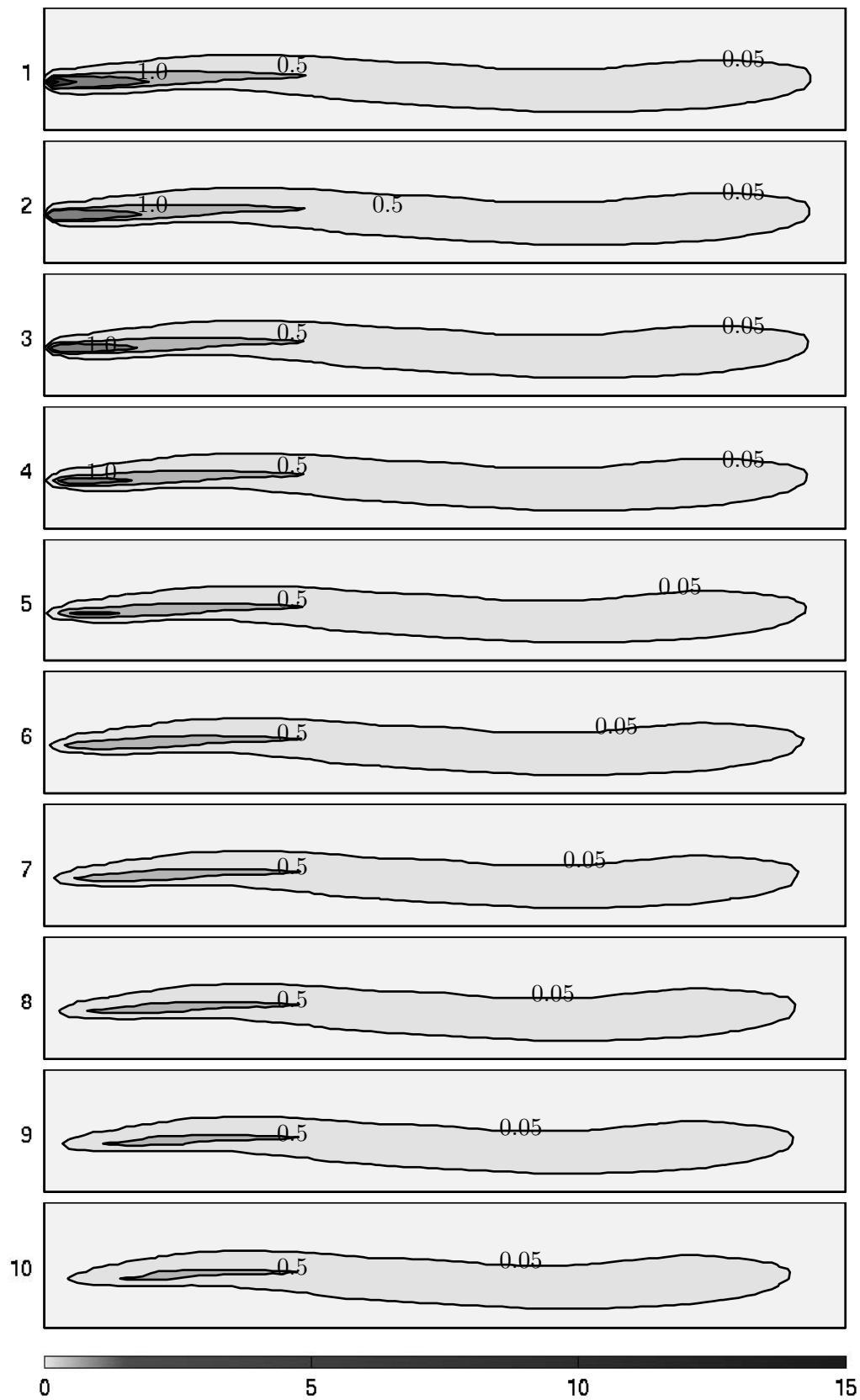


Figure 5.20: Top view of the concentration at different layers after 16 min using the algebraic model for the vertical diffusion.

5.4 Application to the Dutch coast

As mentioned, we applied our 3-D transport solver to the Dutch coastline and simulated a transport scenario in this area. As in chapter 3, we also use the Kuststrook-model, but now with 10 layers forming the vertical dimension.

Compared to hypothetical exercises, real-life applications pose extra difficulties due to all kinds of peculiarities. With regard to the type of problems we are dealing with, these peculiarities can be for example a highly distorted bathymetry or the drying and wetting of some parts of the domain during the simulation period.

5.4.1 The grid

The projection of the grid used here on a horizontal plane corresponds exactly to the grid of figure 3.1. Recall that this grid represent a 400×60 km domain with an average size of one single element of 1000×400 m. The total vertical depth, which varies according to the location from 0 to around 50 m, is divided in 10 layers of equal thickness.

5.4.2 Hydrodynamic coefficients

TRIWAQ was used to provide the necessary flow data. The flow computational time step was 1 minute. The time integrated values of velocities and water depths presented in section 5.2 were computed and stored under the required format into files for every layer and for every time step of the transport simulation taken equal to 10 minutes. The total necessary disk space for the storage was large: around 300 Gb contained in more than 400,000 files for the simulation of a duration of 2 months. We should mention here that the handling and management of such a large number of files is made reasonably feasible thanks to the use of dedicated bash scripts of our Linux-based operating system.

5.4.3 Check for the constancy condition

Using a time step of 10 minutes, the check for the constancy condition was performed successfully. The adopted correction procedure presented in section 5.2 proved to be very convenient for this real-life application especially because flow data do not always satisfy the conservation property up to machine accuracy.

Here also, a check related to term (4.20) was performed. The run of the constant solution problem including this term led to the same conclusion as for the test case: the concentrations remained almost unchanged and the decision to neglect this term is acceptable.

5.4.4 Real-life simulation

Subsequently, we simulated a release in the North Sea is carried out. The point of release was the outlet of the Nieuwe Waterweg in Hoek van Holland (shown by an arrow in figure 5.21). The computational time step was also 10 minutes.

Figure 5.21 shows the spreading of the tracer after one month of the start of the release for both layer 1 and layer 10. Figure 5.22 shows the situation after a period of two months for both layers. These two figures together with figure 5.23, presenting the results in further intermediate layers, show that the distribution of the tracer is not very different from one layer to another. By examining the concentration time history plots in figure 5.24, we see that the vertical distribution is significant only close to the release point (cell 14489). In general, in these plots, the topmost curve corresponds to the top layer. The selected locations for time history plots are depicted in figure 5.25.

These results were obtained using a uniform vertical diffusion $D_v = 0.01 \text{ m}^2/\text{s}$. A further experiment is carried out with the space-varying algebraic model used earlier for the test case and described in appendix A. Since the coastal model presents very non uniform flow parameters, the resulting vertical diffusion varies also considerably between different locations. Nevertheless, the resulting spreading shown in figure 5.26 was very similar to the one obtained with the uniform D_v . This led to the conclusion that the 3-D coastal model is not really sensitive to the vertical distribution of the vertical diffusion. It is even less sensitive to the order of magnitude of D_v than in the test case.

The oscillations observed in these time history plots are realistic and are due to the tidal effects. Notice that the vertical scale is not the same for all the plots.

The result of the check for the tracer mass conservation is shown in figure 5.27. It shows a good match between the released amount of pollutant - chosen to be constant in time - and the total mass resulting from the computed concentrations. We noticed, however, that by the end of simulation, a slight difference occurred.

In terms of positivity, the results are also very satisfactory. Figure 5.28 shows a plot of the overall maximum and minimum values detected through the whole domain as a function of time. The maximum values are usually registered at the point of release. The periodic pattern clearly follows the tidal movement. Also by the end of the simulation, there are some relatively large negative values. The check made after these large values shows that their number is very limited, usually one single negative value. They also take place in regions likely to undergo drying and wetting. This is confirmed by the moment these errors appear when the tracer starts reaching the northern Waddenzee area, well known by the strong tidal effect which dries out a large parts of this region.

