INTEGRATED 2D-3D FREE SURFACE HYDRO-ENVIRONMENTAL MODELLING

A thesis submitted to the Cardiff University
In candidature for the degree of
Doctor of Philosophy

by:

Ulrich Reza Kamalian

B.Sc., M.Sc.

Cardiff School of Engineering
Cardiff University

August 2011

DECLARATION

This work has not previously been accepted in substance for degree and is not concurrently being submitted in candidature for any degree.

Signed ………………………………………………………………………………(candidate)
Date ……………………………………………………………………………

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated.
Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ………………………………………………………………………………(candidate)
Date ……………………………………………………………………………

STATEMENT 2

I hereby give consent for my thesis, if accepted to be available for photocopying and for interlibrary loan, and for the title and summary to made available to outside organisations.

Signed ………………………………………………………………………………(candidate)
Date ……………………………………………………………………………

ABSTRACT

An integrated horizontally two- and fully three-dimensional numerical model system has been developed based on a combined unstructured and σ-coordinate grid to simulate the flow and water quality process in large water bodies with a focus on the three dimensional behaviours at specific areas. The model is based on the time dependent Reynolds-Averaged Navier-Stokes equations with a non-hydrostatic pressure distribution and a baroclinic force being incorporated in the three dimensional (3D) model. The two sub models interact dynamically during the solution procedure with no time-step restriction due to integration. The main idea is to use a fractional step algorithm for each model and then integrate the two models fraction by fraction. Hybrid 2D-3D finite volume cells have been introduced for the link nodes which are partly in the 2D domain and partly in the 3D domain. Thus an interpolation/averaging procedure at the interface and domain overlapping is no longer needed.

The 3D model uses the projection method for pressure calculation. The advection equation is solved by the semi-Lagrangian method. Other components are solved via the finite element - finite volume (FV) method. The water surface is determined implicitly through a global matrix equation created by assembling the domain's matrices. The cell integrals are calculated analytically to eliminate a common source of numerical diffusion due to the use of approximation techniques for the FV integrals.

The horizontal gradients of the density and shear stresses are calculated on true horizontal planes, in order to avoid artificial velocity and diffusion in highly stratified flows. Neumann interpolation elements with virtual nodes have been introduced at Neumann type of boundaries for more accuracy.

The integrated model has been verified using analytical solutions and benchmark test cases, including the Ekman velocity distribution, wind driven circulation, lock exchange and integrated 2D-3D flows in basin. The results show the model is capable of the model for accurate simulation and implicit 2D-3D integration.

Keywords: integrated modelling, hydrodynamic numerical model, non-hydrostatic, unstructured mesh, hybrid finite element finite volume method.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors Professor Binliang Lin and Professor Roger A. Falconer for the help, guidance and support they were given during this research project.

I am eternally grateful to my parents and my wife for their continuous love, enormous support and encouragement.

Table of Contents

List of Figures

Index of Tables

Notations

A_{cell}	control volume surface area
A_{heat}^{rad}	heat exchange with the atmosphere
$\vec{a}_c$	Coriolis acceleration; the Coriolis force
a_{GAS}	coefficient for the GAS coordinate system
B	width of the interface that connects two models
b_{GAS}	coefficient for the GAS coordinate system
C_w	wind drag coefficient
C_{ch}	Chezy coefficient for the bed roughness
c_{sm}	constant to be calibrated for the Smagorinsky formulation of eddy viscosity
c_{cell}	filled fraction of a VOF cell
c_{heat}^{conv}	convection heat transfer coefficient equal to 1.41x10^{-3}
c_{air}	specific heat of air equal to 1007 J/(kg ^{o}K)
c_{water}	specific heat of water equal to 4186 J/(kg ^{o}K)
c	suspended sediment concentration
c_b	near-bed suspended sediment concentration
c_i	concentration of the ith sediment fraction
c_{floc}	suspended sediment concentration at which flocculation appears
c_{hind}	suspended sediment concentration at which hindered settling appears
c_{gel}	suspended sediment concentration due to gelling point
D	total water depth
D_t	diffusion coefficient
D_h	horizontal diffusion coefficient
$\bar{D}_h$	horizontal diffusion coefficient for the depth averaged salinity/temperature eq.
D_v	vertical diffusion coefficient
d_0	normal distance between the real bed and the model bottom boundary point
d_{Julian}	Julian day of the year
E_{ec}	eccentricity in the solar orbit
E_{heat}	evaporation effect on the heat exchange
$E_{j,total}$	erosion rate of the jth bed layer
$E(\kappa_e)$	turbulence energy spectrum
E_{time}	discrepancy in time due to solar orbit
E_{vap}	evaporation rate
e_{sat}^{vap}	saturated vapour pressure (kPa)
$\vec{F}$	vector sum of all the external forces
f	Coriolis factor
g	Gravity acceleration
H_{GAS}	modified depth factor for the GAS sigma-coordinate transformation
k_s	bed roughness height
k	turbulence kinetic energy
L_s^{tz}	standard longitude for the time zone
L_l^{tz}	local longitude (in degrees)

l_{m}	turbulence mixing length
l_{sm}	length scale or the Smagorinsky formulation of eddy viscosity
l_{wet}	wetted perimeter at the interface that connects two models
$\vec{n}$	unit normal vector
n	Manning number
$n_{sunRise}$	number of sunrise hours
$n_{sunRise}^{max}$	maximum number of sunrise hours
P_{r}	Prandtl number
p	total pressure
p_{a}	atmospheric pressure
p_{dep}	probability function for suspended sediment deposition
$p_{erosion}$	probability function for bed erosion
Q_{dis}	source discharge
Q_{1d}	flow discharge in one-dimensional modelling
Q_{heat}^{surf}	surface net heat flux
q	hydrodynamic part of the pressure
$\hat{q}$	correction to the hydrodynamic pressure ($\hat{q} \equiv q^{n+1} - q^{n}$)
q_{heat}^{vap}	heat flux due to the water vaporization
q_{heat}^{conv}	heat flux due to the convection
$q_{heat}^{shortRad}$	net heat loss due to the short wave radiation
$q_{heat}^{longRad}$	net heat loss due to the long wave radiation
$q_{hourly}^{shortRad}$	amount of the average hourly short radiation
R_{1d}	hydraulic radius in one-dimensional modelling
R_{h}	relative humidity
Ri	Richardson Number
R_{pr}	precipitation rate
S_{cell}	control volume surface
S	water salinity
$\bar{S}$	depth averaged water salinity
S_{ch}	Schmidt number
S_{dis}	salinity of the source
T	water temperature
$\bar{T}$	depth averaged water temperature
T_{dis}	temperature of the source
t_{local}	local time in hours
U	depth averaged velocity in x-direction
U_{*max}	maximum of the bed and the free surface shear velocities
U_{*}	shear velocity
$\vec{u}$	flow velocity vector
u	x-component of the flow velocity
$\acute{u}$	intermediate velocity component of the solution procedure
u_{dis}	velocity component of the discharged source in the x-direction
u_{est}	intermediate velocity (estimated) in the projection method.

$\vec{V}$	velocity vector in the geographical coordinate system (East-North-up)
V	depth averaged velocity in y-direction
V_{cell}	volume of computational cell
$\vec{V}_b$	near-bed velocity vector
v	y-component of the flow velocity
$\acute{v}$	intermediate velocity component of the solution procedure
v_{dis}	velocity component of the discharged source in the y-direction
W	wind speed relative to water velocity
W_x	x-component of the wind speed
W_y	y-components of the wind speed
$W_{2\mathrm{m}}$	wind speed at two meters above the water surface
$W_{10\mathrm{m}}$	wind speed at 10 meters above the water surface
w	z-component of the flow velocity (vertical velocity)
$\tilde{w}$	transformed vertical velocity in the sigma-coordinate system
w_s	suspended sediment settling velocity
$w_{s,i,const}$	constant settling velocity for the ith sediment fraction when the concentration is low
x, y, z	Cartesian coordinate system axis
$\tilde{x}, \tilde{y}, \tilde{z}$	sigma coordinate system axis
α_{heat}^{refl}	reflection coefficient for the net heat loss due to the short wave radiation
$\alpha_{h,j}^{erosion}$	calibration factor for the hard bed erosion
$\alpha_{s}^{erosion}$	calibration factor for the soft bed erosion
β	correction factor in shallow water equation due to non-homogeneity in vertical distribution of velocity
β_c	artificial compressibility parameter
β_{GAS}	first control coefficient for the influence of the variable ϕ on the vertical grid distribution (GAS coordinate system)
$\beta_{heat}^{infraRed}$	fraction of infra-red energy for radiation heat loss which is absorbed near the water surface (usually between 0.2 to 0.6)
Δt	time step of the numerical solution
Δt_{summer}	time-shift due to the summer time
δ_{dec}^{ang}	declination angle in radian
δ_{ij}	Kronecker delta
ϵ	turbulence dissipation rate
η	Water level
ϕ	fluid property
ϕ_b	value of ϕ at the bed
ϕ_l	latitude of a position on the Earth
Γ_{day}	day angle measured in radian
γ_{GAS}	second control coefficient for the influence of the variable ϕ on the vertical grid distribution
κ	von Karman's constant =0.41
κ_e	wave number due to the eddy scale
λ_l	light extinction coefficient (usually between 0.5 and 1.4 m^{-1})
ν	kinematic viscosity of water

ν_t	turbulence eddy viscosity coefficient
ν_{eff}	effective eddy viscosity coefficient
ν_v	vertical eddy viscosity coefficient
ν_h	horizontal eddy viscosity coefficient
$\bar{\nu}_h$	depth averaged eddy viscosity in the horizontal plane
θ^{sh}	shields parameter for the bed shear stress
θ_c^{sh}	critical shields parameter for the bed shear stress
ρ	local density of the water
ρ_{air}	air density
ρ_η	density of the water at the free surface
ρ_o	average water density
τ_b	bed shear stress
$\tau_{c,er}$	critical shear stress for erosion
$\tau_{cr,d}$	critical bed shear stress for sediment deposition
τ_s	free surface shear stress
τ_x^s	x-component of the free surface shear stress
τ_{xy}	viscous stress component in the xy-plane
τ_{xz}	viscous stress component in the xz- plane
τ_y^s	y-component of the free surface shear stress
τ_{yz}	viscous stress component in the yz- plane
Ω_0	angular velocity of the Earth
$\omega_{ang}^{sunRise}$	sunrise angle
$\vec{\omega}_E$	rotation vector of the Earth
ω_{hr}	hour angle
ω_x	projections of $\vec{\omega}_E$ in the x-direction
ω_y	projections of $\vec{\omega}_E$ in the y-direction
ω_z	projections of $\vec{\omega}_E$ in the z-direction

1 Introduction

1.1 Overview

An examination of the trends in human health during recent decades reveals a great deal of good news. In many world regions today, more people are living longer and healthier. Undoubtedly, the improvement of water environment has played a significant role in this regard. The current role of the hydro-environmental engineering sector in the least developed countries is much more than just improvement. For many people living in underdeveloped areas, it is really the matter of to be or not to be. Scientists have made a good progress in this branch of engineering. Nevertheless, there is still a lot of work to do.

The main subject of hydro-environmental engineering today is not just a little more optimization, but it has a clear relation with human health, environmental conservation, peace, and sustainable development. In the least developed countries, one in five children do not live to see their fifth birthday, mostly because of avoidable environmental threats to health.

Many developing countries are now faced with modern toxic pollution problems. Many coastal areas around the world receiving urban and industrial discharges are suffering heavily from eutrophication caused algae decay. Eventually this leads to disturbance of the ecological balance and deterioration of the marine ecosystems. Discharge of cooling water from power plants has effects on the marine ecosystem in coastal areas. Construction of harbours, bridges, causeways and dams have an impact on the coastal and marine ecology not only during the construction phase but also after completion. Environmental Impact Assessment has become a requirement as part of the approval of any major construction work.

Nowadays, numerical modelling is an important technique in hydro-environmental engineering, particularly for problems concerning free surface flows. The evolution of computational fluid dynamic (CFD) models is progressing rapidly. The currently available tools are very valuable, but they are still far from being entirely satisfactory. Currently, two major fields still require significant research and improvement: a) CFD methods/models for specific problems, and b) techniques for coupling models or methods to support integrated environmental modelling. This present study focuses on developing an efficient system of depth integrated two dimensional (2DH) and fully three dimensional (3D) hydrodynamic model based on an unstructured grid and applying them in lakes and reservoirs.

1.2 Objectives of the project

There is a growing realisation that many problems of water management are best solved through an integrated approach. To date, integrated water quality modelling is an important issue and many researchers are focused on different aspects of this phenomenon. Integrating environmental models is not straightforward, the models must be able to link and work together. In most cases, integration means linking models for different physical processes, e.g. linking a free surface model to a groundwater model, or integrating a wave model with a current model. Some times integration means linking models for the same physical processes with different simplifications. Two common examples are: a) linking a 1D river model to a 2D shallow water one for flood management and, b) linking a waste-water model to a 2D free surface flow one.

The aim of the present study is to develop an integrated horizontal 2D-3D hydro-environmental model system for simulating the flow, sediment and water quality on unstructured grid. This kind of integration is applicable in many cases where 3D modelling is needed. The application domain of this model will be primarily lakes, reservoirs and esturine. Therefore, it is desirable that the new model is able to simulate the baroclinic forces (such as density current) and non-hydrostatic pressure distribution within the 3D domain. The computational grid selected unstructured in plan, in order to reflect the complexity in natural water courses .

1D and 2D free surface models are widely used in solving hydro-environmental engineering problems. They are sufficient for many practical cases. However, there are other cases were 3D modelling is necessary. Developing 3D models and using them in practical problems have a growing trend. 3D models are considerably more time consuming than the 2D and 1D models. They also need more detailed input data. In some practical problems, the simulation area is much larger than the area of interest. Usually the reason is the restriction in locating boundaries. Similarly, in most situations, it is not necessary to solve the whole simulation domain with three dimensional details. Such details are often needed just in a limited fraction of the model domain. However, 3D models are used to simulate the whole domain. This approach results in a considerable, but unnecessary, increase in the simulation run-time, computer resource usage and data management. It also results in a higher risk of instability, because a 3D model is more sensitive to the time step size and geometrical complexity than a depth integrated 2D one and it may crash in an area where it's three dimensional hydrodynamic details are actually not needed.

Figure (1.1) illustrates a typical example for which, integrated 2D-3D modelling is useful. Boushehr harbour, located in Boushehr bay at the Northern part of Persian Gulf, is one of the oldest ports in that region and the financial position of many families is related to it. The harbour has serious problems due to cohesive sedimentation in part of its access channel. Field measurements have shown that the velocity among the access channel has sometimes a three-dimensional behaviour with the upper layer's flow crossing the channel and the bottom layer following the channel direction. The velocity field in front of the river mouth, the main source of cohesive sediment material, is also shows a two-layer system. Therefore 3D modelling is needed. The flow field within the bay is mainly affected by the tidal regime in Persian Gulf. Hence the flow field must be simulated also in a regional scale. It is not necessary nor efficient to do 3D simulation in the regional domain. Therefore the need for integrated 2D-3D modelling comes up.

Reservoir simulation is another case where 2D-3D modelling may be useful. The 3D behaviour of the flow regime near the water intakes is important in some studies. However it is not always necessary to simulate the whole reservoir by 3D details. Therefore 3D modelling may be limited to the near-dam region for example to simulate erosion near the sediment bottom outlet, while the rest of the reservoir is simulated by a simpler model.

Sometimes boundary conditions required by a 3D sub-domain model can be obtained from an independent 2D model with a large domain. However, there are cases where the 3D sub-domain interacts with the larger domain. Some of the flow characteristics, such as the large eddies that cross the local models' interface, can not be simulated in this way. These kind of eddies are usually developed around the headlands, islands and bays and when the flow direction is changed by tide. As an example, figure (1.2) illustrates the transient eddies around a bay (Chabahar) located at the northern coast of Oman Sea. They cross the local model's boundary if it is not far from the bay-opening. Another problem with data transfer from an independent regional 2D model to the 3D boundary of a local model is that it requires extensive effort for data management when the flow regime is unsteady. An efficient and general solution is to integrate the 3D sub-domain model with a 2D horizontal (H) one to gain expedition and accuracy together. This is the aim of this study. This research aims to increase the efficiency of 3D modelling by coupling it dynamically with horizontal 2D modelling without additional restrictions (such as time step limitation) due to integration.

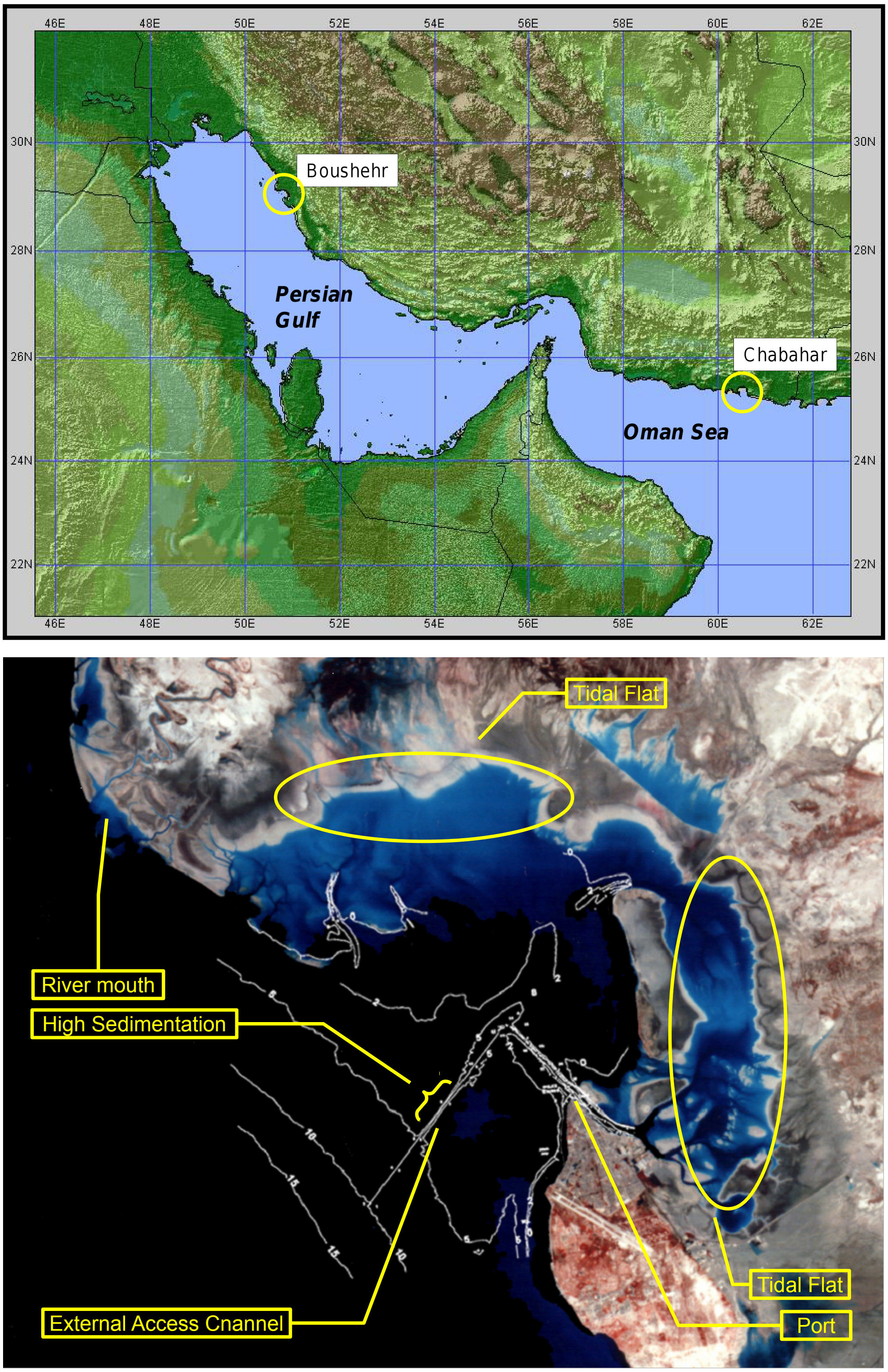


Figure 1.1. Boushehr harbour at the North part of Persian Gulf
with sedimentation problem in the access channel
and 3D velocity behaviour along the access channel and the river mouth

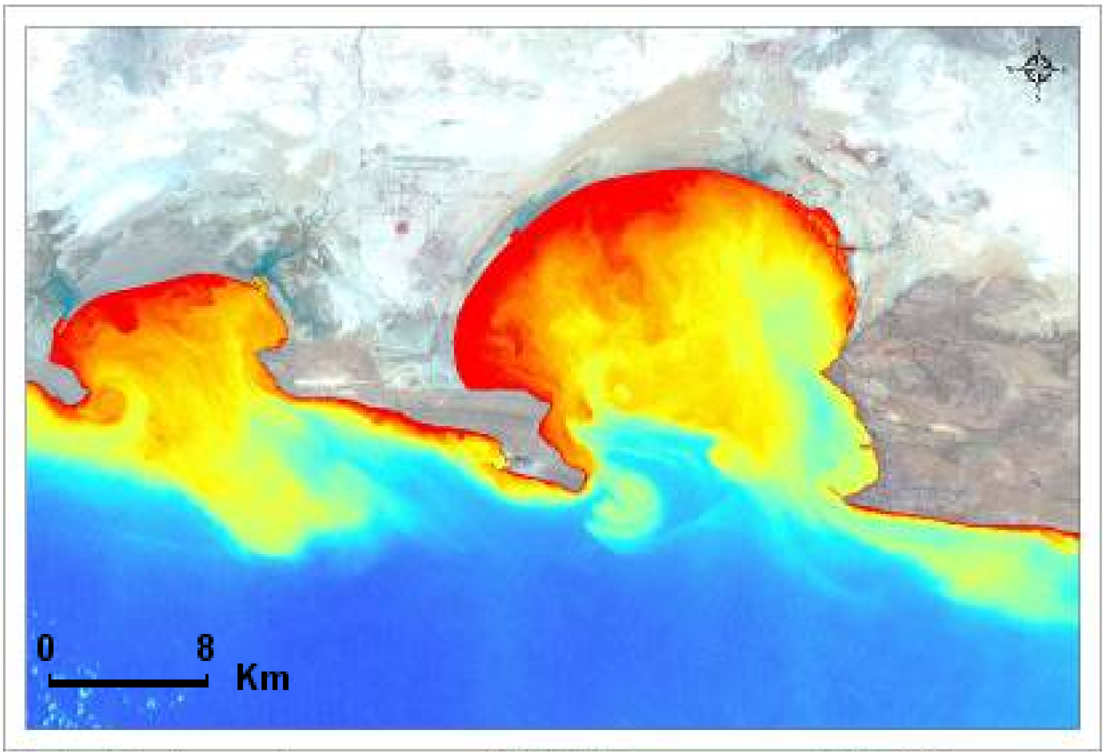


Figure 1.2. Transient eddies clarified by Landsat image of Chabahar and Pozm bays North of Oman Sea, July 17, 2000, with emphasis on colour band 2 (Baird, 2008)

There are just a few research works on integrating 2DH and fully 3D free surface flow and quality models together. Two approaches have been employed in the literature for combining multi-dimensional free surface models. One approach is to use a simpler model in the entire solution domain and a more complex model in confined sub-domains. The simpler model provides boundary conditions for the more complex model, but they are not coupled. This approach is called the hybrid modelling (McAnally et al., 1986; Yu, L. et al, 1998;). The other approach is is to fully couple the models (Wu W., 2007; Sebastian and Shu, 2003; Verway, 2001; McAnally et al., 1986; Wu and Li, 1992; Zhang, 1999; Kashefipour et al, 2002; Namin, 2003; Katopodes and Kao, 2003). The current fully coupled 2D-3D model systems are based on the interpolation/averaging procedure in order to exchange data between the models. Therefore they are not fully conservative at the interface. They use mesh overlapping or nested grid techniques at the interface. By mesh overlapping, which also called the multi-block approach, each model received its boundary condition at the interface from inside the other model's domain. It is not a fully implicit coupling algorithm. The model of Namin (2003) is not based on mesh-overlapping and is fully implicit and dynamic. However his technique can not be extended to the unstructured grid without difficulties and loss of efficiency. It depends completely on the ADI algorithm and the structured grid characteristics. It is also based on the nested grid approach which needs interpolation-averaging and is not fully conservative at the nesting interface. For details see section (7.13).

Some recent 2D and/or 3D models (or model versions) fulfil a general interaction standard named Open MI (Open Modelling Interface and Environment, www.OpenMI.org). There is also a growth in the number of models that fulfil this standard. From the theoretical point of view, it is possible to link two 2DH and 3D models of this type in order to simulate a 2DH-3D domain semi-interactively. However, this approach results in definitive restrictions practically when it is used for the hydrodynamic flow modelling. One reason is that the data interaction at the link boundary will be explicit, which restricts the time step for simulating some of the physical processes such as the free surface. Open MI is a useful toolbox for linking the models via the hybrid mechanism or integrating them via the multi-block approach. However, it can not be used a tool for fully implicit and dynamic model integration.

None of the present coupling techniques present a fully dynamic, implicit and conservative integration procedure on unstructured grid without enforcing solution iterations. The proposed integration strategy in the present work is a novel algorithm which is free from interpolation/averaging so it is fully conservative and accurate at the interface, it is based on unstructured grid, and does not restricts the implicit behaviour of the two models in simulation time step selection.

It is important for an integrated system to design a smooth and non-restricting dynamic interaction algorithm between the models. For example the overall computational time step must not be restricted by an explicit linking algorithm, when each one of the models are implicit separately and can use large time steps when working alone. The sensitivity of different physical processes is not the same in this regard. For example the diffusion process will not restrict the overall computational time step even if it is handled explicitly. But the free surface simulation will restricts the time step considerably when it is handled explicitly at the linking boundary of the two model domains. Therefore, the most efficient integration may be achieved when each one of the models are based on solution techniques that are most efficient for the integration procedure.

The integration algorithm in this research work is based on the following ideas:

a) The models must solve the equations process by process (fractional steps algorithm) and exchange data with each other at the process levels. In other words each common process (fraction step) must be integrated separately by its most efficient method.

b) Hybrid finite volume cells have been introduced for the linking boundary nodes, where the 2D and 3D domains interact (finite volume is the method of choice in this work except for the advection step). A part of the hybrid cells is located in the 2D domain and the remaining part is located at the 3D domain. In this way, each linking node is solved just once per fraction and receives data from both sides.

The result of the present research work includes:

1- A new unstructured-grid 3D model including non-hydrostatic pressure and baroclinic forces among other capabilities, ready for integration.

2- A new unstructured-grid depth integrated 2D model, ready for integration.

3- An integration algorithm that couples the above models dynamically and interactively.

The 2D and 3D models in this research have been developed from scratch, in order to fulfil the proposed framework completely. The computational language C++ has been used because of its flexibility in object-oriented programming, flexibility in mixing the object-oriented program with procedural code segments of Fortran and C in order to gain more efficiency, and its power in memory management.

The ability of simulating baroclinic forces is necessary for the 3D model because it must be able to be used in the dam reservoirs, where sometimes density currents such as turbidity current might happen. The ability of non-hydrostatic pressure simulation is necessary for the 3D model because the vertical acceleration is not negligible at some areas of the dammed reservoir, for example near the outlets. The turbidity current also may not be simulated always accurately without solving the non-hydrostatic pressure component. The pressure treatment in the 3D model is based on the projection method, which is consistent with the fractional steps algorithm. The unstructured grid is crucial for both of the 2D and the 3D models, because their application domain has sometimes very irregular boundaries with complex domain shapes and rapid changes (for example many of the mountainous dam reservoirs). It is difficult to reflect such boundaries with efficient structured grid (even curvilinear or nested grid). Using unstructured grid guarantees the flexibility that is needed in such problems. Sigma coordinate system has been used in the vertical direction. However the horizontal gradients of the water density have been computed on real horizontal planes.

The time step fractions of the present 3D flow model are advection step, horizontal diffusion step with baroclinic force, free surface step with barotropic force, vertical diffusion step, and hydrodynamic pressure step with velocity projection, respectively. The fractions of the 3D advection-diffusion solver are advection, horizontal diffusion, and vertical diffusion. The time step fractions of the 2D flow model are advection step, horizontal diffusion step, and free surface step. The fractions of the 2D advection-diffusion solver are advection and horizontal diffusion steps. The common fractions of the 2D and 3D models are handled together with similar procedures and data exchange directly. The advection step is solved by the semi-Lagrangian method. Horizontal diffusion is solved explicitly via finite volume (FV). Hybrid cells handle the nodes on the link boundary. Free surface steps of the two models are solved together implicitly. The depth integrated forms of the continuity and momentum equations are solved by FV while the hybrid cells link the domains. The vertical diffusion is solved implicitly column by column via 1D FV cells.

The FV cell integrals are calculated analytically using finite element triangles. Unstructured finite difference method and the divergence theorem are used for nodal and cell-integral values of the spatial gradients respectively.

Figure (1.3) illustrates the flow chart of the new 2DH and 3D models in company with the new integration algorithm.

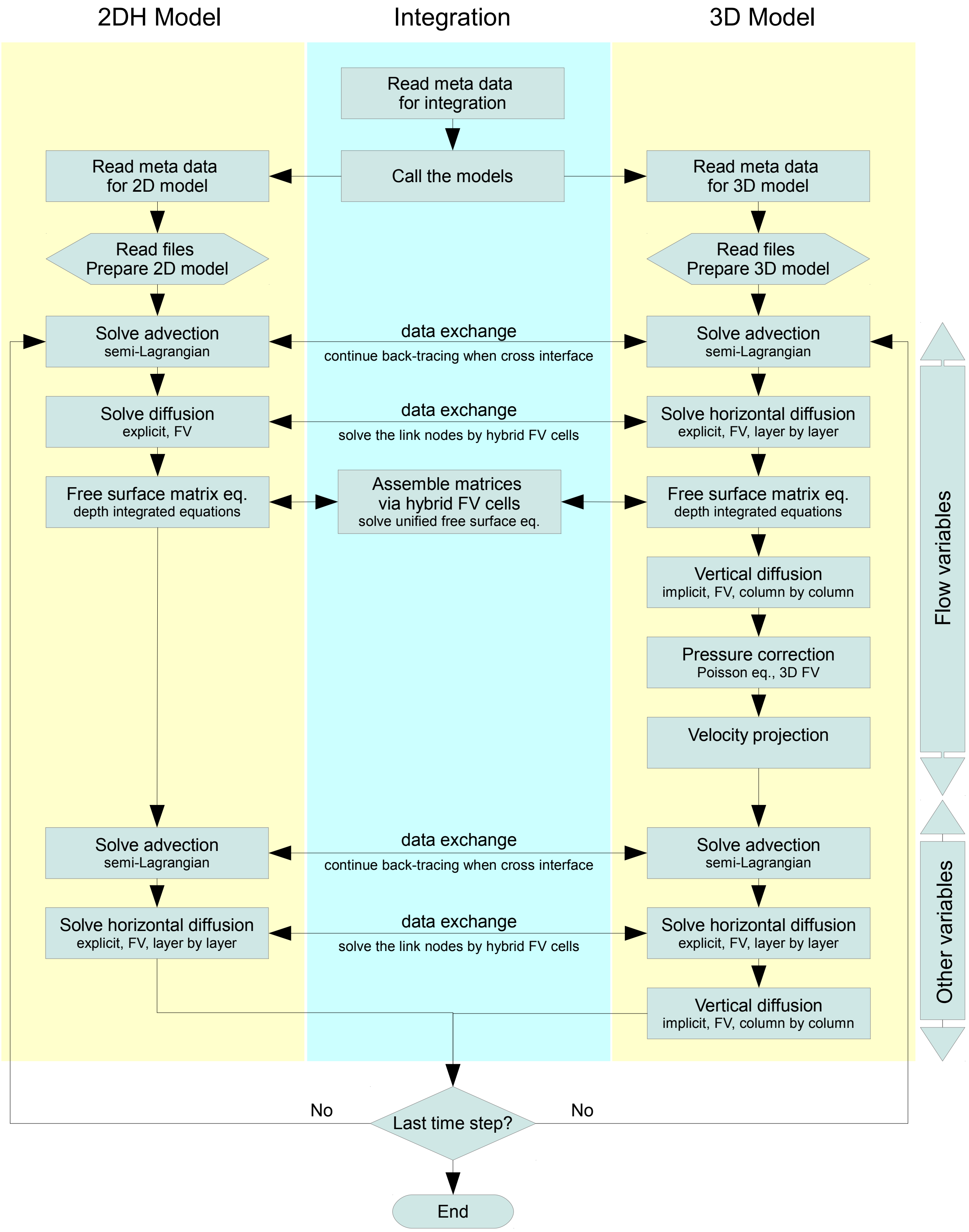

2DH Model
Integration
3D Model
Read meta data for integration
Call the models
Read meta data for 2D model
Read meta data for 3D model
Read files Prepare 2D model
Read files Prepare 3D model
Solve advection semi-Lagrangian
data exchange
continue back-tracing when cross interface
Solve advection semi-Lagrangian
Solve diffusion explicit, FV
data exchange
solve the link nodes by hybrid FV cells
Solve horizontal diffusion explicit, FV, layer by layer
Free surface matrix eq. depth integrated equations
Assemble matrices via hybrid FV cells solve unified free surface eq.
Free surface matrix eq. depth integrated equations
Vertical diffusion implicit, FV, column by column
Pressure correction Poisson eq., 3D FV
Velocity projection
Flow variables
Solve advection semi-Lagrangian
data exchange
continue back-tracing when cross interface
Solve advection semi-Lagrangian
Solve horizontal diffusion explicit, FV, layer by layer
data exchange
solve the link nodes by hybrid FV cells
Solve horizontal diffusion explicit, FV, layer by layer
Vertical diffusion implicit, FV, column by column
Other variables
No
Last time step?
No
End

Figure 1.3. Flow chart of the new models and the new integration algorithm

1.3 Outline of the Thesis

This document is divided into nine chapters. After introduction, there is a presentation about the physical and mathematical behaviour of free surface flows as chapter two. It includes descriptions about turbulence, turbidity current (a special case of density current), shallow water modelling, and connection conditions for model integration. It also describes the special situation with most hydro-environmental models, where the horizontal scale is much larger than the vertical one.