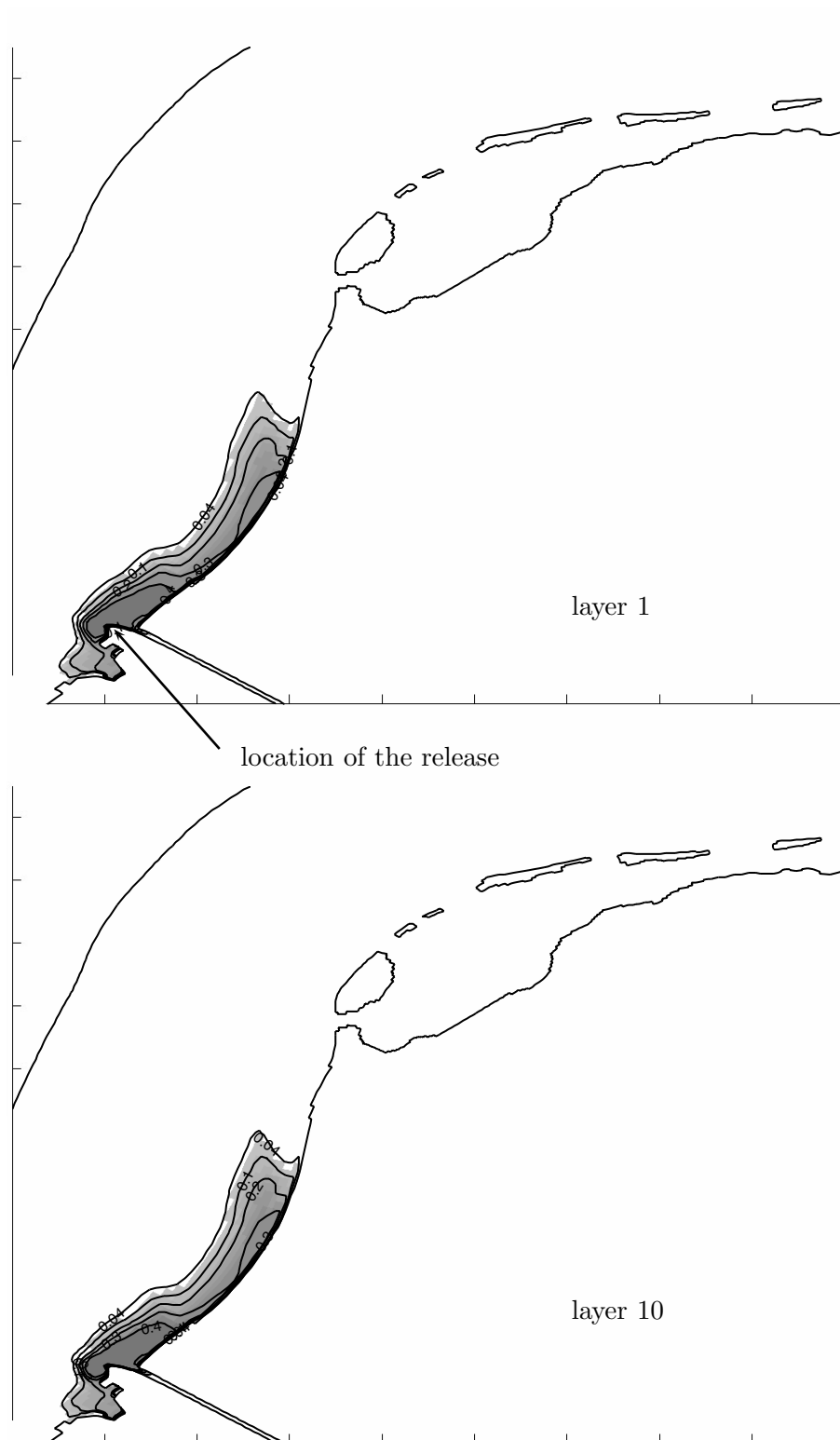


Figure 5.21: Spreading pattern after one month from the start of the release

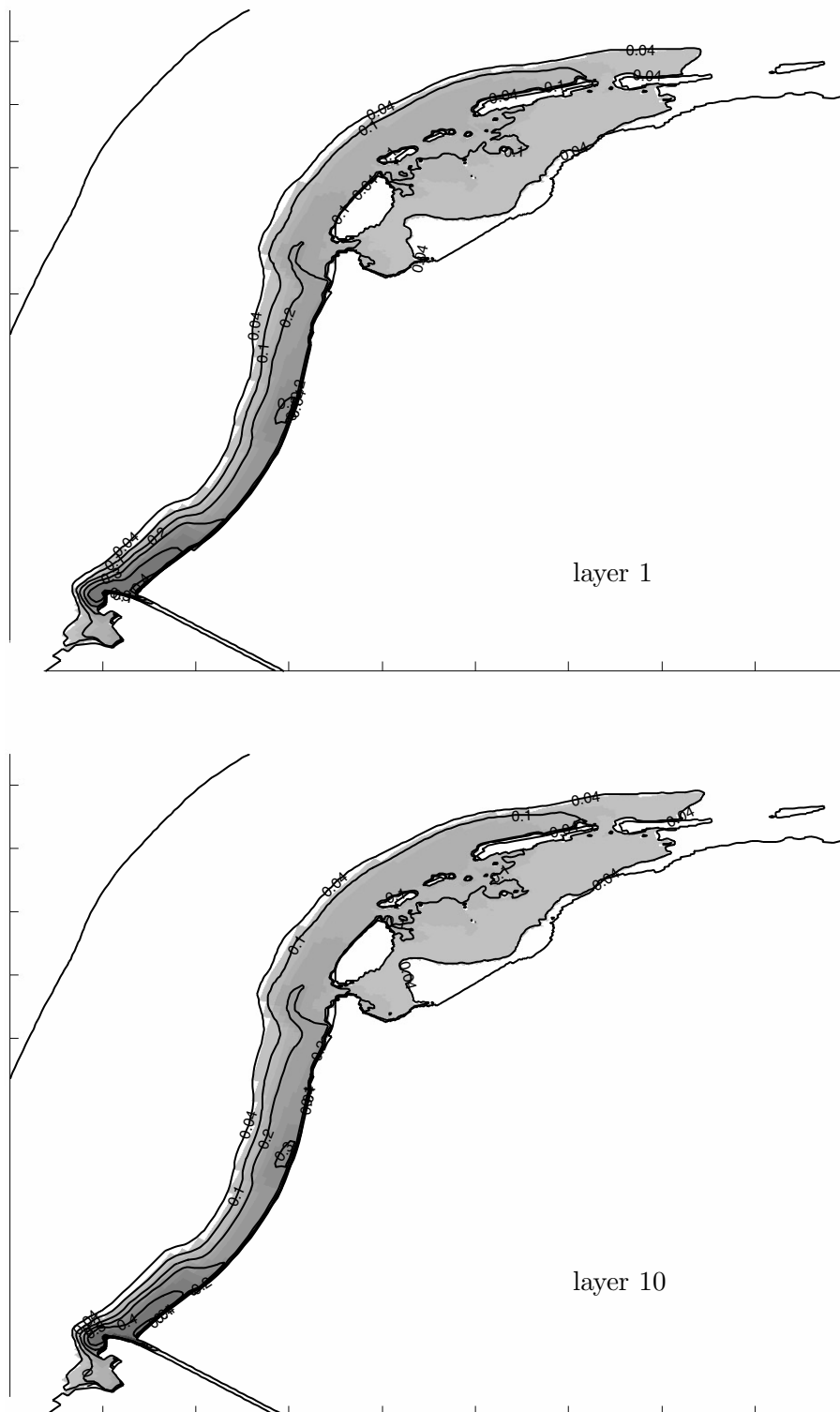


Figure 5.22: Spreading pattern after two months from the start of the release

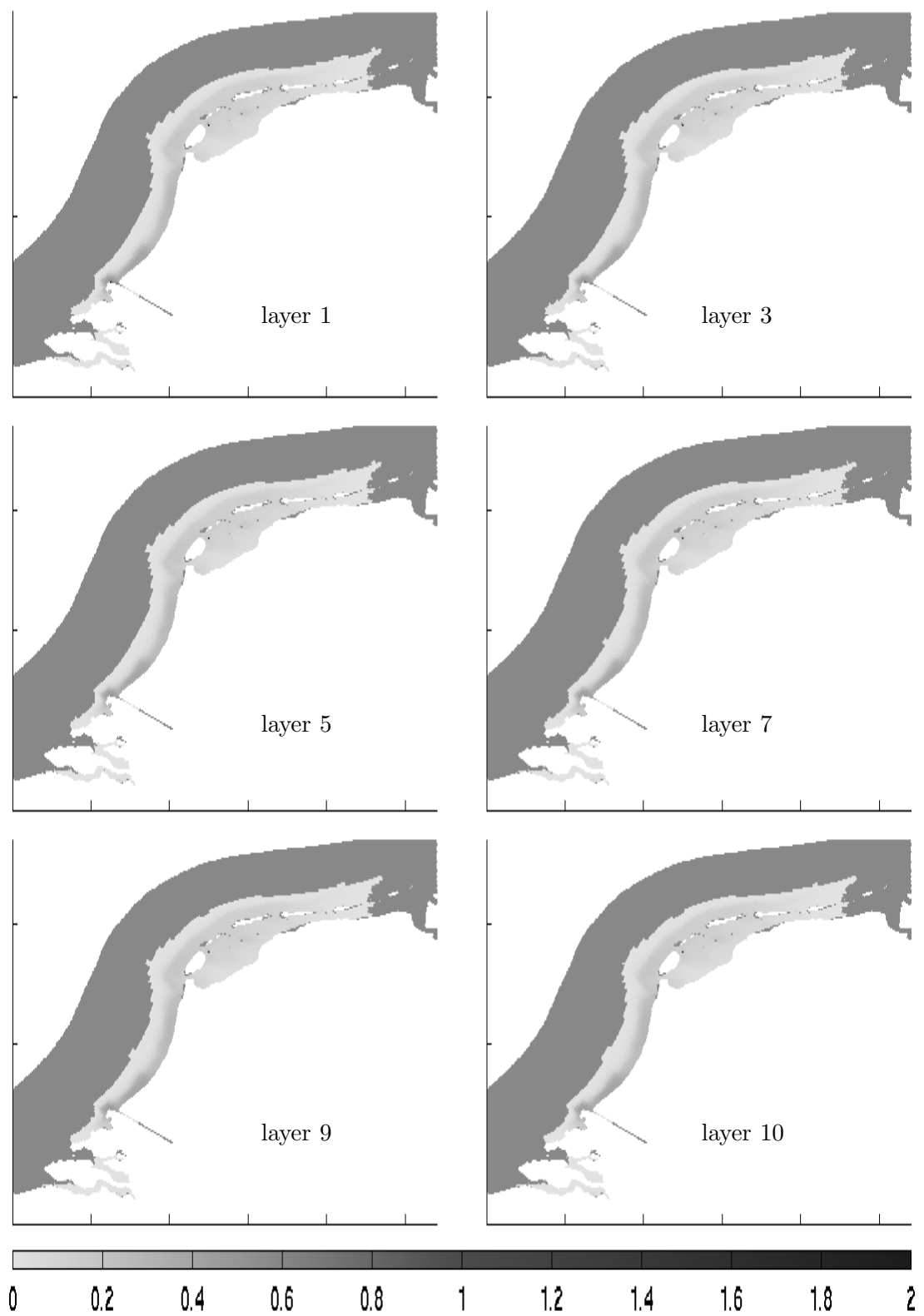


Figure 5.23: Result of a tracer release simulation (after 63 days)

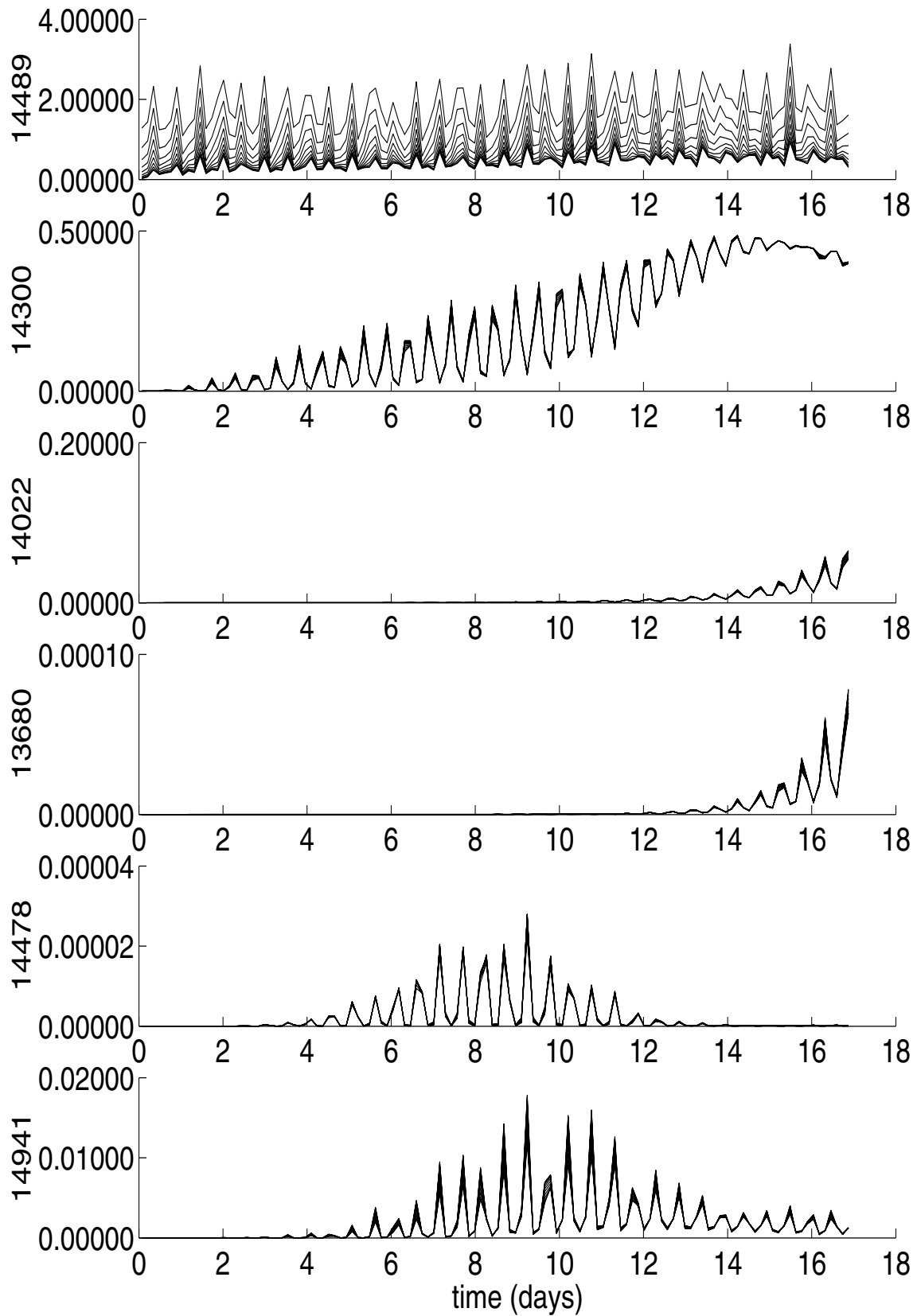


Figure 5.24: Result of a tracer release simulation (after 63 days)