Chapter three reviews the solution strategies for three dimensional free surface flow modelling and the 2DH-3D coupling strategies. It deals with the pressure-velocity relationship for incompressible flows, defining the unknown and moving free surface domain boundary, and common procedures for the pressure simulation. The selected algorithms for this study have been marked at the end of the chapter.

Chapter four reviews the general discretisation algorithms needed for numerical simulation of flow and water quality. Chapter five is focused on the governing equations that must be solved in a numerical model for hydrodynamic flow and water quality. It also deals with the sigma-coordinate transformation procedure and the transformed equations.

Chapter six presents the discretisation algorithms that are used in the present work. It includes the horizontal interpolation finite volume element, the vertical interpolation elements, and the 3D interpolation procedure with is needed for the advection solver. It also describes the horizontal and vertical and 3D finite volume cells which are of hybrid finite element – finite volume type. The analytical procedure of integration over the cells and their boundaries has also described in this chapter. This chapter also describes the unstructured finite difference technique for point-wise spatial gradient calculations and also application of the divergence theorem for computing the cell-integration of the gradients directly.

Chapter seven presents the structure of the 3D and 2D models of this study. It contains the numerical procedures and the overall structures of the 3D model, the 2D model and their integration. The numerical procedure for each fraction is also described (the models are based on the fractional steps algorithm). Chapter eight concerns the model evaluation. It presents the model results for some typical benchmarks and compares the results with analytical solutions whenever possible. Chapter nine is for the conclusions and recommendations.

2 Free Surface Hydrodynamics

2.1 Introduction

Numerical modelling of hydrodynamic phenomena has become a standard approach in dealing with engineering problems and research. In this regard, the clarity of the related mathematical equations is necessary. Many natural processes can be described by conservation laws. Conservation equations of mass and momentum are the key for simulating free surface flows. They are described in section (2.2). The mass and momentum equations can be averaged to remove oscillations and the need for non-realistic dense temporal and spatial solution grids. The price of this benefit is the addition of new unknowns in the momentum equations. It leads to the topic of turbulence modelling, which is the subject of section (2.3). Section (2.4) describes the effect of density gradient as a driving force and its effect on the structure of turbulence.

Generally, the horizontal scale is some order of magnitudes larger than the vertical scale, the water depth, in hydro-environmental modelling problems. Therefore the vertical grid spacing is much smaller than the horizontal correspondent in numerical simulation. Thus some special techniques are necessary in modelling such problems, which are the subject of section (2.5). Section (2.6) concerns the depth integration and two dimensional shallow water equations. Because this study concerns the integration of two and three dimensional hydrodynamic modelling. Section (2.7) involves the necessary conditions needed for model integration at the connection interface.

2.2 Flow Modelling

No matter how complicated the detailed evolution of a system might be, during the evolution of a fluid, a certain number of properties, such as mass, generalized momentum and energy, are conserved during the whole process and at all times. In addition, they can completely determine the behaviour of the system. Therefore the behaviour of a physical system is completely determined by conservation laws. The conservation laws also provide our most common basic controls upon the reasonableness of the behaviour of a computation. The concept of conservation means that the variation of a conserved flow quantity within a given volume is due to the net effect of some internal sources and the amount of the quantity which is crossing the boundary surface. (Hirsch, 1988; Abbott and Basco, 1989).

Conservation laws can be derived by considering a given quantity (a parcel) of matter or control mass and its extensive properties such as mass, momentum and energy. This approach is used to study the dynamics of solid bodies. In fluid flow, however, it is difficult to follow a parcel of matter. It is more convenient to deal with the flow within a certain spatial region (a control volume), rather than in a parcel of matter which quickly passes through the region of interest. This method of analysis is called the control volume approach (Ferziger and Peric, 2002).

The key governing equations of fluid flow, which represent mathematical statements of the conservation laws of physics, are:

- The mass of a fluid is conserved,
- The rate of change of momentum equals the sum of the forces on a fluid particle (Newton's second law),
- The rate of change of energy is equal to the sum of the rate of heat addition to and the rate of work done on a fluid particle (first law of thermodynamics).

2.2.1 Mass Conservation

Mass can neither be created nor destroyed. The increase in mass within a control volume is equal to the mass inflow minus the mass outflow through the control surface plus the effect of sources and sinks (such as intakes, outfalls and exchange for some two-phase flows). In absence of source/sinks, the mass conservation law for a control volume in mathematical notation is (Abbott and Basco, 1989):

$$\int_{V_{cell}} \frac{\partial \rho}{\partial t} dV + \int_{S_{cell}} (\rho) \vec{u} \cdot \vec{n} \, dA_{cell} = 0 \qquad (2.1)$$

where ρ is the mass density, $\vec{u}$ is the fluid velocity, $\vec{n}$ is a unit normal vector (positive outward), V_{cell} is the control volume, and A_{cell} is the control volume surface area.

For a very small element of fluid like the one shown in figure (2.1), the mass conservation or continuity equation can be written as follows (Versteeg and Malalasekera, 1995):

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \qquad (2.2)$$

or in more compact vector notation:

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \vec{u}) = 0 \qquad (2.3)$$

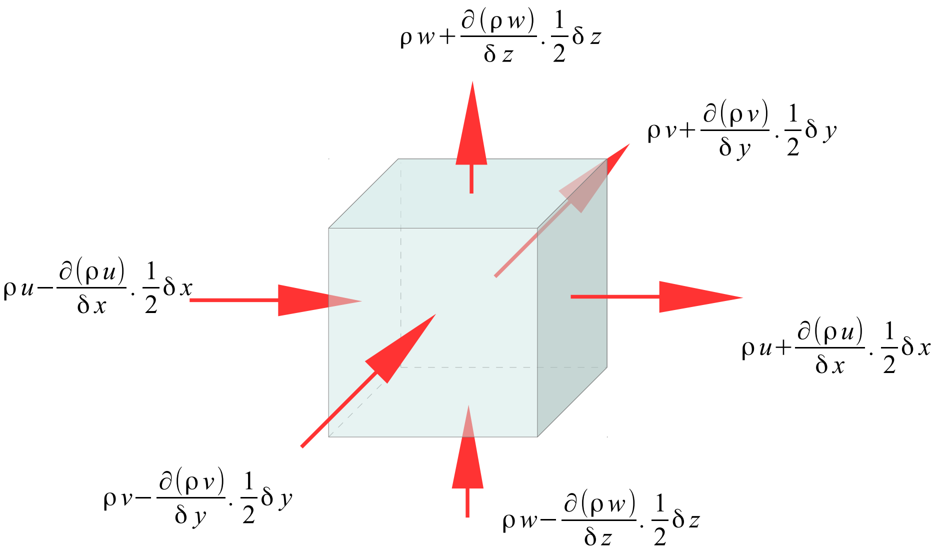

$\rho w+\frac{\partial(\rho w)}{\delta z}.\frac{1}{2}\delta z$
$\rho v+\frac{\partial(\rho v)}{\delta y}.\frac{1}{2}\delta y$
$\rho u-\frac{\partial(\rho u)}{\delta x}.\frac{1}{2}\delta x$
$\rho u+\frac{\partial(\rho u)}{\delta x}.\frac{1}{2}\delta x$
$\rho v-\frac{\partial(\rho v)}{\delta y}.\frac{1}{2}\delta y$
$\rho w-\frac{\partial(\rho w)}{\delta z}.\frac{1}{2}\delta z$

Figure 2.1. A differential element of fluid and mass balance

The first term on the left hand side of equation (2.2) is the density rate of change in time (mass per unit volume). The second term describes the net flow of mass out of the element across its boundaries and is called the convective term.

For an incompressible fluid (i.e. a liquid) the mass conservation equation simplifies to:

$$\operatorname{div}\vec{u}=0 \tag{2.4}$$

or in longhand notation:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{2.5}$$

The above equation is not a mathematically exact relationship. It is just a very good approximation of the original mass conservation equation. It is acceptable in situations where the density is not affected by change in pressure or net mass inflow/outflow (which can affect the pressure).

Incompressibility means that the density of an element of fluid is not affected by pressure change. Still, change in some other parameters like temperature, salinity and suspended sediment concentration can affect the fluid density. Hence, incompressibility does not imply that density is constant. In this manner, incompressibility is a property of flow, not a property of the fluid itself. It is possible to have incompressible flow of a compressible matter; the pattern of air flow can be similar to that of water in some conditions.

There are specific conditions in which, the flow can be assumed incompressible. The most important one is related to the Mach number (the flow speed / the speed of sound in the field). Whenever $Ma^2 << 1$ everywhere (say $Ma^2 < 0.1$, that is $Ma < \sim 0.3$), incompressibility is a good approximation.

2.2.2 Momentum Conservation

Newton's second law states that the rate of change of momentum of a fluid particle equals the sum of the forces on that particle. The most useful forms of the conservation laws for CFD are concerned with flow property changes for a stationary element in space (control volume), not with fluid particles. The relationship between any property of a moving fluid particle and a fixed fluid element is as follows (Versteeg and Malalasekera, 1995):

$$\begin{array}{ccccc} \text{Rate of increase of} & & \text{Net rate of flow of} & & \text{Rate of increase of} \\ \phi & + & \phi & = & \phi \\ \text{of fluid element} & & \text{out of fluid element} & & \text{for a fluid particle} \end{array}$$

Hence the Newton's second law can be expressed as:

The vector sum of all external forces acting on a control volume is equal to the sum of; a) the total rate of change of momentum of mass within the control volume, and b) the rate of flux of momentum through the control surface.

In mathematical notation, it is (Abbott and Basco, 1989):

$$\int_{V_{cell}} \frac{\partial(\rho\vec{u})}{\partial t} dV_{cell} + \int_{S_{cell}} (\rho\vec{u})(\vec{u}.\vec{n}) dA = \vec{F} \tag{2.6}$$

where $(\rho\vec{u})$ is the momentum “flux density” and $\vec{F}$ is the vector sum of all the external forces.

External forces can be categorized into surface and body forces. Surface forces contain pressure forces and viscous forces. Body forces contain gravity force, centrifugal force, Coriolis effect and electromagnetic force.

Surface forces have an important role in deriving the usual form of momentum conservation equations (Navier-Stocks or N.S. equations). The action of surface forces (pressure and viscous stresses) on a differential control volume are illustrated in figures (2.2) and (2.3).

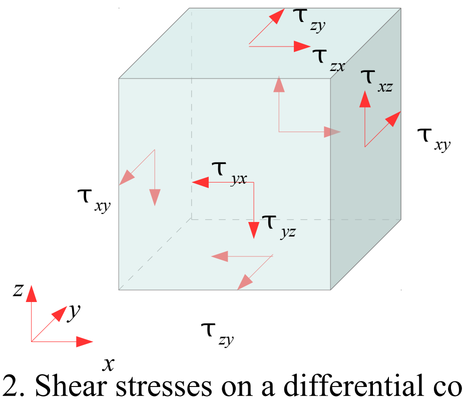


Figure 2.2. Shear stresses on a differential control volume

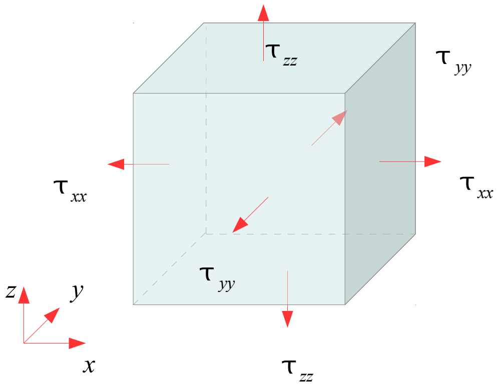


Figure 2.3. Normal forces (pressure) on a differential control volume

The most common body force is due to gravity. Its value is $-\rho g$ in the vertical direction (if the upward direction is positive).

The effect of external forces on a differential control volume leads to momentum change. It is equal to the momentum change for the mass within the control volume plus flux of momentum through the control surface[1]. It means:

$$\rho\frac{Du}{Dt}\equiv\frac{\partial(\rho u)}{\partial t}+\mathrm{div}(\rho u\vec{u}) \qquad (2.7)$$

$$\rho\frac{Dv}{Dt}\equiv\frac{\partial(\rho v)}{\partial t}+\mathrm{div}(\rho v\vec{u}) \qquad (2.8)$$

$$\rho\frac{Dw}{Dt}\equiv\frac{\partial(\rho w)}{\partial t}+\mathrm{div}(\rho w\vec{u}) \qquad (2.9)$$

for x,y and z components of the velocity field respectively.

Therefore, for the three components of momentum equation, we have (Versteeg and

[1] As described for the relationship between a fluid particle and a stationary fluid element.

Malalasekera, 1995):

$$\rho\frac{Du}{Dt}\equiv\frac{\partial(\rho u)}{\partial t}+\mathrm{div}(\rho u\vec{u})=-\frac{\partial p}{\partial x}+\left(\frac{\partial\tau_{xx}}{\partial x}+\frac{\partial\tau_{yx}}{\partial y}+\frac{\partial\tau_{zx}}{\partial z}\right)+BodyForces_x \tag{2.10}$$

$$\rho\frac{Dv}{Dt}\equiv\frac{\partial(\rho v)}{\partial t}+\mathrm{div}(\rho v\vec{u})=-\frac{\partial p}{\partial y}+\left(\frac{\partial\tau_{xy}}{\partial x}+\frac{\partial\tau_{yy}}{\partial y}+\frac{\partial\tau_{zy}}{\partial z}\right)+BodyForces_y \tag{2.11}$$

$$\rho\frac{Dw}{Dt}\equiv\frac{\partial(\rho w)}{\partial t}+\mathrm{div}(\rho w\vec{u})=-\frac{\partial p}{\partial z}+\left(\frac{\partial\tau_{xz}}{\partial x}+\frac{\partial\tau_{yz}}{\partial y}+\frac{\partial\tau_{zz}}{\partial z}\right)+BodyForces_z \tag{2.12}$$

The governing equations also contain, as further unknowns, the viscous stress components (τ_{ij}). In many fluid flows the viscous stresses can be expressed as functions of the local deformation rate. For a Newtonian fluid (like water), the viscous stresses are proportional to the rate of deformations. Assuming that the fluid is isotropic, we have:

$$\begin{aligned}\rho\frac{Du}{Dt}&\equiv\frac{\partial(\rho u)}{\partial t}+\mathrm{div}(\rho u\vec{u})=-\frac{\partial p}{\partial x}\\&+\frac{\partial}{\partial x}\overbrace{\left(2\mu\frac{\partial u}{\partial x}+\lambda\,\mathrm{div}\,\vec{u}\right)}^{\tau_{xx}}+\frac{\partial}{\partial y}\overbrace{\left(\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right)}^{\tau_{yx}}+\frac{\partial}{\partial z}\overbrace{\left(\mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)\right)}^{\tau_{zx}}+BodyForces_x\end{aligned} \tag{2.13}$$

and similar relationships for y- and z- momentum components.

Liquids are incompressible, so that the mass conservation equation is $\mathrm{div}\,\vec{u}=0$, so the term $\lambda\,\mathrm{div}\,u$ is eliminated from equation (2.13) and its y and z correspondents. After some rearrangements, the famous Navier-Stocks equations are derived:

$$\rho\frac{Du}{Dt}\equiv\frac{\partial(\rho u)}{\partial t}+\mathrm{div}(\rho u\vec{u})=-\frac{\partial p}{\partial x}+\mathrm{div}\left(\mu\,\overrightarrow{\mathrm{grad}\,u}\right)+BodyForces_x \tag{2.14}$$

$$\rho\frac{Dv}{Dt}\equiv\frac{\partial(\rho v)}{\partial t}+\mathrm{div}(\rho v\vec{u})=-\frac{\partial p}{\partial y}+\mathrm{div}\left(\mu\,\overrightarrow{\mathrm{grad}\,v}\right)+BodyForces_y \tag{2.15}$$

$$\rho\frac{Dw}{Dt}\equiv\frac{\partial(\rho w)}{\partial t}+\mathrm{div}(\rho w\vec{u})=-\frac{\partial p}{\partial z}+\mathrm{div}\left(\mu\,\overrightarrow{\mathrm{grad}\,w}\right)+BodyForces_z \tag{2.16}$$

or in extended notation:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)+\frac{1}{\rho}BodyForces_x \tag{2.17}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\frac{\mu}{\rho}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right)+\frac{1}{\rho}BodyForces_y \tag{2.18}$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial z}+\frac{\mu}{\rho}\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right)+\frac{1}{\rho}BodyForces_z \tag{2.19}$$

There are many other equivalent ways to represent the viscous terms, advection terms, and non-linear terms in Navier-Stokes equations using vector algebra.

2.2.3 Body Forces

For hydro-environmental problems, body forces in the momentum equations involve the gravity effect and Coriolis effect. When the z-direction is exactly vertical, the gravity affects just the z-momentum equation, but does not appear in the x- and y- momentum equations.

Coriolis effect is not a real force, but acts exactly as a real force in the governing equations. It is the result of earth rotation. It is possible to derive momentum equations in a really fixed astronomical coordinate system without Coriolis terms. However, using a local coordinate system which is fixed to the earth and is rotational is much easier. The cost is just adding Coriolis terms to the equations.

Considering a location with latitude ϕ_l on the sphere of the Earth. A local coordinate system may be set up with the x-axis in the east direction, the y-axis in the north direction and the z-axis vertically upwards. Considering that the Earth is rotating around the north-south axis with the angular velocity of $\vec{\omega}_E$, the rotation vector, the velocity of movement and Coriolis acceleration expressed in this local coordinate system are:

$$\vec{\omega}_E = |\omega| \cdot \begin{bmatrix} 0 \\ \cos\phi_l \\ \sin\phi_l \end{bmatrix}, \quad \vec{V} = \begin{bmatrix} v_{east} \\ v_{north} \\ v_{upp} \end{bmatrix}, \quad \vec{a}_c = -2 \cdot \vec{\omega}_E \times \vec{V} = 2 \cdot |\omega| \cdot \begin{bmatrix} v_{north} \cdot \sin\phi_l - v_{upp} \cdot \cos\phi_l \\ v_{east} \cdot \sin\phi_l \\ v_{east} \cdot \cos\phi_l \end{bmatrix} \qquad (2.20)$$

For shallow water conditions the vertical velocity is small and the vertical component of the Coriolis acceleration is small compared to the gravity. In such a condition, the body forces in the right hand side of the momentum equations are ($+f\,v$) for the x-momentum equation, ($-f\,u$) for the y-momentum equation, and ($g \simeq 9.81$) for the z-momentum equation respectively, with:

$$f = 2\Omega_0 \sin(\phi_l) \qquad (2.21)$$

where Ω_0 is the rotational speed of the Earth (7.2921159 × 10^-5 rad/s) and ϕ_l is latitude.

It is important to note that if the x-direction of the coordinate system is not in the eastward direction, then the Coriolis forces in the hydrodynamic equations are different from what mentioned above. The general form of Coriolis forces, for a generally directed coordinate system, may be written as:

$$\vec{a}_c = -2 \cdot \vec{\omega}_E \times \vec{V} = -2 \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} = 2 \cdot \begin{bmatrix} v \cdot \omega_z - w \cdot \omega_y \\ w \cdot \omega_x - u \cdot \omega_z \\ u \cdot \omega_y - v \cdot \omega_x \end{bmatrix} \qquad (2.22)$$

For applications to meso-scale oceanic flows the vertical component of the Coriolis effect should also be incorporated (Casulli and Stelling, 1998).

2.2.4 Boundary Conditions

The boundary and initial conditions distinguish the simulation results of different problems. The necessary boundary data for a 3D problem are:

- Pressure at the lateral open boundaries[1]
- Flux or velocity at the lateral open boundaries
- Barometric pressure field at the free surface[2] (or the horizontal gradients)
- Shear stress at free surface or wind velocity and shear coefficient
- Bed resistance

The kinematic boundary condition at the free surface is:

$$\frac{\partial \eta}{\partial t}+u_\eta \frac{\partial \eta}{\partial x}+v_\eta \frac{\partial \eta}{\partial y}=w_\eta \tag{2.23}$$

The kinematic boundary condition at the bottom is:

$$u_b \frac{\partial z_b}{\partial x}+v_b \frac{\partial z_b}{\partial y}=w_b \tag{2.24}$$

Free surface shear stress is a boundary condition for vertical shear term. It is caused by the horizontal velocity difference between water and air. The relevant formula for x-direction is as follows:

$$\frac{\tau_{xz}}{\rho_{water}}=\nu_T\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)=\frac{\rho_{air}}{\rho_{water}}\cdot C_w \cdot W \cdot W_x \tag{2.25}$$

where ρ_{air} is the air density, ρ_{water} is the water density, W is the wind speed relative to water velocity at free surface, and C_w is the wind drag coefficient.

The relationship in the y-direction is similar. An empirical formula for wind friction factor in SI unit system is as follows:

$$\begin{aligned} &C_w=C_0+\left(C_{24}-C_0\right).\frac{Wind\ Velocity}{24.0} && if && Wind\ Velocity\leqslant 24.0 \\ &C_w=C_{24} && if && Wind\ Velocity>24.0 \end{aligned} \tag{2.26}$$

where $C_0=0.0013$ and $C_{24}=0.0026$.

[1] Sometimes surface level may be adequate, when hydrostatic pressure assumption is sufficient at lateral boundaries.

[2] In most numerical models, pressure is needed at the topmost computational point, which is not exactly at the free surface, but half a cell below it. The pressure distribution between this point and free surface may assume hydrostatic.

Bottom shear stress is another boundary condition for vertical shear term. It is common to assume a logarithmic vertical profile for horizontal (or bed-parallel) velocity in the most down-side computational cell. In this way, the following relationship may be used for the bed shear stress:

$$\frac{\tau_{xz}}{\rho}=\nu_t\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)=\left[\frac{1}{\kappa}\ln\left(\frac{d_0}{k_s/30}\right)\right]^2.|u|.\,u_x \tag{2.27}$$

where $\kappa=0.41$ is von Karman's constant, k_s the bed roughness height, d_0 the distance from bed, $|u|$ the current speed at distance d_0 above the bed, and u_x the x component of velocity at the same distance from bed.

The relationship in the y-direction is similar. The drag coefficient formula must be consistent with the vertical turbulence closer model.

In most practical applications, it is acceptable and sufficient to assume the horizontal velocity distribution near the lateral boundaries to be linear.

For shallow-water models, boundary conditions at the bed mostly include just vertical diffusion terms using wall functions. Additionally, the horizontal diffusion terms are also needed when bed slope is not small and pressure is non-hydrostatic (see Stansby and Zhou, 1998 for more details).

2.3 Turbulence Modelling

Turbulence modelling is one of the key elements in Computational Fluid Dynamics. Turbulence is a natural phenomenon in fluids that occurs when velocity gradients are high, resulting in flow disturbance as a function of space and time. Most flows encountered in engineering practice, become unstable and turbulent above a certain Reynolds Number[1], even when the controlling factors of the flow, such as the model geometry and upstream conditions, are stationary. In fact, it is impossible to obtain steady-state results in many situations of viscous flow (Zienkiewicz and Taylor, 2000). For slightly viscous fluids, such as water and air, large (turbulent) Reynolds number corresponds to anything stronger than a small swirl or a puff of wind. From a mathematical point of view, the instabilities result from the interaction between the Navier-Stokes equation's non-linear inertial terms and viscous terms (Wilcox, 1994).

Turbulence can be described as a state of continuous instability, whereas it is still possible to separate the fluctuations from the mean flow properties (Jasak, 1996). The most common source of turbulence is shear flow instability (Abbott and Basco, 1989). It arises in contact with walls or in between two neighbouring layers of different velocities. With velocity gradients increasing,

[1] The Reynolds number of a flow gives a measure of the relative importance of inertia forces (associated with convective effects) and viscous forces.

the flow becomes rotational, leading to a vigorous stretching of vortex lines, which cannot be supported in two dimensions. Thus, turbulent flows are always physically three-dimensional (Chung, 2002). Figure (2.4) shows an example of 3D turbulent vortexes.

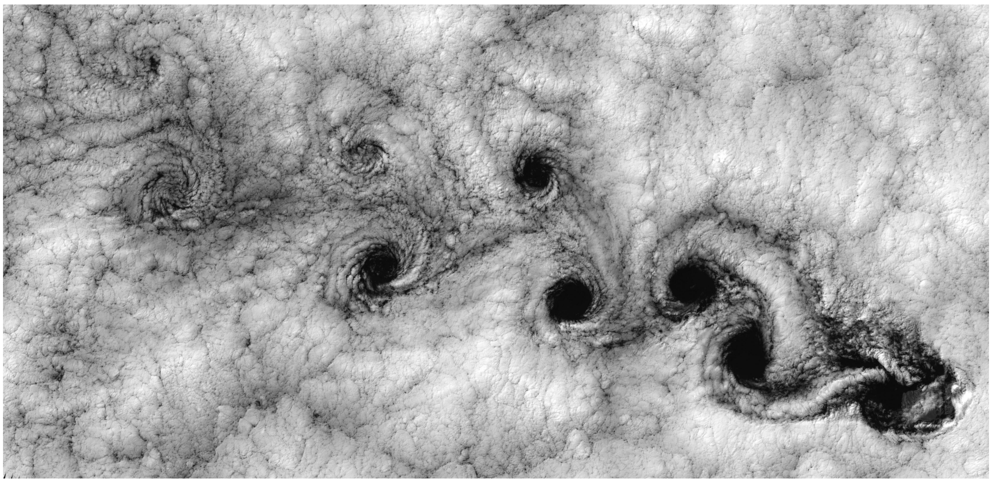


Figure 2.4. 3D behaviour of turbulence within cloud, (Landsat7 08Aug05)

Turbulence increases the rate of mixing (accomplished by diffusion) by orders of magnitude and reduces the gradients. The increased mixing of momentum leads to increased frictional forces and energy dissipation. The dissipated energy is converted into internal energy of the fluid, i.e. an increase in temperature.

Turbulent flows fluctuate on a broad range and continuous spectrum of length and time scales (Ferziger and Peric, 2002). They contain rotational flow structures (turbulent eddies) with a wide range of length scales (Versteeg and Malalasekera, 1995). Large eddies have a long length scale and time scale (period of generation or rotation), associated with a low frequency, while small eddies have a high frequency associated to small length and time scales (Abbott and Basco, 1989). The range of scales in turbulent flow is several orders of magnitude. The smallest turbulent eddies characterised by micro-scales, but still are far larger than any molecular length scale. As an example, for a fully-developed turbulent boundary layer flow, the length scale appropriate to the large energy-bearing eddies (often referred to as the integral scale in statistical turbulence theory) is one-tenth times the boundary-layer thickness. At the same time, the smallest scale (Kolmogorov length scale) outside the viscous near-wall region is less than one ten-thousandth times the thickness of the boundary layer (Wilcox, 1994; Jasak, 1996). The smallest scale is dictated by viscosity. At these scales (lengths on the order of 0.1 to 0.01 mm and frequencies around 10 kHz in typical turbulent engineering flows) viscous effects become important. Thus the energy associated with the eddy motions is dissipated and converted into thermal internal energy (Versteeg and Malalasekera, 1995).

The larger-scale turbulent motion carries most of the energy and is mainly responsible for the enhanced diffusivity and stresses. The characteristic velocity and length of the larger eddies are of the same order as the mean flow. The presence of mean velocity gradients in sheared flows distorts the rotational turbulent eddies, because one side is forced to move faster than the other. Larger eddies also carrying smaller ones. The stretching work done by the mean flow on the large eddies provides the energy which maintains the turbulence. Vortex stretching creates motions at smaller transverse eddies which have smaller time scales. In this way, the turbulence energy is handed down from large eddies to progressively smaller and smaller ones. This procedure is termed the turbulence energy cascade. Ultimately, the smallest eddies dissipate into heat through the action of molecular viscosity (Versteeg and Malalasekera, 1995; Ferziger and Peric, 2002; Wilcox, 1994; Abbott and Basco, 1989). Figure (2.5) compares the largest eddies' size with the whole flow domain at the north part of Persian Gulf, of about 220 Km width, as an example.

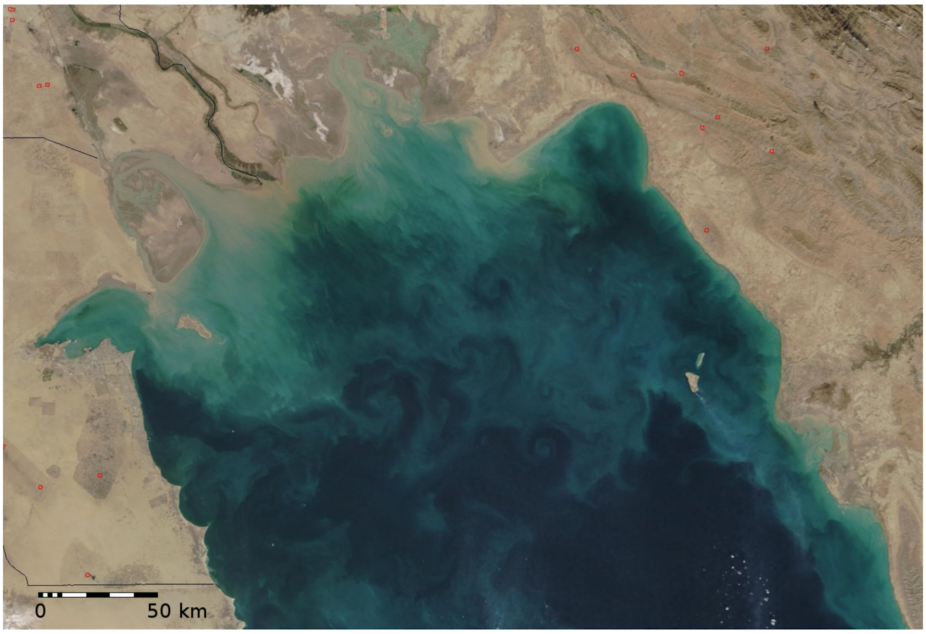


Figure 2.5. The characteristic length of large eddies are of the same order as the mean flow (Part of Persian Gulf, Modis, 04Nov02)

The physical procedure mentioned above clarifies that the effects of large and small turbulence scales on the mean flow are not the same. Momentum exchange and diffusion are mainly affected by larger scales. Hence larger eddies are more important when defining turbulence diffusive effects on the mean velocity field. It means that modelling only larger eddies might be sufficient for diffusion calculation in many cases. On the other hand, turbulence energy dissipation must be known too, which is affected by the smallest scales of turbulence. Nevertheless, modelling smaller scales is not always necessary. There is not any other source of turbulence energy except the largest eddies. The amount of energy that transfers to the smallest eddies (and dissipates), is controlled by the production rate of the large eddies. Therefore it may be possible to use some kind of dissipation modelling based on data from larger eddies (via calibration factors). In spite of the fact that the scale of the smallest eddies is controlled by the fluid viscosity, this property does not control the amount of energy dissipation.

Figure (2.6) illustrates a schematic turbulence energy distribution versus the eddy scale. The vertical axis illustrates the turbulence energy spectrum $E(\kappa_e)$ while the horizontal axis illustrates the wave number κ_e standing for the eddy scale.

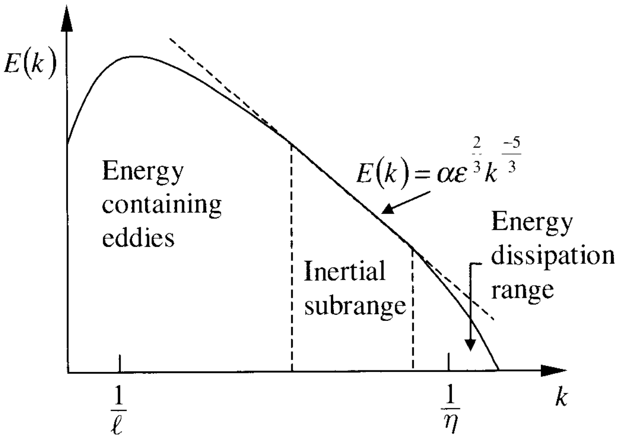


Figure 2.6. Energy spectrum versus spatial scale, log-log scales (after Chung, 2002)

The turbulence energy spectrum is related by the turbulence kinetic energy as follows (Chung, 2002):

$$k=\frac{1}{2}\cdot\overline{u'_i\cdot u'_i}=\int_0^\infty E(\kappa_e)\cdot d\kappa_e \tag{2.28}$$

The distribution of energy spectrum $E(\kappa_e)$ versus wave number κ_e is divided into three regions as shown in Figure (2.6): the region of energy containing large eddies (small κ_e), followed by the inertial sub-range (medium κ_e) and energy dissipation sub-range at smallest eddy scales (large κ_e). The wave numbers l_k and η_k in figure (2.6) are the energy bearing length scale and the so called Kolmogorov micro-scale, respectively (Chung, 2002). The structure of large eddies is different from the small ones. Largest eddies are highly anisotropic (directional). The smallest eddies are isotropic (non-directional) because of the diffusive action of viscosity (Versteeg and Malalasekera, 1995).