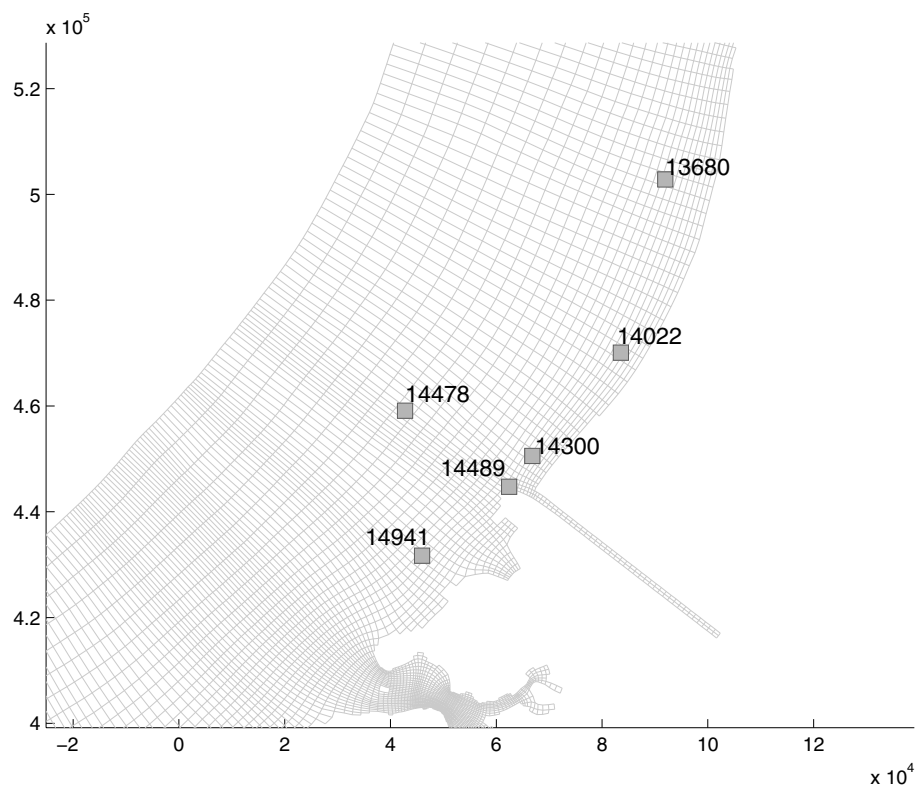


Figure 5.25: Location of time history graphs

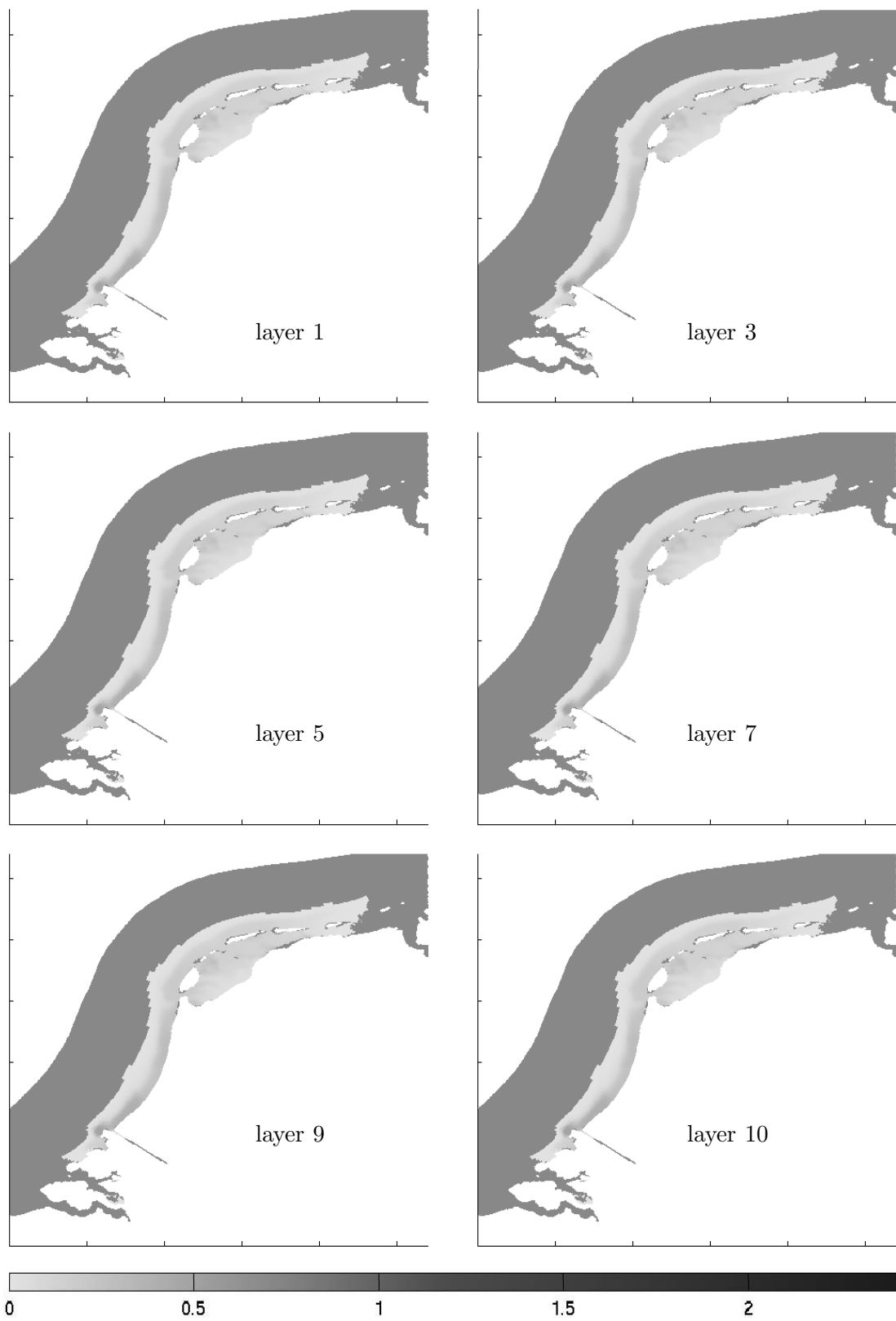


Figure 5.26: Result of a tracer release simulation (after 63 days) using an algebraic model for the vertical diffusion