The use of computer has altered what is considered as “mean flow modelling” and what is considered as “turbulence modelling”. The modern computational fluid dynamics definition of turbulence modelling is simply all of the phenomena due to the irregular motion that occurs at scales below those resolvable on the grid employed for computational purposes. What is turbulence modelling to one modeller using a coarse grid may then constitute resolvable fluid flow to another modeller using a finer grid (Abbott and Basco, 1989). Research on turbulence has started from 1895 and still continues. It is difficult to summarize a complete and still compact history of all the of relevant works.

2.3.1 Modelling Approaches

There are several possible approaches for turbulence simulation. The most natural and accurate way is to simulate the original N.S. governing equations over the whole range of turbulent scales. This is the so called Direct Numerical Simulation (DNS) method. By DNS, a very refined mesh is used so that all of the eddy scales, large and small, are resolved. This approach is not feasible for general engineering computations. The requirements on mesh resolution and time-step size force very high demands on computer resources. For example something about 10^{10} grid points may be needed for a typical problem and the time step must be proportional to grid space.

For most engineering purposes, defining the turbulence effect of the mean flow is sufficient, with details of turbulent fluctuations being not needed. Therefore a more practical idea is to simulate just the effect of turbulence on the mean flow and leave the details of turbulence structure unresolved. This idea was introduced by Reynolds (1895). The basis of this idea is to separate the mean and fluctuating values of each variable in conservation equations and derive some N.S.-like relationships for the mean values. This is a statistical approach, which eliminates the grid and time resolutions needed for DNS. Different filtering or averaging[1] algorithms are available for such a statistical approach:

1. Time averaging in a fixed point of space, for stationary turbulence,
2. Space filtering or averaging for a fixed moment in time in the case of homogeneous turbulence,

[1] Time and space averaging are specific kind of filtering. Averaging results in a filtering-out of certain wave periods or length-scales of information (Abbott and Basco, 1989).

3. Ensemble averaging for a series of identical experiments. This is the most general form of averaging.
4. Mass (Favre) averaging, which is used for compressible flows.

The first major statistical approach with turbulence is time averaging on the Navier-Stokes equations (Reynolds, 1895). This is the so called Reynolds averaged Navier-Stokes (RANS) approach. In this way, the whole of turbulence spectrum is modelled (not simulated directly) . There are several versions of RANS method.

The second major statistical (or semi-statistical) approach is Large Eddy Simulation (LES). Its key idea is space filtering of N.S. equations. It separates the turbulence frequency spectrum into two parts (sub-grid and super-grid scales). The large eddies are resolved directly, while the influence of smaller ones is modelled.

A brief description about RANS and LES approaches follows.

2.3.2 Temporal Averaging (Reynolds Averaging)

Engineers are normally interested in knowing just a few quantitative properties of a turbulent flow, such as the average quantities of pressure and velocity (Ferziger and Peric, 2002). The most important feature of turbulence from an engineering point of view is its enhanced diffusivity, which greatly enhances the transfer of mass, momentum and energy by orders of magnitude. Even if we had a complete time history of a turbulent flow, we would usually integrate the flow properties of interest over time to extract time-averages (Wilcox, 1994). The idea of Reynolds-averaging is to filter out variable fluctuations from the governing equations. Thus the equations and the variable quantities must be averaged on time. Time averaging procedure for an unsteady variable, ϕ , is as follows:

$$\phi=\frac{1}{T}\int_{0}^{T}\phi\,dt+\phi'=\overline{\phi}+\phi' \qquad (2.29)$$

For a steady mean flow, the time interval, T, goes to infinity. For an unsteady mean flow, T is chosen large enough with respect to the time scale of the turbulent fluctuations, but has to remain small with respect to the time scales of other time-dependent effects in the flow that we do not wish to regard as belonging to the turbulence. If these two time scales are not some orders of magnitudes different, then time averaging is not adequate (Hirsch, 1988; Wilcox, 1994). In such a condition, the mean of a property at time t is taken to be the average of the instantaneous values of the property over a large number of repeated identical experiments; the so-called ensemble averaging (Versteeg and Malalasekera, 1995).

By averaging, the non-linearity of the N.S. equations gives rise to new unknowns. For example, averaging the product of velocity and another instantaneous quantity leads to:

$$\overline{u_i\phi}=\overline{(\bar{u}_i+u')(\bar{\phi}+\phi')}=\bar{u}_i\bar{\phi}+\overline{u_i'\phi'} \qquad (2.30)$$

The last term of right-hand is zero only if the two quantities are uncorrelated; this is rarely the case in turbulent flows (Ferziger and Peric, 2002; Wilcox, 1994).

The non-linear terms in N.S. Equations are of the form $u_i u_j$. Hence, Reynolds averaging in 3D leads to six unknown quantities of the form $\overline{-\rho u_i'u_j'}$. They called Reynolds-stresses.

By introducing new unknowns, the momentum and continuity system of equations is no longer a close system (the unknowns are more than the relationships). In order to complete the system of equations, the Reynolds stress has to be expressed in terms of the mean quantities of velocity and pressure. However, no general law for this is known. This is known as the closure problem. In practice semi-empirical relationships are introduced, leading to the so-called turbulence models.

Several approaches have been introduced by researchers to define Reynolds stresses. Many of them are based on Boussinesq approximation.

2.3.2.1 Boussinesq Approximation

Boussinesq eddy viscosity concept relates the time-averaged Reynolds stresses to the time-averaged velocity gradients as follows:

$$\overline{-u_i'u_j'}=\nu_t\left(\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}-\frac{2}{3}\frac{\partial\bar{u}_k}{\partial x_k}\delta_{ij}\right)-\frac{2}{3}k\delta_{ij} \qquad (2.31)$$

where v_t is the turbulent eddy kinematic viscosity and k is the turbulent kinetic energy.

Boussinesq eddy-viscosity approximation assumes that the principal axes of the Reynolds-stress tensor are coincident with those of the mean strain-rate tensor at all points in a turbulent flow. The coefficient of proportionality between these two is the eddy viscosity, v_t (Wilcox, 1994). This approach is based on the presumption that there exists an analogy between the action of viscous stresses and of Reynolds stresses on the mean flow (Versteeg and Malalasekera, 1995).

Models based on the Boussinesq approximation provide excellent predictions for many flow types of engineering interest, but not for all of them. Some applications for which this approximation fails are sudden change in the mean strain rate, flow over curved surfaces, and flow in ducts with secondary motion and flows with boundary-layer separation (Wilcox, 1994).

2.3.2.2 Classification of RANS Turbulence Models

The most regular RANS turbulence models may be classified as follows:

1. Zero equation models
2. One equation models
3. Two-equation models
4. Reynolds stress simulation models
5. Algebraic Reynolds stress models
6. Other turbulence models

The titles of the first three classes refer to the number of differential equations used for transport (advection-dispersion) of turbulence quantities. A brief review of each class follows.

2.3.2.3 Zero Equation Models (Algebraic Models)

In an algebraic model the eddy viscosity is estimated via a simple algebraic relationship. The relationship involves local velocity characteristics (for example local velocity gradient) in addition to some empirical constants. Some of the famous zero-equation models are:

- mixing length model (after Prandtl, 1925),
- free shear layer model (after Prandtl, 1945) which is a specific case of mixing length,
- mixing length models with correction factors (for example Van Driest, 1956 and Cebeci, Smith, 1974), and
- two-layer or N-layer mixing length models (for example Baldwin Lomax, 1978 and Degani and Schiff, 1986).

In a mixing length model the eddy viscosity in shear layers is estimated by the following relationship:

$$v_t = l_m^2 \left| \frac{\partial \bar{u}}{\partial y} \right| \tag{2.32}$$

where y axis is normal to the boundary and l_m is the mixing length. The value of l_m has been calibrated for different kind of flows via many research works.

The mixing length is the distance in the y direction, in which the eddy lumps retain their x-directed momentum. Near solid boundaries, the mixing length value is proportional to distance from the surface. For free shear flows such as the jet, wake and mixing layers, the mixing length value is proportional to the width of the layer. The value of the mixing length can be calibrated for specific class of flows.

Zero-equation models assume local equilibrium for production and dissipation of turbulence at each point of the velocity field. They are simple and easy to use in comparison to other turbulence models. They rarely cause unexpected numerical difficulties (Wilcox, 1994). Each Zero-equation model works well only for flow types for which it has been tuned.

2.3.2.4 One-Equation Models

Local equilibrium assumption is an important restriction for zero-equation models. In most cases turbulence eddies migrate across the flow. Therefore turbulent behaviour at a given position depends upon upstream history too. One-equation models are the first answer to this problem. They simulate one of the turbulence characteristics via an advection-dispersion transport equation. Considering that at least two properties are needed to have a good estimation from turbulence (for example turbulent kinetic energy and the mixing length or the energy dissipation), one-equation models relay on calibration for the other property. through one-equation modelling, the unsteadiness of turbulence in time and spaces receives attention.

In most one-equation methods turbulence kinetic energy, k, is chosen for numerical simulation[1] while the mixing length is their calibration parameter to close the problem.

The first one-equation turbulence model was introduced by Kolmogorov (1942) and then by Prandtl (1945). They assumed the following relationship between the eddy viscosity and turbulence kinetic energy:

$$\nu_t = c'_\mu \sqrt{k}\, l_m \tag{2.33}$$

where c'_μ is an empirical constant and l_m is the mixing length.

The transport equation for turbulence kinetic energy is derived from Navier-Stokes equations. Its exact form for high Reynolds numbers is as follows:

$$\frac{\partial k}{\partial t} + u_i \frac{\partial k}{\partial x_i} = -\frac{\partial}{\partial x_i}[u'_i(\overline{\frac{u'_i u'_j}{2} + \frac{P}{\rho}})] - \overline{u'_i u'_j}\frac{\partial u_i}{\partial x_j} - \beta_t g_i \overline{u'_i \rho'} - \nu \frac{\overline{\partial u'_i}}{\partial x_j}\frac{\partial u_i}{\partial x_j} \tag{2.34}$$

From left to right, the constituent terms are: temporal change, transport of turbulence energy with mean velocity, diffusion of turbulence because of itself and pressure effect[2], turbulence

[1] Some one-equation models use other turbulence characters. For example Bradshaw (1967) proposed a one equation model which solves a transport equation directly for turbulence shear stress.

[2] In many applications, the fluctuating-pressure flow-work term is often neglected. This is primarily because p is poorly correlated with velocity except near boundaries (Abbott and Basco, 1989).

production by shear stress, buoyancy effect (may be positive or negative), and dissipation of turbulence energy in the smallest eddies.

Most of existing turbulence models use a simplified version of equation (2.34). The cost of simplification is some extra constants, which have been defined empirically. The simplified equation is given as:

$$\frac{\partial k}{\partial t}+u_i\frac{\partial k}{\partial x_i}=\frac{\partial}{\partial x_i}(\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_i})+\nu_t(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i})\frac{\partial u_i}{\partial x_j}-\frac{g}{\rho}\frac{\nu_t}{\sigma_k}\frac{\partial \rho}{\partial z}-C_D\frac{k^{3/2}}{l_m} \qquad (2.35)$$

The mixing length l_m in the above equation must be calibrated for each flow regime. Many researchers have focused on estimating this value for different flow types. There is a valuable literature about it. The zero-equation mixing length model from Prandtl is a specific simplified form of equation (2.35).

The one-equation turbulence model given in (2.35) is applicable for high Reynolds numbers. It is not adequate for flow types with low Reynolds numbers such as near wall flows.

There are also some other one-equation models, such as Bradshaw (1967), Baldwin and Barth (1990), Goldberg (1991), and Spalart and Athmaras (1994), and Menter (1997).

2.3.2.5 Two-Equation Models

Estimating the mixing length in one equation models is not always an easy and accurate task. This method cannot estimate turbulence behaviour accurately for a large number of hydro-environmental problems. Especially in situations where two or more flow regimes exist at the same time (for example jet flow and boundary shear flow). For a more general RANS model, at least two characteristics of turbulence must be simulated numerically via the advection diffusion equations. It is not necessary to simulate the mixing length directly as the second character. A combination of ($k^n \cdot l_m^{\ m}$) may also be used. In fact, all of known two-equation models use something different from l_m itself. Several combinations have been used by researchers such as $k^{1/2}/l_m$ (Kolmogorov), k/l_m (Rotta), $k^{1/2}/l_m$ which is turbulence vortex (Spalding), and $k^{3/2}/l_m$ which is turbulence dissipation (by most researchers).

The first two-equation turbulence model is introduced by Kolmogorov (1942). The second quantity which he used was the energy dissipation rate per volume per time (ω). The relationships between this parameter and energy dissipation and the mixing length are as follows:

$$\epsilon=k\cdot\omega \qquad (2.36)$$

$$l_m = k^{1/2}/\omega \tag{2.37}$$

A wide variety of k-ε models exists. The most common versions are the standard k-ε model by Launder and Spalding (1972) and the RNG k-ε model by Yakhot and Orszag (Jasak, 1996).

The standard k-ε model has gained the most attention from researchers and model users. In addition to equation (2.35) for turbulence kinetic energy, it involves another equation for turbulence dissipation. The dissipation equation, like the kinetic energy equation, has been derived from N.S. equations. Its original form involves several complicated non-linear terms, which are difficult to simulate. Most k-ε models use a simplified version of the original ε equation. The simplification is based on some engineering assumptions and empirical constants. Here is the standard form of k-ε model:

$$\frac{\partial k}{\partial t}+u_i\frac{\partial k}{\partial x_i}=\frac{\partial}{\partial x_i}\left(\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_i}\right)+\nu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j}+\frac{g_i}{\rho}\frac{\nu_t}{\sigma_t}\frac{\partial \rho}{\partial x_i}-C_D\frac{k^{3/2}}{l_m} \tag{2.38}$$

$$\frac{\partial \epsilon}{\partial t}+u_i\frac{\partial \epsilon}{\partial x_i}=\frac{\partial}{\partial x_i}\left(\frac{\nu_t}{\sigma_\varepsilon}\frac{\partial \epsilon}{\partial x_i}\right)+\frac{\epsilon}{k}\left(c_{1\varepsilon}\nu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j}+c_{3\varepsilon}\frac{g_i}{\rho}\frac{\nu_t}{\sigma_t}\frac{\partial \rho}{\partial x_i}-c_{2\varepsilon}\epsilon\right) \tag{2.39}$$

$$\Gamma=\frac{\nu_t}{\sigma_t},\ \nu_t=c_\mu\frac{k^2}{\epsilon} \tag{2.40}$$

The second equation relates turbulence energy dissipation to mean flow and to large eddies. How it is possible when dissipation occurs just in the smallest eddies? The reason is that the smallest eddies receive their energy from larger ones via the so-called energy cascade and the energy cascade is controlled by larger eddies.

The dissipation equation (2.39) involves some empirical factors ($\sigma_\varepsilon, c_{1\varepsilon}, c_{2\varepsilon}, c_{3\varepsilon}$) as part of simplification cost. The last factor is just for buoyancy effect. It may be eliminated in situations where buoyancy has no effect. For the standard k-ε model, the empirical constants have been calibrated as follows:

Table 2.1. Empirical constants in the standard k-ε model

c_μ	$c_{1\varepsilon}$	$c_{2\varepsilon}$	$c_{3\varepsilon}$	σ_k	σ_ε	σ_t
0.09	1.44	1.92	0.0 – 1.0	1.0	1.3	0.8

The result from a k-ε model simulation is highly sensitive to the factors $c_{1\varepsilon}$ and $c_{2\varepsilon}$. With buoyant flow, the values of σ_t and $c_{3\varepsilon}$ must be calibrated per case. The values in table (2.1) are not the best choices for every kind of flow. It is possible to extent and improve the standard k-ε model by fine-tuning its constants for specific flow regimes or via replacing by carefully-chosen functions. Many researchers have been worked on this matter. Among them are Yakhot et al (1992), Chukkapalli and Turan (1995), Chang et al (1995) and Hwang et al (1998). Some two-equation models are not based on Boussinesq approximation. They are some kind of second-order closer models.

The RNG k-ε model from Yakhot et al (1992) has gained a lot of attention by other researchers. It is a second-order closer model. It tries to solve accurately the non-linear terms by orders more than two. It is based on a statistical technique called Re-normalization Group (RNG) theory. This model does not restricted to high-Reynolds numbers. It introduces effective viscosity coefficient (ν_{eff}) instead of normal eddy viscosity. So ν_{eff} is used instead of ($\nu+\nu_t$).

The effective viscosity is given as:

$$\nu_{eff}=\nu_{mol}\left(1+\sqrt{\frac{c_\mu}{\nu_{mol}}\frac{k}{\sqrt{\epsilon}}}\right)^2 \qquad (2.41)$$

where ν_{mol} is the molecular viscosity.

The RNG k-ε equations are as follows:

$$\frac{\partial k}{\partial t}+u_i\frac{\partial k}{\partial x_i}=\frac{\partial}{\partial x_i}\left(\frac{\nu_{eff}}{\sigma_t}\frac{\partial k}{\partial x_i}\right)+\nu_{eff}\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)^2+\frac{g_i}{\rho}\frac{\nu_{eff}}{\sigma_t}\frac{\partial \rho}{\partial x_i}-C_D\frac{k^{3/2}}{l_m} \qquad (2.42)$$

$$\frac{\partial \epsilon}{\partial t}+u_i\frac{\partial \epsilon}{\partial x_i}=\frac{\partial}{\partial x_i}\left(\frac{\nu_{eff}}{\sigma_t}\frac{\partial \epsilon}{\partial x_i}\right)+\frac{\epsilon}{k}\left(c_{1\varepsilon}\nu_{eff}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j}+c_{3\varepsilon}\frac{g_i}{\rho}\frac{\nu_{eff}}{\sigma_t}\frac{\partial \rho}{\partial x_i}-c_{2\varepsilon}^{*}\epsilon\right) \qquad (2.43)$$

where:

$$c_{2\epsilon}^{*}=c_{2\epsilon}+\frac{c_\mu\eta^3\left(1-\frac{\eta}{\eta_0}\right)\epsilon^2}{1+\beta_0\eta^3}\frac{\epsilon^2}{k} \qquad (2.44)$$

Equation (2.43) is identical to the standard ε equation except that the coefficient $c_{2\epsilon}$ has been replaced by $c_{2\epsilon}^{*}$. The effect of the rate of strain appears in this coefficient.

For the RNG k-ε model, the empirical constants have been calibrated as follows:

Table 2.2. Empirical constants for the standard k- ε model

c_μ	$c_{1\varepsilon}$	$c_{2\varepsilon}$	σ_k	σ_ε	σ_t
0.0845	1.42	1.68	$1/\alpha$	$1/\alpha$	$1/\alpha$

The value of α can be obtained from the following graph:

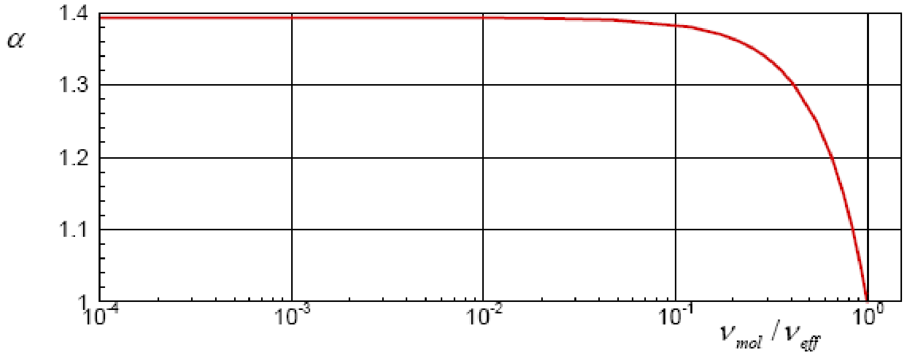


Figure 2.7. The value of α for RNG k-ε model (Fluent, 1998)

RNG k-ε model is more responsive to the effects of rapid strain and streamline curvature than the standard version.

Some other two-equation models are $k-k\,l_m$ model (by Rotta, 1968), k-ω model (by Wilcox, 1988), and realisable k-ε model (by Shih et al, 1995). There are still some other two-equation models which are from a different category. They are second order closer models such as k-τ model (by Speziale et al, 1992), another one by Nisizima and Yoshizawa (1987), and another one from Rubinstein and Barton (1990).

2.3.2.6 Reynolds Stress Simulation Models

Reynolds Stress models improve turbulence modelling in complex flow cases such as streamline curvature. They are more accurate for unsteady flows involving anisotropic turbulence Reynolds stresses. Reynolds stress models are not based on Boussinesq eddy-viscosity approximation. They are more general. Some relevant research results are due to Daly and Harlow (1970), Launder et al. (1975), Lumley (1978), Reynolds (1987) and Speziale (1985, 1991).

Reynolds stress models are very complex in comparison with models based on Boussinesq approximation. They contain six Reynolds stress tensor components ($\overline{u_i u_j}$) in addition to three scalar components of the form ($\overline{u_i \Phi}$) plus $\overline{\Phi^2}$. Therefore, there are 10 partial differential equations that must be solved together. More research is needed before Reynolds stress models can be used in practical engineering software.

2.3.2.7 Algebraic Reynolds Stress Models

As mentioned, Reynolds stress models contain several equations with complex terms. Therefore they are difficult to solve. Some researchers have tried to simplify the complex terms via reasonable assumptions. Therefore some models have been derived in which the complex equations have been simplified to simple algebraic relationships. These new models have reserved many positive properties of the original Reynolds stress models. The purpose of developing algebraic Reynolds stress models is to avoid the solution of differential equations and to obtain the Reynolds stress components directly from algebraic relationships. Difficulties still remain in geometrically and physically complicated flow fields (Chung, 2002).

Algebraic Reynolds stress models have been used successfully in cases containing buoyancy, separation and secondary eddies. Some of the relevant works are from Cebeci and Smith, Meller and Herring, Patankar andSpalding, and Michel et al.

2.3.2.8 Other turbulence models

There are some other turbulence models which have been developed for specific purposes. For example viscous effects are important near the walls, while common models do not take it into account. Some relevant research works are from Hanjalic and Launder (1976), Shima (1988) and Lia and So (1991). Such models can be found under the category of wall-functions too.

There are also some turbulence models especially developed for low-Reynolds conditions. Some relevant works are after Lam and Bermhorst (1981), Launder and Sharma (1974), Chein (1982), and Nagano and Hishida (1987).

Another category of turbulence models are multi-scale models. These models are used to simulate all parts of the turbulence energy spectrum. Some models of this category are Hanjalic et al. (1980), Kim and Chen (1989) and Durbin (1993).

2.3.3 Boundary Conditions for k and ε

The boundary conditions for k and ε, as the choice of this study, are as follows:

- inlet: k and ε must be known,
- outlet: $\partial k/\partial n=0$ and $\partial \epsilon/\partial n=0$ where *n* is perpendicular to the boundary,
- free surface: $\partial k/\partial z=0$ and $\epsilon=\left(k_\eta\sqrt{c_\mu}\right)^{3/2}/\left(0.07\kappa\left(\eta-z_b\right)\right)$,
- bed and lateral closed boundaries: wall function.

The most accurate simulations can only be achieved by supplying measured values of k and ε at inlet boundaries, but it is not possible by most cases. Thus some kind of estimation is needed.

2.3.4 Wall Function

The physics of turbulence in the vicinity of walls is considerably different from the other parts of the flow (Jasak, 1996). At the wall boundary, the velocity gradient is high, Reynolds number is low (viscous effects are important), and turbulence eddies are highly anisotropic. Normal turbulence models are not adequate for this region, because they are calibrated for high Reynolds numbers and more or less isotropic conditions. Some turbulence models have been introduced especially for low-Reynold flows. For a detailed treatment, such models must be used. In this way, the computational mesh must be very fine near the wall and a lot of computational resources are needed for that region. In addition, numerical solution of near-wall turbulence models is more complex than normal ones. Many of these models use viscous damping functions, which makes them stiff[1] (Wilcox, 1994). The numerical algorithms which are adequate (and stable) for a stiff system of equations, are more complicated than the normal ones.

There is also a simpler approach to consider near-wall turbulence into account without resolving the near-wall region. This approach based on wall functions. Wall-functions represent a simplified model of turbulence[2], which mimics the near-wall behaviour of the velocity. In numerical simulation, wall functions are used to bridge the near-wall region and couples with a high Reynolds turbulence model in the rest of the domain (Jasak, 1996).

The formula for wall function is as follows (Olsen, 2000):

$$\frac{U}{U_*}=\frac{1}{\kappa}\ln\left(\frac{30\,y}{k_s}\right) \tag{2.45}$$

in which U is near-wall velocity (it is parallel to the wall), U_* is the shear velocity, y is the distance from the wall, k_s is the wall roughness, and κ is the von Karman constant. The relationship mentioned above shows the typical velocity profile in a boundary layer. It is valid in the following range:

$$30<\frac{U_*y}{\nu}<3000 \tag{2.46}$$

where ν is the kinematic viscosity of water.

For a numerical simulation, the centre of the nearest computational cell to the bottom must be in the range mentioned above. For most engineering problems, it is not a restricted bound. For a

[1] An equation, or system of equations, is said to be stiff when there are two or more very different scales of the independent variable on which the dependent variables are changing. The equivalent situation for a system of equations is to have eigenvalues of the characteristic equation of very different magnitudes (Wilcox, 1994).

[2] The flow behaviour assumes to be like a fully developed turbulent boundary layer.

nearly steady uniform free surface flow, the boundary layer involves whole of the water column up to the free surface.

Implementing the wall laws in the CFD code, the main idea is to include a sink term for the velocity equations. The shear stress at the wall appears as a negative source in the momentum equation. The shear stress is estimated using the wall laws (Olsen, 2000). The values of k and ε near the wall are as follow (Namin, 2003):

$$k=\frac{U_*^2}{\sqrt{c_\mu}} \tag{2.47}$$

$$\epsilon=\frac{U_*^3}{\kappa\, y} \tag{2.48}$$

2.3.5 Spatial Averaging (Large Eddy Simulation)

Turbulent flow contains a wide range of length and time scales. Large scale eddies are generally much more energetic than the small scale ones; their size and strength make them by far the most effective transporters of the conserved properties (such as momentum). The small scales are usually much weaker and provide little transport of these properties. A simulation which treats the large eddies more precisely than the small ones may make sense; large eddy simulation is just such an approach (Ferziger and Peric, 2002). The underlying premise is that the largest eddies are directly affected by the boundary conditions and must be computed. By contrast, the small-scale turbulence is more nearly isotropic and has universal characteristics; it is thus more amenable to modelling (Wilcox, 1994).

Large eddy simulation (LES) is a compromise between DNS and RANS. Here, large-scale eddies are computed (resolved) and small scales are modelled (leaving them unresolved). Small-scale eddies are isotropic, hence their modelling is simpler than in RANS. The mesh refinements are required much more than in RANS, but not as much as in DNS (Chung, 2002).

In order to define a velocity field containing only the large-scale components, it is necessary to filter the variables of the governing equations. There are two major steps with LES analysis: filtering and sub-grid scale modelling. Filtering is carried out using the box function, Gaussian function, or Fourier cut-off function. Sub-grid modelling includes eddy viscosity model, structure function model, dynamic model, scale similarity model, and mixed model, among others (Chung, 2002). The filter provides a formal definition of the averaging process and separates the resolvable scales from the sub-grid scales (Wilcox, 1994).

Sufficient grid resolution is critical in LES methods. The ultimate test of grid convergence is the requirement that excessive energy must not accumulate in the smallest scales. Details of the dissipating eddies are unimportant in LES. By contrast, DNS requires accurate simulation of the dissipating eddies (Wilcox, 1994).

In the context of LES, τ_{ij} is called the sub-grid scale Reynolds stress. These stresses are there because of filtering. The models used to approximate τ_{ij} are called sub-grid scale or sub-filter scale models. The earliest and most commonly used sub-grid scale model has been introduced by Smagorinsky (1963). It is not free from some kind of calibration. There are two other approaches which do not require any externally provided information such as model constants (Ferziger and Peric, 2002). The first approach is called dynamic modelling. Dynamic models contain a procedure to estimate the sub-grid scale model parameter via filtering the velocity field using a filter broader than the one used in the LES itself to obtain a very large scale field and then subtraction of the two fields. In this way, a kind of self-consistent sub-grid scale model is produced. Any sub-grid scale model can be used as a basis for this approach. The other approach, which is more recent, is based on the deconvolution concept. These models attempt to estimate the unfiltered velocity from the filtered one (Ferziger and Peric, 2002).

The simulation time needed by LES modelling on a sufficiently refined grid is more than the simulation time needed by RANS methods. Nowadays, some published hydro-environmental models propose Smagorinsky LES method without stressing on dense grid. In this way, simulation is faster than many RANS models. However, adjusting the calibration factor is more difficult and must be addressed case by case. In addition, for certain turbulence behaviour, the value of the calibration factor depends on the distance between the grid points. The calibration factor must be changed when the mesh density changes. It is not easy to control the turbulence behaviour precisely if an unstructured grid is used.

2.4 Turbidity Current and Its Influence on Turbulence

For some engineering problems, non-homogeneity of water density acts as a driving force or stabilizer, which affects the velocity field and sediment transport. In particular, this phenomenon is the main factor in stratified flow and turbidity current. The cause of non-homogeneity in density is non-homogeneity in temperature, salinity or suspended sediment concentration. The temperature-, sediment- or salt-caused density gradients may have similar or opposite sign and the aggregate effect will determine which kind of flow type may occur. Turbidity current in dam reservoir is a specific type of density current. It is the most effective density current detected in reservoirs.

Turbidity current is the flow of water laden with sediment that moves down-slope in otherwise still water of reservoir. Turbidity current may occur in ocean, lake and reservoir. Turbidity current in dam reservoir is one of the most interesting and practically important kinds of density driven flows. The dynamic of turbidity flow has major impact on water quality and sedimentation pattern in reservoirs. Turbidity current can transport sediment a long way inside the reservoir, even cause sediment deposition near the dam in front of water intakes. The process also redistributes sediments from the river mouth to the deeper part of the lake/reservoir. It has the ability to carry large quantities of sediment, nutrients and various chemical substances. Field measurement at several reservoirs has showed the velocity range between 0.03 to 0.5 m/s and travel distance up to 100 km without notably mixing/vanishing in the ambient water (Batuca, 2000; Olsen, 2004). Figure (2.8) illustrates phases of turbidity current in a reservoir.

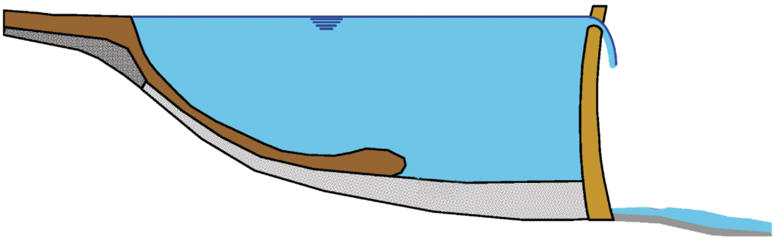


Figure 2.8. Typical turbidity current in a reservoir (after Cesare, 1998)

During significant flood events, the suspended sediment concentration at the upstream river becomes very high. Therefore the density of inflowing water becomes much more than the density of still water in the reservoir. The heavier muddy inflow continues its motion as a momentum-dominated flow. Meanwhile, water depth increases and current speed decreases continuously because of the bed slope. Finally the muddy flow reaches to a specific depth where the current velocity is too low to remain dominant. The buoyancy effect becomes dominant at this stage. Then plunging occurs; the muddy water separates from the water surface and pulls down toward the reservoir bottom. After plunging, the turbid water continues his forward motion as a separate layer towards the downstream end of the reservoir, without notable mixing with ambient water.

The plunge point is often visible at the water surface, as the river water contains more sediment and has a different colour than the water in the reservoir. The sediment-laden river water often has a darker colour than the cleaner reservoir water. It is therefore sometimes possible to observe the plunge point at the water surface (Olsen, 2004).

After the plunging point, turbidity current becomes buoyancy dominated. Most of reservoirs have a sedimentation region called "delta[1]". The bottom slope at the front of delta region is relatively sharp. When turbidity current travels in this region, its velocity increases significantly and often it is able to erode the bottom material. Bed erosion leads to more suspended sediment concentration in the turbidity layer. After the delta region, the bottom slope becomes gentle. If the bottom slope changes rapidly or has discontinuity at this position, a change in turbidity current regime may occur from supercritical to sub-critical flow via a hydraulic jump. Large amounts of erosion and sedimentation may accrue in different parts of reservoir during flood events by turbidity current. Figure (2.9) illustrates typical processes at the front of the turbidity current.

Buoyancy can influence the turbulence structure significantly, wherever there is large density gradient in vertical direction. When the heavier water is above the clear water, the situation is unstable which leads to increase in turbulence. When the heavier water is below the clear water, the stratification is stable which leads to decrease in turbulence. There is a sharp density gradient at the interface between turbidity layer and the ambient water, which is of stable stratification type. This phenomenon decreases the turbulence viscosity significantly at the interface. This is the key reason which prevents mixing and permits forward movement of turbidity current for a long time without major mixing with ambient water.

[1] In the vertical cross section, it is similar to a reverse Δ in the longitudinal direction.