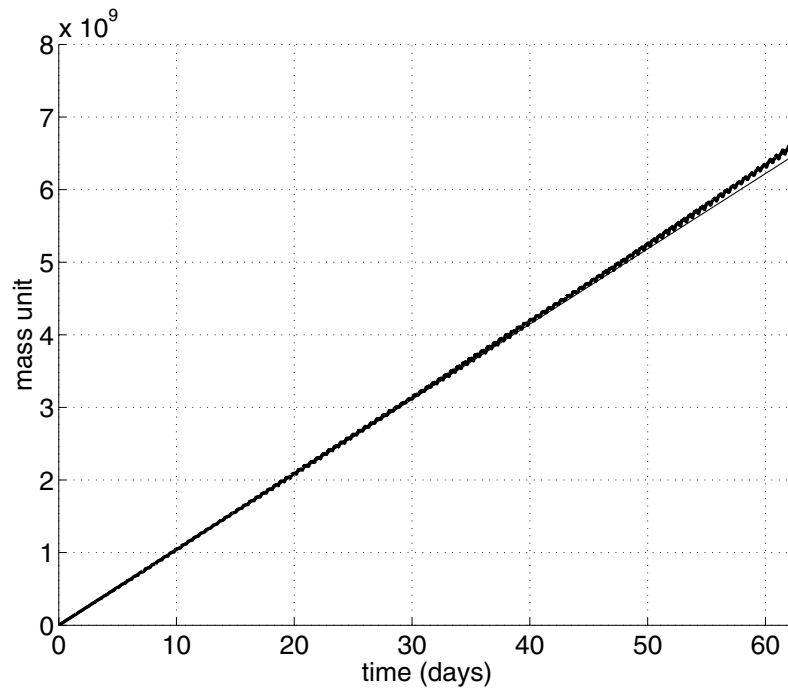


Figure 5.27: Total mass in the domain

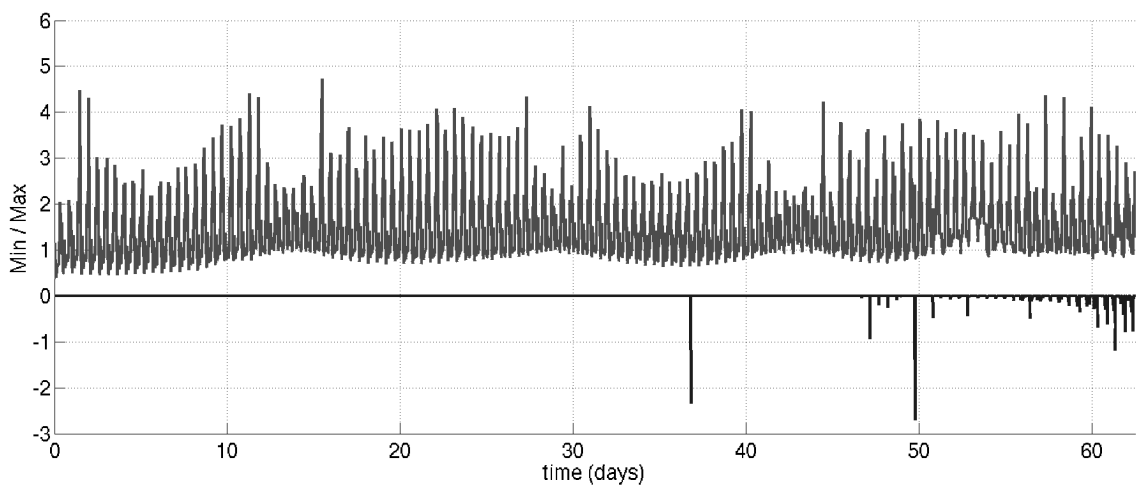


Figure 5.28: Maximum and minimum computed concentrations.

5.5 Conclusions

The newly constructed 3-D model is applied for two cases: a hypothetical test case, specially created for the implementation phase, and a real life case to test for the performance and the capabilities of the model to deal with all sort of difficulties related to real life cases. For both applications, the constancy condition is satisfied up to the machine accuracy thanks notably to the proposed preprocessing technique.

For the tracer release experiment, the results obtained for the test case are compared to the results of the transport package of TRIWAQ. The profile of the tracer spreading is very similar in both experiments. Our solver presents a very good mass conservation property and furthermore it has the advantage of not producing negative concentrations. With respect to the vertical distribution of the concentration, it is found out that the value of the vertical diffusion coefficient is the key parameter which controls this distribution. Three different profiles are tested for the vertical diffusion, namely a constant profile, a parabolic profile and a third profile using an algebraic model which computes the vertical diffusion based notably on the flow parameters. The main conclusion was that the order of magnitude of the vertical diffusion is the most important factor rather than the shape of the profile.

The 3-D solver also performs well when used with the Dutch coastal model. A similar experiment to the one carried on with the 2-D model shows that no significant vertical distribution of the concentration takes place. The spreading in the 10 layers looks very much the same and appears not to be really sensitive to the vertical profile of the vertical diffusion. It is also similar to the spreading of the 2-D case and also to the spreading obtained with the 2-D transport package of WAQUA. The check for the total mass present in the domain confirms the good mass conservation properties of the 3-D solver.

Chapter 6

General conclusions and recommendations

6.1 General conclusion

The main goal of this thesis was to improve the 2-D tracer transport solver developed and to extend it to a three-dimensional model. This work was in fact an extension to the earlier works of P. Wilders and G. Fotia [53, 54] who first applied the finite volume technique on triangular unstructured grids to simulate the transport processes. The solver is second order accurate and uses the Enquist-Osher function for the estimation of the numerical advective fluxes. The trapezoidal rule is used for the time integration. The improvements brought in this thesis to the original model consisted in solving problems related to the mass conservation and to the positivity of the solution. It happened in fact that for some practical applications and despite of the fulfillment of the previously developed conditions for positivity, negative concentrations might still occur and problems in mass conservation were detected. Positivity and mass conservation are important issues necessary to ensure an accurate transport simulation. The main idea of the new improvement was to avoid the appearance of artificial source and sink terms by ensuring the so-called constancy condition which basically states that in absence of external sources, a system with a uniform concentration should keep his concentration uniform as times runs. The following preprocessing steps were then proposed to ensure this constancy condition. These steps constitute at the same time a main conclusion of the first part of this thesis as well as a recommendation to be considered in any future work of the same context.

- The transport solver is uncoupled from the flow computations. Nevertheless, a vigorous check on the quality of the flow output was required. Essentially, the computed flow parameters have to satisfy the discrete flow continuity equation up to machine accuracy. A posteriori check on whether these data do really satisfy the continuity equation is very recommended. If this is not the

case, a correction procedure might be needed to provide exactly conservative coefficients.

- Based on the flow parameters obtained from the flow solver and checked in the previous step, new flow coefficients were computed. We do know the numerical scheme used to compute them and consequently the discrete form of the continuity equation used to compute the flow is also known. Based on that, we proposed a new method to generate another set which satisfy the discrete transport continuity equation.
- This step is rather related to real-life cases where drying and wetting of some regions of the domain can occur. A special attention is then necessary to match the way drying and wetting is conceived between the flow and the transport solvers. This an important and delicate point which may even corrupt the whole computation if not dealt with properly.

The improved 2-D transport solver was applied to the Dutch coast to simulate the propagation of the pollution originating from the river Rhine. After ensuring the steps mentioned above, the output of WAQUA, the flow solver used to provide the flow parameters, was successfully preprocessed and consequently the constancy condition could be satisfied. The results of the simulation of a tracer release were compared to the results from the transport solver of WAQUA and a good correspondence was obtained. The mass conservation was also satisfied during the whole computation.

The choice of uncoupled computation of the flow and transport can lead to a demand of a large disk space exceeding the available possibilities. On the other hand, a constraint on the time step for the transport solver to be the same as the one of the flow might also be imposed. In this thesis, we developed a technique to overcome these disadvantages. It consisted in creating new set of data at relatively large time steps by combining the data computed with smaller time step. This technique led to remarkable results with the valuable advantages of reduced storage and faster computations.

A second investigated technique to reduce the disk space required and to allow for a cheaper flow computation was the so-called periodic continuation. It is based on the assumption of the periodic character of the flow parameters in relation with the tidal changes. This technique showed a high sensitivity to the chosen length of the unit period and furthermore it led to the appearance of negative concentrations. The conclusion with this respect was that the usage of this technique should be limited.

The second part of this thesis was dedicated to the development of a new three dimensional version of the transport solver. The model solves a layer averaged advection-diffusion equation over an unstructured grid with fixed number of layers. The finite volume technique used in the 2-D solver is extended to cope with the multi layered problem.

In order to test and to evaluate the performance of the 3-D model, we did apply it in two different experiments: a hypothetical test case, and a real-life case, the Dutch coastal model.

The test case consisted in simulating the transport of a tracer in a stretch of river with irregular bed forms. The grid contained 10 layers in the vertical direction. The constancy condition was satisfied thanks to the new approach used in the data pre-processing. The result of a tracer release experiment showed a good similarity with the result obtained with the transport module in the TRIWAQ package although the latter produced very slightly less diffusive spreading in the horizontal direction. No negative concentrations were produced and the check for the total mass was satisfactory. The sensitivity analysis carried out for this test case revealed that the magnitude of the vertical diffusion coefficient plays the dominant role in the shape of the vertical profile of the concentration. It was therefore recommended to have a good estimation of this coefficient. Three different models were used to estimate the vertical diffusion: a uniform vertical profile, a parabolic profile and an algebraic model used for common applications. For the first uniform profile, two different values, 0.01 and 0.001 m²/s were tested. For the parabolic profile, two different profile corresponding to two different orders of magnitude were used. The algebraic model had the advantage of computing the vertical advection out of the space varying flow parameters and using empirical variables recommended for practical use. Two main conclusions were drawn from these different experiments. First, the shape of the diffusion profile does not have much effect on the final distribution and that the order of magnitude of D_v is rather the predominant factor. Second, the results of the algebraic model were very similar to ones obtained with the constant and parabolic profiles when D_v is of the order of 0.01 m²/s.

The second application of the 3-D solver was devoted to the Dutch coastline to simulate a transport scenario in this area. In general, compared to test cases, real life applications usually present extra difficulties notably due to the drying and wetting of some parts of the domain during the simulation period.

The same Kuststrookmodel was used here but now with 10 layers forming the vertical dimension. Using a time step of 10 minutes, the check for the constancy condition was successfully performed. With the same time step, a tracer release at the outlet of the Nieuwe Waterweg in Hoek van Holland was simulated. Using a uniform vertical diffusion $D_v = 0.01$ m²/s, the obtained results showed that the distribution of the tracer is not very different from one layer to another. Only in the area close to the release point, the vertical distribution can be significant.

Further experiments were carried out using the space varying algebraic model. Although this model produced a vertical diffusion which varies considerably between different locations, the resulting spreading is very similar to the one obtained with the uniform D_v . The conclusion drawn was that the 3-D coastal model is not really sensitive to the vertical distribution of the vertical diffusion. It was found out also that it is even less sensitive to the order of magnitude of D_v than in the test case.

Further checks of the results with respect to the tracer mass conservation and to the positivity of the concentrations were performed. In general, there was a very good match between total mass released and the total mass computed out of the concentrations during the simulation period. We noticed, however, that by the end of simulation, the occurrence of a slight difference. At the same time, now and then, some negative values appear and usually limited to one single value through out the domain. It is very likely that the drying and wetting treatment is behind the

appearance of these negative values mainly because this fact occurs when the tracer reaches the northern part of the Dutch coast and where the drying and wetting is a dominant characteristic of the the region. Anyway, we believe that this does not alter the overall result which remains very satisfactory and that future minor fixing can eliminate this problem.

6.2 Recommendations

The results obtained in this thesis showed that the developed transport solver has notably the advantage of being mass conservative and producing positive output. There is however room for further improvements to the solver:

- The modeling of the drying and wetting into the transport solver using correction techniques as we did in this thesis might not be completely satisfactory for coastal regions where this phenomenon is highly preponderant. Errors in the mass conservation may then accumulate quickly. More rigorous methods should therefore be considered.
- In its new three dimensional version, the transport solver requires more than 40 hours of CPU time on a single processor computer (AMD Athlon 1700) to achieve the run of the Dutch coastal model for a period of two months. It is therefore good time to think about investigating the parallelization of the code to speed up the simulations.

Bibliography

- [1] B.A. Bakhmetev. *Hydraulics of open channels*. McGraw-Hill, 1932.
- [2] T. Barth and M. Ohlberger. Finite volume methods: foundation and analysis. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1, United Kingdom, 2004. Wiley & Sons.
- [3] T.J. Barth. *On unstructured grids and solvers*. Von Karman Institute for Fluid Dynamics. Lecture Series, 90-03, Rhode-Saint-Genese, Belgium, 1990.
- [4] M. Berzins. Positive cell-centered finite volume discretization methods for hyperbolic equations on irregular meshes. *Applied Numerical Mathematics*, 16:417–438, 1995.
- [5] L. Bonaventura, L. Kornbluh, M. A. Giorgetta, E. Roeckner, T. Heinze, D. Majewski, P. Ripodas, B. Ritter, T. Ringler, and J. Baudisch. The ICON shallow water model: scientific documentation and benchmark tests. Technical report, Max Planck Institut für meteorologie, Hamburg, Germany, 2004.
- [6] F.D. Bramkamp. *Unstructured h-adaptive finite-volume schemes for compressible viscous fluid flow*. PhD thesis, Rheinisch-Westfälischen Technischen Hochschule Aachen, July 2003.
- [7] B. Cockburn, S.Y. Lin, and C.W. Shu. TVD Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws iii: one dimensional system. *Journal of computational physics*, 84:90–113, 1989.
- [8] M. Colombini and A. Stocchino. 3D transport of a passive pollutant for wind-driven flows in enclosed basins. In G.H. Girka and W.S.J. Uijtewaal, editors, *Shallow flows*, pages 429–437, Delft, The Netherlands, June 2003. Taylor & Francis, London, 2004.
- [9] M.G. Crandall and A. Majda. Monotone difference approximations for scalar conservation laws. *Mathematics of computation*, 34(149):1–21, 1980.
- [10] C. Dawson. Conservative, shock-capturing transport methods with nonconservative velocity approximations. *Computational geosciences*, 3:205–228, 2000.

-
- [11] C. Dawson, S. Sun, and M.F. Wheeler. Compatible algorithms for coupled flow and transport. *Computer methods in applied mechanics and engineering*, 193(23-26):2565–2580, 2004.
- [12] C. de Blois. Rhine transport model, dynamic pollutant transport modeling for policy evaluation for policy evaluation in the rhine basin. Technical report, Delft University of Technology, The Netherlands, March 1993.
- [13] K. de Jong. *Tidally averaged transport models*. PhD thesis, Delft University of Technology, October 1998.
- [14] M. Delanaye. *Polynomial reconstruction finite volume schemes for the compressible Euler and Navier-Stokes equations on unstructured adaptive grids*. PhD thesis, Université de Liège, September 1996.
- [15] E. Deleersnijder. Enforcing the continuity equation in numerical models of geophysical fluid flows. *Applied mathematics letters*, 14:867–873, 2001.
- [16] L.J. Durlofsky. Triangle based adaptive stencils for the solution of hyperbolic conservation laws. *Journal of computational physics*, 98:64–73, 1992.
- [17] L.J. Durlofsky. A triangle based mixed finite element-finite volume technique for modelling two phase flow through porous media. *Journal of computational physics*, 105:252–266, 1993.
- [18] EDS. TRIWAQ: Three dimensional shallow water flow model, technical documentation. Technical Report 99-01, Ministry of Transport, Public Works and Water Management. Directorate-General for Public Works and Water Management, Leidschendam, 1999.
- [19] EDS. User’s guide WAQUA. general information. Technical report, Ministry of Transport, Public Works and Water Management. Directorate-General for Public Works and Water Management, Leidschendam, April 2000.
- [20] I. El Mahi. *Schémas volumes finis pour la simulation numérique de problèmes à fronts raides en maillages non structurés adaptatifs*. PhD thesis, Université De Rouen, 1999.
- [21] E.S. Gross, L. Bonaventura, and G. Rosatti. Consistency with continuity in conservative advection schemes for free-surface models. *International Journal for Numerical Methods in Fluids*, 38:307–327, 2002.
- [22] W. Hunsdorfer, B. Koren, M. van Loon, and J Verwer. A positive finite-difference advection scheme. *Journal of computational physics*, 117:35–46, 1995.
- [23] R. J. Leveque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM journal on numerical analysis*, 33(2):627–665, 1996.