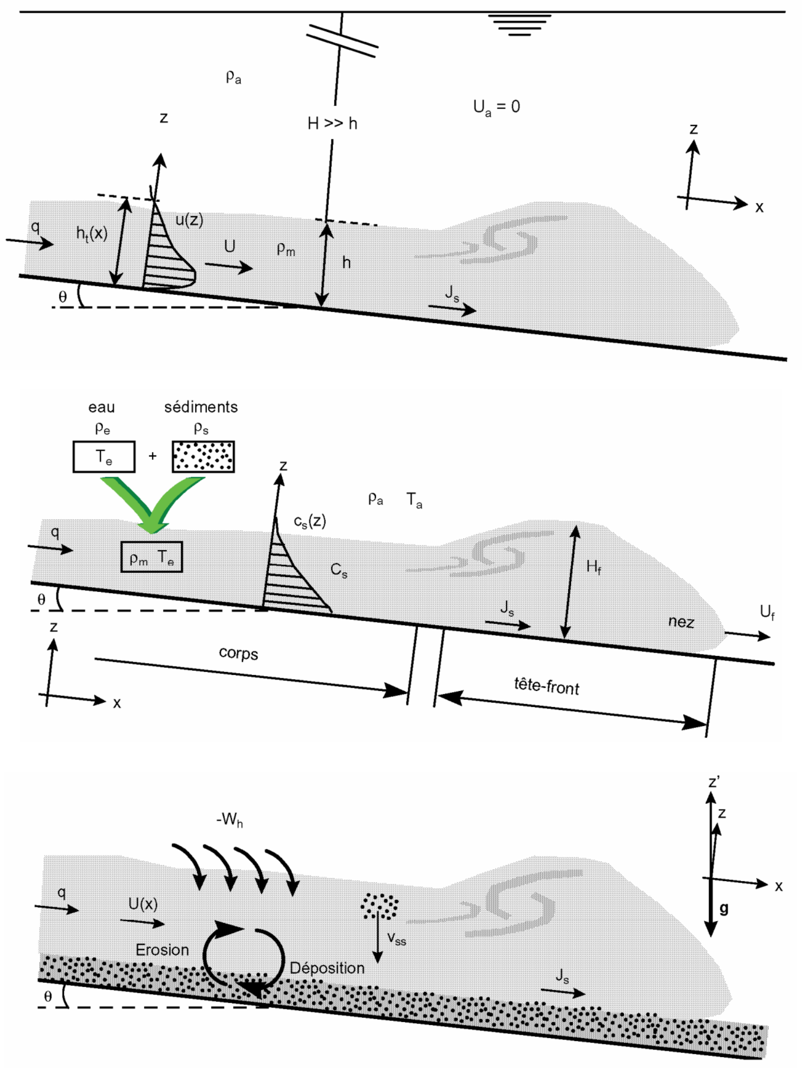


Figure 2.9. Typical processes at the front of turbidity current (after Cesare, 1998)

It is not possible to simulate turbidity current without taking the buoyancy effect on turbulence into account. There are two approaches for buoyant turbulence simulation. The first one is to add buoyancy terms into the k-ε equations. This is the best way if the k-ε model has been selected for turbulence. It must be noted that it will be more difficult to satisfy the stability of k-ε solution when buoyancy terms are effective. The second approach is to correct the values using an estimated analytic formula after normal computation of diffusion/dispersion coefficients (without buoyancy effect). This is the only possible approach when simple methods for turbulence modelling, such as Prandtl method, are used.

The buoyancy terms in the k-ε equations are been shown below:

$$\frac{\partial k}{\partial t}+u_j\cdot\frac{\partial k}{\partial x_j}= \frac{\partial}{\partial x_j}\left(\frac{\upsilon_t}{\sigma_k}\cdot\frac{\partial k}{\partial x_j}\right)+\upsilon_t\cdot\frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\overbrace{\frac{g_i}{\rho}\cdot\frac{\upsilon_t}{\sigma_t}\cdot\frac{\partial \rho}{\partial x_i}}^{buoyancy\ effect}-c_D\cdot\frac{k^{3/2}}{l_m} \tag{2.49}$$

$$\frac{\partial \epsilon}{\partial t}+u_j\cdot\frac{\partial \epsilon}{\partial x_j}= \frac{\partial}{\partial x_j}\left(\frac{\upsilon_t}{\sigma_k}\cdot\frac{\partial \epsilon}{\partial x_j}\right)+\frac{\epsilon}{k}\cdot\left(c_{1\varepsilon}\cdot\upsilon_t\cdot\frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\overbrace{c_{3\varepsilon}\cdot\frac{g_i}{\rho}\cdot\frac{\upsilon_t}{\sigma_k}\cdot\frac{\partial \rho}{\partial x_i}}^{buoyancy\ effect}-c_{2\varepsilon}\cdot\epsilon\right) \tag{2.50}$$

Table (2.1) illustrates the constant values used in equations (2.49) and (2.50).

If the buoyancy effect is not included in the k-ε equations, or if another turbulence model such as Prandtl's mixing length method has been used, then the calculated eddy viscosity and dispersion coefficients must be corrected at each time step. Different correction formulae have been proposed in the literature for different kind of density currents. The general form of them is given as one of the following two equations:

$$\begin{aligned} v&=v_o\cdot\left(1+\beta_{Ri}\cdot Ri\right)^{\alpha_{Ri}}\\ v&=v_o\cdot e^{c\cdot Ri}\end{aligned} \tag{2.51}$$

where ν_o is the eddy viscosity or diffusion/dispersion coefficient without the buoyancy effect, ν is eddy viscosity or diffusion/dispersion coefficient corrected by the buoyancy effect, and α_{Ri} and β_{Ri} are empirical constants.

The definition of Richardson number for a 2D vertical flow field is as follows:

$$Ri=-\frac{g}{\rho}\cdot\frac{\partial \rho}{\partial z}\cdot\left(\frac{\partial u}{\partial z}\right)^{-2} \tag{2.52}$$

and for a three dimensional flow (Mike3 user manual 2005):

$$Ri=-\frac{g}{\rho_0}\frac{\partial \rho}{\partial z}\left(\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2\right)^{-1} \tag{2.53}$$

Some of the proposed formulas for eddy viscosity and dispersion coefficient, under different kind of density currents, are as follows:

Munk and Anderson (1948):

$$\begin{aligned} v &= v_o\left(1+10\,Ri\right)^{-1/2} \\ D_t &= v_o\left(1+3.33\,Ri\right)^{-3/2} \end{aligned} \tag{2.54}$$

Pacanowski and Philander (1981):

$$\begin{aligned} v &= v_o\left(1+5\,Ri\right)^{-2}+10^{-4} \\ D_t &= v\left(1+5\,Ri\right)^{-1}+10^{-5} \end{aligned} \tag{2.55}$$

Lehfeldt and Bloss (1988):

$$\begin{aligned} v &= v_o\left(1+3\,Ri\right)^{-1} \\ D_t &= v_o\left(1+3\,Ri\right)^{-3} \end{aligned} \tag{2.56}$$

Park and Kuo (1993):

$$\begin{aligned} v &= v_o\left(1+0.75\,Ri\right)^{-1/2} \\ D_t &= v_o\left(1+0.75\,Ri\right)^{-3/2} \end{aligned} \tag{2.57}$$

Perrels and Karelse (1981):

$$\begin{aligned} v &= v_o\cdot e^{-4\,Ri} \\ D_t &= v_o\cdot e^{-15\,Ri} \end{aligned} \tag{2.58}$$

Rodi (1993):

$$v = v_o\,.\,e^{-C\,.\,Ri} \tag{2.59}$$

Olsen and Lysne (2000) for ice-covered lake:

$$v = v_o\left(1+10\,Ri\right)^{-1.3} \tag{2.60}$$

Some modellers do not use buoyancy terms in their k-ε simulation to gain more stability. They use Richardson correction after each k-ε simulation step.

For most hydro-environmental engineering problems, buoyancy does not affect the turbulence structure in the horizontal plane. It just affects the turbulence structure in the vertical direction. The difference between horizontal and vertical turbulence structures will be described later.

When water is abstracted from a stratified reservoir, different situations may occur. When water discharge is less than a critical value, only the water in a layer at the same level (as the intake) is abstracted. For this case, the thickness of the abstracting layer depends on the rate of discharge, water density and vertical density gradient. When water discharge is above the critical value, then water will be abstracted from the whole depth (Olsen, 2004).

The effect of buoyancy on turbulence is not the only matter that a model developer must take into account for density currents. Including gravity directly into the calculation, and letting density varies according to the resulting concentration field, result in large source terms that may cause instabilities. From a mathematical point of view, these large source terms must cancel out each other, but in a numerical model the differences between the large source terms may cause instability.

A more stable approach for density flow is to include just the difference from a base value, instead of including absolute density values[1]. Gravity current will usually move close to the bed, giving large source terms only in cells in this region. Most of the cells may therefore not be affected by large source terms. Also, the density increase is usually one or two orders of magnitude smaller than the water density. The magnitude of the source terms therefore decreases correspondingly. Thus, the additional source term in the vertical direction in the Navier-Stokes equations becomes (Olsen, 2000):

$$Source_{vertical} = V_{cell} \cdot g \cdot (\rho_s - \rho_w) c \tag{2.61}$$

where V is the volume of computational cell, g is the acceleration of gravity, ρ_s and ρ_w are the sediment and water density respectively, and c is the sediment concentration.

Simulating horizontal density gradients for stratified lakes is difficult too. Modelling stratified flows using gravity in the vertical direction would lead to large source terms and instabilities. Adding additional forces due to density deviation from a standard water density would affect most of the cells in the geometry, as the density variation is often over the whole depth of a lake. Instabilities would thereby occur. One solution is to compute and include density gradients only in the horizontal directions, as in quasi-3D models (Olsen, 2000).

2.5 Effects of Difference in Vertical and Horizontal Scales

For a problem where the vertical scale is of the same order as the horizontal scales, turbulence eddies are mostly isotropic and completely three-dimensional. With hydro-environmental

[1] Avoiding large source terms which cancel each other, is a common strategy in numerical modelling. An example is the approach for eliminating the hydrostatic part of pressure and use of excess pressure in order to prevent flow caused by computer round-off errors (for example see Mike3 HD scientific documentation).

engineering problems, the vertical scale is some order of magnitudes smaller than the horizontal ones. This kind of problem has two specific behaviours.

The first specific behaviour is that such a problem has two different turbulence structures, which are more or less independent from each other. One of them is for large horizontal 2D eddies, which are much larger than the water depth. The other is for small nearly-isotropic 3D eddies produced by the bottom friction. Both phenomena are effective and must be taken into account. Sometimes the second phenomenon is called sub-depth scale turbulence[1].

The effect of the second specific behaviour on modelling hydro-environmental problems is that the numerical model grid for such a problem is inhomogeneous; the grid spacing in the vertical direction is some order of magnitudes less that grid spacing in horizontal plane.

Because of the two mentioned characteristics, it is necessary for hydro-environmental modellers to use different turbulence models in the vertical and horizontal directions. Even when the two models are of one type, the calibration parameters may be radically different. Therefore the momentum equations may read as follows:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\frac{\nu_h}{\rho}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{1}{\rho}\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u}{\partial z}\right)+\frac{1}{\rho}BodyForces_x \quad (2.62)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\frac{\nu_h}{\rho}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+\frac{1}{\rho}\frac{\partial}{\partial z}\left(\nu_v\frac{\partial v}{\partial z}\right)+\frac{1}{\rho}BodyForces_y \quad (2.63)$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\frac{\nu_h}{\rho}\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}\right)+\frac{1}{\rho}\frac{\partial}{\partial z}\left(\nu_v\frac{\partial w}{\partial z}\right)+\frac{1}{\rho}BodyForces_z \quad (2.64)$$

If k-ε model has been selected for calculating the vertical eddy viscosity, then the production terms in the k and ε equations have to be modified. Such production terms involve just the vertical gradients of the horizontal velocities in this regard, without taking the horizontal gradients into account. Therefore the production terms in the equations (2.38), (2.39), (2.49) and (2.50) have to be reduced as follows:

For the k-equation:

$$\nu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j} \;\rightarrow\; \nu_v\left(\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2\right) \quad (2.65)$$

For the ε-equation:

$$c_{1\varepsilon}\frac{\varepsilon}{k}\nu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j} \;\rightarrow\; c_{1\varepsilon}\frac{\varepsilon}{k}\nu_v\left(\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2\right) \quad (2.66)$$

[1] for example see Nadaoka and Yagi, 1998

There are also simpler turbulence models for calculating the vertical eddy viscosity which are sufficient in many cases. One of the common models is the log-law relationship which gives the standard parabolic profile (Mike3/21 2007):

$$\nu_v = U_{*max} \cdot D \cdot \left(0.41 \left(\frac{z - z_b}{z_s - z_b} \right) - 0.41 \left(\frac{z - z_b}{z_s - z_b} \right)^2 \right) \tag{2.67}$$

where $D = z_s - z_b$ is the total water depth and U_{*max} is the maximum of the bed and the free surface shear velocities.

Another challenge for hydro-environmental models is that they have to use computational cells which have large aspect ratios (horizontal space to vertical space). It must be taken into account when choosing a numerical scheme.

The problems mentioned here prevent many classical fluid dynamic models (originated from the mechanical engineering sector) being used for hydro-environmental simulations.

2.6 Shallow Water Modelling

The flow of water in shallow regions is of obvious practical importance. For many engineering problems, the horizontal velocities are of primary importance and the problem can be reasonably approximated in two dimensions. Consequently, shallow water models are practical tools for many engineering problems in the hydro-environmental modelling sector. They result from simplifying the original 3D equations by assuming that the vertical acceleration of flow is negligible, the flow field is nearly horizontal, and the distribution of flow variables along depth are not far from homogeneity.

The integration of the original 3D equations (equations 2.17-2.19) over the water depth, in addition to the mentioned assumptions, leads to the following equations:

$$\frac{\partial \eta}{\partial t} + \frac{\partial (DU)}{\partial x} + \frac{\partial (DV)}{\partial y} = D Q_{dis} \tag{2.68}$$

$$\begin{aligned} \frac{\partial (DU)}{\partial t} + \frac{\partial (\beta D U U)}{\partial x} + \frac{\partial (\beta D U V)}{\partial y} &= f D V - g D \frac{\partial \eta}{\partial x} + \frac{\tau_{sx}}{\rho} - \frac{\tau_{bx}}{\rho} \\ &+ \bar{\nu}_h D \left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \right) + D u_{dis} Q_{dis} \end{aligned} \tag{2.69}$$

$$\frac{\partial(DV)}{\partial t}+\frac{\partial(\beta DUV)}{\partial x}+\frac{\partial(\beta DVV)}{\partial y}=-fDU-gD\frac{\partial\eta}{\partial y}+\frac{\tau_{sy}}{\rho}-\frac{\tau_{by}}{\rho}$$
$$+\bar{\nu}_h D\left(\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2}\right)+D v_{dis} Q_{dis} \tag{2.70}$$

where D is the water depth, $\bar{\nu}_h$ is the depth averaged eddy viscosity in the horizontal plane, U and V are the depth averaged velocities defined by:

$$U=\frac{1}{D}\int_{z_b}^{\eta} u\,\mathrm{dz} \text{ and } V=\frac{1}{D}\int_{z_a}^{\eta} v\,\mathrm{dz}\,, \tag{2.71}$$

Q_{dis} is the source discharge, u_{dis} and v_{dis} are the velocity components of the discharged source, f is the Coriolis factor, $\tau_{sx}=\nu_t(\partial u/\partial z)_\eta$ and τ_{sy} are the wind shear stress components at the water surface, and $\tau_{bx}=\nu_t(\partial u/\partial z)_{z_b}$ and τ_{by} are the bed shear stress components.

Equation (2.68) has been derived by integrating the continuity equation (2.17) along the water depth. Equations (2.69) and (2.70) have been derived by integrating the x- and y-momentum (N.S.) equations (2.18) and (2.19). Integration of the z-momentum equation leads to the hydrostatic pressure distribution formula, as follows:

$$-\frac{1}{\rho}\frac{\partial p}{\partial z}=g \tag{2.72}$$

The factor β in equations (2.69) and (2.70) is due to the non-homogeneity of velocity distribution in the vertical direction. It can be assumed to be equal to 1.0 for many engineering problems; for example when the vertical distribution of velocity is logarithmic such as tidal or channel flows. When wind induced current is dominated, the factor is different.