- [24] Q. Liang, A.G.L. Borthwick, and J. Huang. Godunov-type quadtree model of species dispersion in shallow flows. In G.H. Girka and W.S.J. Uijtewaal, editors, *Shallow flows*, pages 439–447, Delft, The Netherlands, June 2003. Taylor & Francis, London, 2004.
- [25] S.J. Lin and R.B. Rood. Multidimensional flux-form Semi-Lagrangian transport schemes. *Monthly Weather Review*, 124:2046–2070, 1996.
- [26] S.Y. Lin, T.M. Wu, and Y.S. Chin. Upwind finite volume method with a triangular mesh for conservation laws. *Journal of computational physics*, 107:324–337, 1993.
- [27] X.D. Liu. A maximum principle satisfying modification of triangle based adaptive stencils for the solution of scalar hyperbolic conservation laws. *SIAM journal on numerical analysis*, 30(3):701–716, 1993.
- [28] J.L. Martin and S.C. McCutcheon. *Hydrodynamics and transport for water quality modeling*. Lewis, Boca Raton, Florida, 1999.
- [29] D.J. Mavriplis. Unstructured grid techniques. *Annual Review of Fluid Mechanics*, 29:473–514, 1997.
- [30] A. Mazzia and M. Putti. Three dimensional Godunov mixed methods on tetrahedra for the advection-dispersion equation. In S.M. Hassanizadeh et al., editor, *Developments in water science. Computational methods in water resources*, volume 1, pages 381–388, Delft, The Netherlands, June 2002. Elsevier, Amsterdam, 2002.
- [31] F. Naifar and P. Wilders. A 2DH flexible finite volume transport solver for unstructured grids. Application to pollution along the Dutch coast. In M.S. Altinakar, S.S.Y. Wang, K.P. Holz, and M. Kawahara, editors, *Advances in hydro-science and -engineering*, volume VI, pages 1–10, Brisbane, Australia, May-June 2004. the University of Mississippi, USA, 2004. CD Rom proceedings, ISBN 0-937099-12-0.
- [32] F. Naifar and P. Wilders. Towards a conservative and positive computation of pollutant transport. In P. Neittaanmaki, T. Rossi, and K. Majava, editors, *4th European congress on computational methods in applied sciences and engineering*, pages 1–21, Jyväskylä, Finland, July 2004. CD Rom proceedings, ISBN 951-39-1868-8.
- [33] S.P. Neuman. Adaptive Eulerian-Lagrangian finite element method for advection-dispersion. *International journal for numerical methods in engineering*, 20:321–337, 1984.
- [34] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM journal on numerical analysis*, 21(2):217–235, 1984.
- [35] B. Perot. Conservation properties of unstructured staggered mesh schemes. *Journal of Computational Physics*, 159:58–89, 2000.

-
- [36] R.J. Randall. *Finite volume methods for hyperbolic problems*. Cambridge, United Kingdom, 2002.
- [37] W. Rodi. *Turbulence models and their application in hydraulics - A state of the art review*. IAHR, Delft, 1984.
- [38] P.L. Roe and D. Sidilkover. Optimum positive linear schemes for advection in two and three dimensions. *SIAM Journal on numerical analysis*, 29:1542–1568, 1992.
- [39] Y. Saad. *Iterative Methods for sparse linear systems*. Siam, Philadelphia, USA, 2003.
- [40] W. Salomons, B.L. Bayne, E.K. Duursma, and U. Förstner. *Pollution of the North Sea. An assessment*. Springer-Verlag, 1989.
- [41] M Shashkov, B. Swartz, and B. Wendroff. Local reconstruction of a vector field from its normal components on the faces of grid cells. *Journal of computational physics*, 139:406–409, 1998.
- [42] G. Stelling. *On the construction of computational methods for shallow water flow problems*. PhD thesis, Delft University of Technology, 1983. Also available in Rijkswaterstaat Communications No. 35/1984.
- [43] J.W. Stijnen. *Numerical methods for stochastic environmental models*. PhD thesis, Delft University of Technology, 2002.
- [44] R.E. Uittenbogaard, J.A.Th.M. van Kester, and G.S. Stelling. Implementation of three turbulence models in TRISULA for rectangular horizontal grids. Technical Report Z162, Delft Hydraulics, Delft, 1992.
- [45] S. van der Baan and P. Wilders. Flexible finite volumes for tracer transport in coastal regions. Technical Report ISSN 1389-6520, Department of applied mathematical analysis, Delft, 2628 CD, September 2000.
- [46] S. van der Baan, P. Wilders, and F. Naifar. Flexible finite volumes for tracer transport in coastal regions. In A. Heemink, L. Dekker, H. de Swaan Arons, I. Smit, and T. van Stijn, editors, *Shaping future with simulation. Proceedings 4th Eurosim 2001 congress*, pages 1–7, Delft, The Netherlands, June 2001. DBSS, 2001. CDRom proceedings, ISBN 90-806441-1-0.
- [47] V. Venkatakrishnan. Perspective on unstructured grid flow solvers. *AIAA J.*, 34:533–547, 1996.
- [48] V. Venkatakrishnan and T.J. Barth. Application of direct solvers to unstructured meshes for the Euler and Navier-Stokes equations using upwind schemes. *AIAA*, 1989-364, 1989. Presented at the 27th aerospace sciences meeting, Jan. 9-12, Reno, NV.

-
- [49] D. Vidović. *Superlinearly convergent unstructured staggered schemes for compressible and incompressible flows*. PhD thesis, Delft University of Technology, 2005.
- [50] D. Vidovic, A. Segal, and P. Wesseling. A superlinearly convergent finite volume method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 198:159–177, 2004.
- [51] C.B. Vreugdenhil. *Numerical Methods for Shallow-Water Flow*. Taylor & Francis, New Palm Beach, Florida, 1995.
- [52] P. Wesseling. *An introduction to multigrid methods*. Wiley, New York, 1992.
- [53] P. Wilders and G. Fotia. Distributed implicit stepping with unstructured finite volumes for 2D-transport. *Journal of Computational and Applied Mathematics*, 82:433–446, 1997.
- [54] P. Wilders and G. Fotia. A positive spatial advection scheme on unstructured meshes for tracer transport. *Journal of Computational and Applied Mathematics*, 140:809–821, 2002.
- [55] G.T. Yeh. A Lagrangian-Eulerian method with zoomable hidden fine mesh approach to solving advection-dispersion equations. *Water resources research*, 26 (6):1133–1144, 1990.
- [56] Ch. Zheng and G.D. Bennet. *Applied contaminant transport modeling*. Wiley Inter-Science, 2002.
- [57] M. Zijlema and P. Wesseling. Higher-order flux-limiting schemes for the finite volumes computation of incompressible flow. *International Journal of Computational Fluid Dynamics*, 9:89–109, 1998.

Appendix A

An algebraic turbulence model for the vertical diffusion

The vertical diffusivity D_v is assumed to be proportional to the eddy viscosity ν_v :

$$D_v = \frac{\nu_v}{\sigma_c} \quad (1.1)$$

where σ_c is the turbulent Prandtl or Schmidt number assumed in most cases to be constant and for the transport of a tracer, it is set to 0.7.

The so-called zero-equation or algebraic model determine the eddy viscosity through empirical formulae for k and L , which are suitable for shallow flow applications.

The so-called algebraic models are the simplest of all turbulence models. The model used in chapter 5 is based on the Kolmogorov-Prandtl relation:

$$\nu_v = c'_\mu \sqrt{k} L \quad (1.2)$$

where L is a specific length scale and c'_μ is an empirical constant. Details may be found in [37]. According to Uittenbogaard et al. [44], $c'_\mu = 0.58$. The length scale L , which depends on the flow geometry, needs to be prescribed. According to [1], the following algebraic expression for L suitable for hydraulic applications can be used:

$$L = \kappa (z + d) \sqrt{\left(1 - \frac{z + d}{H}\right)} \quad (1.3)$$

with κ the Von Kármán constant (≈ 0.41). It follows from this equation that at the bottom and free surface the length scale equals zero. If we neglect the wind stress, we can use the following expression for k proposed by Uittenbogaard et al. in [44]:

$$k = \frac{1}{\sqrt{c_\mu}} \left(1 - \frac{z + d}{H}\right) (u_*^b)^2 \quad (1.4)$$

where $u_*^b = \sqrt{\frac{|\tau_b|}{\rho}}$ is the friction velocity at the bottom and τ_b is the bottom stress. c_μ is a closure constant taken equal to 0.09.

Substituting (1.2), (1.3) and (1.4) into (1.1) leads to a parabolic vertical diffusion profile across the total water depth with a maximum located at half the depth. For the estimation of the friction velocity, we can use the logarithmic wall-law for rough beds expressed as:

$$(u_*^b) = \kappa \frac{u\left(\frac{\Delta z}{2}\right)}{\ln\left(\frac{\frac{\Delta z}{2} + z_o}{z_o}\right)} \quad (1.5)$$

where $u\left(\frac{\Delta z}{2}\right)$ corresponds to the norm of the velocity vector at the center of the cell of the bottom layer. z_o is a roughness coefficient ($\approx 0.0025m$). This leads therefore to a space varying vertical diffusion.

With respect to our quadrilateral unstructured grid, we only have normal velocities at cell faces and therefore we need to estimate the velocity at cell centers. We propose a simple method to obtain this estimation. Illustrated in figure A.1, it consists in adding all the velocity vectors at faces and taking half of the norm of the resulting sum vector.

$$u\left(\frac{\Delta z}{2}\right) = \frac{1}{2} \left\| \sum_e \vec{u}_e \right\| \quad (1.6)$$

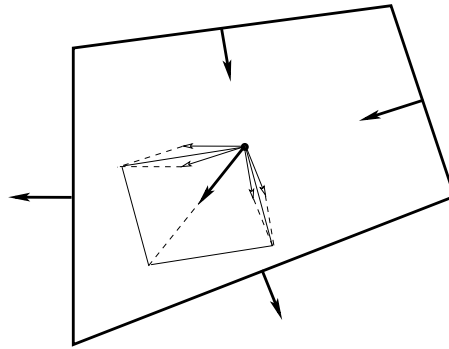


Figure A.1: Reconstruction of the velocity vector at the cell center.

This is a very simple procedure to approximate the velocity vector at the cell center. In our case, in fact, there is no real need for more elaborate methods. However, references [49, 50, 41] are recommended to the reader in case there is necessity for more accurate methods.

Summary

The simulation of the tracer transport process is an increasingly important tool to predict the distribution of pollution released in coastal waters. The equation that governs the tracer transport is the advection-diffusion equation. Thanks to the tremendous developments in computer resources and in computing techniques, it has become possible to solve this equation for multi-dimensional problems and in complex domains.

This thesis, devoted to this subject of tracer transport, consists of two main parts. The first deals with the existing 2-D transport solver and emphasizes notably on the improvements brought to this solver. Based on the finite volume method, the solver uses input data obtained from a separate flow solver. It was shown that a preprocessing step of the flow parameters according to newly developed techniques is required in order to ensure the positivity of the computed concentrations and the mass conservation of the overall solution. In order to be able to adopt a different time step in the transport solver than used for the flow computation, the usage of time integrated flow data has proven to be an efficient technique. The improvements brought to the 2-D solver were tested through the numerical application to a real-life case. The Dutch coastline case was chosen because of its importance from the environmental point of view.

In the second part of the thesis, a 3-D version of the transport solver is developed. In the vertical direction, the physical domain is subdivided in a fixed number of layers, while in the horizontal direction, the domain is represented using an unstructured grid. The numerical estimation of the different fluxes of the adopted finite volume method are described in detail. The 3-D version requires also the preprocessing of the input flow data prior to their use in order to avoid the occurrence of unrealistic sources and sinks.

The testing of the newly developed 3-D solver is achieved using two numerical applications. The first is a hypothetical test case consisting of simulating a tracer release in a river stretch and the second is a real-life application concerning the Dutch coastline. Different vertical profiles for the vertical diffusion coefficient were tested since this term plays an important role in the vertical mixing.

Samenvatting

Een belangrijke methode om de verspreiding van vervuilingen in water te voorspellen is het simuleren van het transport van passieve opgeloste stoffen (tracer transport). De vergelijking die de het tracer transport beschrijft is de zogenoemde advection-diffusie vergelijking en simulatie vindt plaats op basis van een numerieke solver die benaderingen van oplossingen van deze vergelijking genereert. Mede door computer technische ontwikkelingen is het tegenwoordig mogelijk om deze vergelijking op te lossen op complexe domeinen en ook voor multi-dimensionale problemen. Dit proefschrift behandelt tracer transport in de Nederlandse kustzone.

Dit proefschrift valt uiteen in twee delen. Het eerste deel gaat uit van de reeds bestaande 2-D transport solver en behandelt met name de aangebrachte verbeteringen. De transport solver gebruikt data die extern verkregen zijn, in ons geval met het pakket TRIWAQ, een pakket dat stromingen uitrekent. Aangetoond wordt dat voorbewerking van de stromingsgegevens nodig is om positieve waarden van de berekende concentraties te garanderen en om verzekerd te zijn van massabehoud van de oplossing. Hiertoe worden technieken ontwikkeld. Daarnaast wordt aangetoond dat stromingsgegevens samengenomen kunnen worden (time integrated flow data) en dat dit aanleiding geeft tot een doeltreffende techniek om de tijdstap in de transport solver verschillend te kunnen kiezen van de tijdstap in de stromingsberekening. De verbeteringen van de 2-D transport solver zijn getest door middel van een numerieke toepassing op een probleem rond de Nederlandse zee kust. Er is voor de Nederlandse kust gekozen omdat milieuvervuilingen hier van groot belang zijn.

In het tweede deel van dit proefschrift wordt een 3-D variant van de transport solver ontwikkeld. In verticale richting is het domein verdeeld in een eindig aantal lagen. In horizontale richting daarentegen is de verdeling ongestructureerd. Toepassing van de eindige volume methode vraagt om evaluatie van verschillende fluxen. De gekozen numerieke benaderingen worden gedetailleerd beschreven. De 3-D versie vereist ook een voorbewerking van de stromingsgegevens om het optreden van onrealistische bronnen en putten te vermijden.

Gebruikmakende van twee numerieke toepassingen is de nieuwe 3-D solver getest. In de eerste toepassing is een hypothetisch rivier segment genomen met een continue lozing. In het tweede geval is de solver toegepast op een probleem met betrekking tot de Nederlandse zee kust. Verschillende profielen voor de verticale diffusie coëfficiënt zijn getest, omdat de verticale diffusie term de primaire aandrijving vormt van de verticale menging.

Curriculum Vitae

Fahmi Naifar was born in Sfax, Tunisia on March 10, 1964. After he completed his high-school education, he continued his studies at the civil engineering department of the “Ecole Nationale d’Ingénieurs de Tunis” where he graduated in 1988 as engineer in hydraulics. Afterwards, he started his engineering career at the flood control section of the Tunisian Ministry of Equipment and Housing. In parallel with his daily work, he followed a DEA (Diplôme d’Etudes Approfondie) course in applied mechanics, transfer of heat and mass in the “Faculté des Sciences de Tunis” and graduated in July 1992. In 1998, he received a scholarship from the Dutch government to follow an International Master course at UNESCO-IHE in Delft, the Netherlands. There, he obtained a Master of Science degree with distinction in hydrology and water resources. Starting from November 2000, he enrolled at Delft University of Technology as PhD researcher in the Department of Applied Mathematical Analysis, nowadays part of the Delft Institute of Applied Mathematics.