The main assumption for the validity of shallow-water equations is that the water depth is much smaller than the horizontal length scales. It implies that the wavelength of surface waves should be much larger than the depth (more than 20 times) (Wesseling, 2001)

Assuming that $\beta=1$ and using the continuity equation (2.68), the left hand side of equation (2.69) may be expanded and simplified as follows:

$$\begin{aligned}
&\frac{\partial(DU)}{\partial t}+\frac{\partial(DUU)}{\partial x}+\frac{\partial(DUV)}{\partial y}\\
&=D\frac{\partial U}{\partial t}+U\frac{\partial D}{\partial t}+DU\frac{\partial U}{\partial x}+U\frac{\partial(DU)}{\partial x}+DV\frac{\partial U}{\partial y}+U\frac{\partial(DV)}{\partial y}\\
&=D\frac{\partial U}{\partial t}+DU\frac{\partial U}{\partial x}+DV\frac{\partial U}{\partial y}+U\left(\frac{\partial D}{\partial t}+\frac{\partial(DU)}{\partial x}+\frac{\partial(DV)}{\partial y}\right)\\
&=D\frac{\partial U}{\partial t}+DU\frac{\partial U}{\partial x}+DV\frac{\partial U}{\partial y}+UDQ_{dis}
\end{aligned} \tag{2.73}$$

The right hand side of equation (2.70) may be simplified in a similar way. Using these simplifications in equations (2.69) and (2.70) and dividing by D, the non-conservative forms of the momentum equations may be derived as follows:

$$\begin{aligned}&\frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+V\frac{\partial U}{\partial y}=fV-g\frac{\partial \eta}{\partial x}+\frac{\tau_{sx}}{\rho D}-\frac{\tau_{bx}}{\rho D}\\&-\frac{1}{\rho}\frac{\partial p_a}{\partial x}-\frac{gD}{2\rho}\frac{\partial \rho}{\partial x}+\bar{\nu}_h\left(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}\right)+(u_{dis}-U)Q_{dis}\end{aligned} \tag{2.74}$$

$$\begin{aligned}&\frac{\partial V}{\partial t}+U\frac{\partial V}{\partial x}+V\frac{\partial V}{\partial y}=-fU-g\frac{\partial \eta}{\partial y}+\frac{\tau_{sy}}{\rho D}-\frac{\tau_{by}}{\rho D}\\&-\frac{1}{\rho}\frac{\partial p_a}{\partial y}-\frac{gD}{2\rho}\frac{\partial \rho}{\partial y}+\bar{\nu}_h\left(\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2}\right)+(v_{dis}-V)Q_{dis}\end{aligned} \tag{2.75}$$

It is possible to solve the shallow water equations for the velocities (U, V) or the fluxes (DU, DV). From the mathematical point of view, they are the same, but the numerical solution of the flux form is generally more stable. The reason is that rapid change in the bed topography or the channel cross section leads to rapid change in the velocity, while the flux still changes gradually (tracking sudden change is challenging for the numerical models). However, using a depth averaged model in a problem with sudden depth changes is questionable.

Using Cartesian coordinates for SWE means that the domain is assumed to be small enough to be considered flat. For very large domains, spherical coordinate system is a better choice.

2.7 Connection Conditions for Model Integration

Two approaches have been employed in the literature for combining 1D, 2D, and 3D models. One approach is to use a simpler model in the entire solution domain and a more complex model in confined sub-domains. The simpler model provides boundary conditions for the more complex model, but they are not coupled. The other approach is to fully couple the models (Wu and Li, 1992; Zhang, 1999; Verwey 2001) by simultaneously solving all component models. The fully coupling approach is more general. Flow (and any other quantity such as sediment transport) should satisfy continuity conditions at the interfaces between 1D, 2D, and 3D sub-domains, as discussed below.

2.7.1 Water Level

The water levels at the interfaces should satisfy (Wu W., 2007):

$$B \cdot z_{s,1\mathrm{d}} = \int_0^B z_{s,2\mathrm{d}} \, dy = \int_0^B z_{s,3\mathrm{d}} \, dy \tag{2.76}$$

where $z_{s,1\mathrm{d}}$, $z_{s,2\mathrm{d}}$, and $z_{s,3\mathrm{d}}$ are the water levels calculated in the 1D, 2D, and 3D sub-domains, respectively; B is the width of the interface; and y is the coordinate along the interface.

For a 3D model that solves the full Navier-Stokes equations, the dynamic pressure needs to be provided at the interfaces, whereas the 1D and depth-averaged 2D models assume a hydrostatic pressure distribution there. To overcome this problem, the interfaces should be located in the regions where the flow varies gradually and the hydrostatic pressure assumption is valid (Wu W., 2007).

When only 2D/3D integration is considered, the connection condition may be addressed more precisely. If the two meshes are the same at the interface (all of the 3D nodes on the interface coincides with the 2D nodes and vice versa) it is sufficient to satisfy the following relationship at each interface node:

$$z_{s,2\mathrm{d}} = z_{s,3\mathrm{d}} \tag{2.77}$$

If the 3D mesh is denser at the interface than the 2D one, the quantity B in equation (2.76) may be the distance between each two neighbouring nodes of the 2D mesh on the interface.

2.7.2 Flow Discharge

The flow discharges at the interfaces should satisfy the continuity condition (Wu W., 2007):

$$Q_{1\mathrm{d}} = \int_0^B U_{2\mathrm{d}} D_{2\mathrm{d}} \, dy = \int_0^B \int_{z_b}^{z_s} u_{3\mathrm{d}} \, dz \, dy \tag{2.78}$$

where $Q_{1\mathrm{d}}$ is the flow discharge calculated in the 1D sub-domain, $D_{2\mathrm{d}}$ and $U_{2\mathrm{d}}$ are the flow depth and depth-averaged velocity in the 2D sub-domain, $u_{3\mathrm{d}}$ is the local velocity in the 3D sub-domain, and z is the vertical coordinate.

When only 2D/3D integration is considered, the connection condition may be addressed more precisely. If the two meshes are the same at the interface (all of the 3D nodes on the interface coincides with the 2D nodes and vice versa) it is sufficient to satisfy the following relationship at each interface node:

$$U_{2\mathrm{d}} D_{2\mathrm{d}} = \int_{z_b}^{z_s} u_{3\mathrm{d}} \, dz \tag{2.79}$$

If the 3D mesh is denser at the interface than the 2D one, the quantity B in equation (2.78) may be the distance between each two neighbouring nodes of the 2D mesh on the interface.

2.7.3 Flow Resistance

The bed shear stresses at the interfaces satisfy (Wu W., 2007):

$$l_{wet}\cdot\tau_{b,1\mathrm{d}}=\int_0^{l_{wet}}\tau_{b,2\mathrm{d}}\,dl=\int_0^{l_{wet}}\tau_{b,3\mathrm{d}}\,dl \tag{2.80}$$

where $\tau_{b,1\mathrm{d}}$, $\tau_{b,2\mathrm{d}}$, and $\tau_{b,3\mathrm{d}}$ are the bed shear stresses calculated in the 1D, 2D, and 3D sub-domains, respectively; and l_{wet} is the wetted perimeter at the interfaces.

It must be noted that using similar Manning values in 1D and 2D models does not mean that the bed roughness in the two models are equivalent. In order to satisfy equation (2.80) at the interface between 1D and 2D reaches, $n_{1\mathrm{d}}$ and $n_{2\mathrm{d}}$ may have to be given different values. Inserting the Manning equations for 1D and 2D uniform flows into equation (2.80) leads to (Wu W., 2007):

$$n^2_{1\mathrm{d}}\cdot U^2_{1\mathrm{d}}\cdot\frac{A_{1\mathrm{d}}}{R^{4/3}_{1\mathrm{d}}}=\int_0^{l_{wet}}\frac{n^2_{2\mathrm{d}}\cdot U^2_{2\mathrm{d}}}{D^{1/3}_{2\mathrm{d}}}\,dl \tag{2.81}$$

where $R_{1\mathrm{d}}$ and $U_{1\mathrm{d}}$ are the hydraulic radius and the section-averaged flow velocity calculated in the 1D model respectively; and $n_{1\mathrm{d}}$ and $n_{2\mathrm{d}}$ are the Manning coefficients in the 1D and 2D models, respectively.

The bed shear stress must be identical at the interface of 2DH/3D model integration. The bed shear stress is usually determined using the wall-function approach in the 3D models, which links the bed shear stress to the near-bed velocity. On the other hand the bed shear stress in the 2D models (in the horizontal plane) is usually linked to the depth averaged velocity. Thus, equation (2.80) may not be satisfied at the interface between 2D and 3D models, without some corrections.

The conservation lows have to be satisfied for any other quantity too, which is simulated in addition to the flow itself. Several quantities may be named in this regard such as the bed load, suspended sediment load, bed level change, salinity, temperature, and water quality parameters.

2.8 Summary

The theoretical background of hydrodynamic models has been presented in this chapter. The mass and momentum conservation lows have been described in this regard with attention to the body forces and boundary conditions. It has been stressed that the Coriolis effect has a vertical component which is not negligible in meso-scale oceanic flows and that the famous form of the

Coriolis terms, which have been used in many numerical models, is acceptable only if the x- and y-direction of the coordinate system are in the east- and north-ward directions.

There are always some fluctuations in the flow properties that are not important for the engineers. The averaging techniques to filter-out such fluctuations and the averaging effect on the N.S. Equations has been presented in this section. The physical behaviour of turbulence, as continues instability, and its modelling approaches has been illustrated in this regard. It helps adequate turbulence model selection for each flow case. The broad range and continuous spectrum of length and time scales of the eddies has been considered into account with the importance of large and small eddies in momentum exchange and energy dissipation respectively.

Several Reynolds averaging (temporal averaging) models has been illustrated including zero equation models, one equation models, two-equation models and so on. The k-ε models are the most common two-equation models. A wide variety of k-ε models exists. The most common versions are the standard and the RNG k-ε models which have been described in more detail. Large eddy turbulence modelling has also been illustrated, which is some kind of spatial averaging. The idea behind this approach is that large scale eddies are by far the most effective transporters of the conserved properties (such as momentum) in comparison with the small scale eddies.

The water density gradient affects not only the flow field but also the turbulence intensity. Turbidity current is a specific type of density gradient flow which has been illustrated in the present section. When the heavier water is above the clear water, the situation is unstable which leads to increase in turbulence. When the heavier water is below the clear water, the stratification is stable which leads to decrease in turbulence. The effect of vertical density gradient, the buoyancy effect, on the turbulence modelling has been addressed in detail.

The vertical scale of flow field in hydro-environmental engineering problems is generally some order of magnitudes smaller than the horizontal scale and there are two distinguished turbulence structures; a) the small nearly-isotropic 3D eddies produced by the bottom friction and b) the large horizontal 2D eddies much larger than the water depth. Therefore two eddy viscosities are needed in the N.S. Equations; for the vertical eddies and for the horizontal ones. Each viscosity needs its own turbulence model. General k-ε turbulence models are not suitable in this regard. Instead, specific versions of the k- ε models are used with modified production terms. The difference has been illustrated in this chapter.

The shallow water equations and the connection conditions for 2D/3D model integration have been addressed in the the rest of this chapter.

3 Review on Solution Strategies for 3D Free Surface Flow

3.1 Introduction

The complexity of 3D numerical solution for the free surface flow, in comparison with 2D horizontal modelling, is more than just an extension to a third dimension or by taking another equation into account. The pressure treatment and water surface variability are the main issues in this regard. The pressure gradient is one of the main driving forces in the momentum equations. However, there is not a direct relationship for representing the pressure field. Determining the free surface is the other issue. The free surface position is needed for the solution procedure while it is variable and must be determined during the solution. The variability of the free surface also leads in a variable solution domain. The necessity of dealing with very large number of unknowns is an extra difficulty.

This chapter deals with the above mentioned problems. The pressure treatment issue is addressed in section (3.2). The problem is described first and the solution approaches are reviewed. The most popular methods are described with more details, which include the hydrostatic pressure assumption, the artificial compressibility method, the SIMPLE family of methods, the PISO method and the projection method. The moving upper boundary is addressed in section (3.3). It starts with the problem description and continues with illustrations of the most popular solution approaches. The marker and cell (MAC) method, volume of fluid (VOF) method, level-set formulation methods, kinematic boundary condition method, depth integrated continuity equation method and the shallow water approach are described. The sigma-coordinate and its vertical gradient adaptive version are reviewed in section (3.3.3) to address the variability of the solution domain. Section (3.4) describes the Poisson equation used for computing the pressure and the associated solution methods.

Numerical solution of the pressure equation is part of this study because the projection method is used. Because the resulting system of equations is very large in a real 3D problem, special matrix solvers are needed in this regard. Several algorithms have been listed in section (3.4.2) and some of them are reviewed in the remaining sections. They are the conjugate gradient method, preconditioned conjugate gradient method and multigrid method. Matrix solvers are needed not only for solving the Poisson pressure equation, but also for some other parts of the numerical model. For example the preconditioned bi-conjugate gradient stabilised (BI-CGSTAB) method (van der Vorst, 1992) has been used in the present study for computing the free surface implicitly.

3.2 Pressure-Velocity Relation

3.2.1 The Problem and Solution Ideas

Pressure gradient has an important role on the flow field via the momentum equations. In fact, it is the main momentum source term in most flows of engineering importance (Versteeg and Malalasekera, 1995). In compressible flows, the continuity equation can be used to determine the density. Then the pressure is calculated from the equation of state. This approach is not appropriate for incompressible or low Mach number flows (Ferziger and Peric, 2002). With incompressible flows, there is not any independent equation to solve the pressure field, in spite of pressure gradient's contribution and importance in the momentum equations as driving force. This is a problem for 3D modelling of incompressible flows.

Several approaches have been proposed by researchers to deal with this problem. They can be categorized into three classes:

- Pressure eliminating methods
- Artificial compressibility methods
- Pressure-velocity decoupling methods

The first class comprises methods which eliminate the pressure variable from the momentum equations. The most common method of this kind is the hydrostatic assumption whereby the pressure is replaced by the water surface elevation. This technique is useful where the vertical acceleration component is small enough to enable the hydrodynamic part of the pressure field to be neglected. The second class comprises methods that change the continuity equation in order to include an artificial compressibility term.

The third class comprises methods which decouple pressure from velocity. They differ fundamentally from the first class by retaining pressure as one of the prognostic variables. There are two main sub-classes of this type, including pressure-correction methods and projection methods. There are different pressure correction-methods: SIMPLE-family (SIMPLE, SIMPLER and SIMPLEC) and PISO methods. The projection methods include a large family of solution procedures.

There are also some other techniques that are not widely used in 3D solutions, such as solving non-primitive variables such as velocity-vorticity. The definition of vorticity is suitable for 2D-vertical problems, but rarely used in three dimensions. The reason is that with 2DV problems one vorticity variable is sufficient, while for 3D problems three vorticity are required (Mike3 manual).

A brief description for the above mentioned approaches is given in the following sections.

3.2.2 Hydrostatic Pressure Assumption

This is a pressure eliminating method, which eliminates the pressure term from the momentum equations. This approach neglects the non-hydrostatic part of the pressure field. The pressure term is replaced by the water surface elevation. The hydrostatic part of the pressure field at elevation z is given as:

$$p_z = p_a + g\int_z^\eta \rho . dz \qquad (3.1)$$

The momentum equations need only the horizontal gradient of the pressure field. There is no need for knowing the pressure itself. Hence it is possible to eliminate the atmospheric pressure (pa) from the equations to avoid round-off error problems due to cancelling large magnitude terms. For constant water density equation (3.1) becomes:

$$p_z = \rho g(\eta - z) \qquad (3.2)$$

The hydrostatic pressure distribution assumption is useful in cases where the vertical acceleration of velocity is negligible. It can be used in many practical engineering problems.

The atmospheric pressure terms must not be cancelled-out for problems with large solution domains, because the horizontal gradient in the atmospheric pressure may be effective in sudden storm conditions.

3.2.3 Artificial Compressibility

With the artificial compressibility method (ACM), the continuity equation is modified to include an artificial compressibility term which vanishes when the steady state is reached. The idea is to change the mathematical characteristics of the incompressible equations to make them similar to the compressible ones. The compressible flow equations are hyperbolic which means that they have real characteristic lines on which signals travel at finite propagation speeds[1]. By contrast, the incompressible equations have a mixed parabolic-elliptic characteristic (Chorin, 1967; Ferziger and Peric, 2002).

In free surface flows only the slow processes are of interest and usually the fast processes (such as the shock waves) have no substantial influence on the slow processes (such as the free surface waves). Hence they may be removed without significantly loss of information. The fast processes are easily eliminated by replacing the time derivative of the density in the mass conservation equation with the pressure term in the equation of state, whereby a compressibility of the fluid is introduced. The fast processes are then subsequently eliminated through an artificial compressibility, whereby the system mathematically has become hyperbolically dominated. The continuity equation becomes:

$$\frac{1}{\beta_c}\frac{\partial p}{\partial t}+\frac{\partial\left(\rho u_i\right)}{\partial x_i}=0 \tag{3.3}$$

where β_c is an artificial compressibility parameter[2].

The artificial compressibility is an excellent tool to enable faster convergence of model solution in steady state situations. However it is not obvious that the same technique can also be applied accurately for unsteady incompressible flows. The addition of a time derivative of the pressure to the continuity equation means that we are no longer solving the true incompressible equations. Thus the time history generated may not be accurate. In spite of that, this method has been used successfully for simulating unsteady free surface cases too (for example Mike3). It seems that for free surface flows, accumulated pressure can be released during each time step via adjusting the surface elevation. If no free surface is present, the artificial compressibility method cannot be used for unsteady applications unless with very small time steps (Ferziger and Peric, 2002; Mike3 manual).

The compressibility value must not become smaller than the celerity of the free surface waves. If it is too high the system becomes stiff (the pressure information travels fast). If it is too low, the flow does not propagate with the correct physical celerity. The appropriate range for the compressibility is a function of grid spacing, time step and the maximum water depth. It must be low enough to overcome stiffness associated with a disparity in the magnitudes of the eigenvalues, but high enough such that pressure waves (moving with infinite speed at incompressible limit) be allowed to travel far enough to balance viscous effects (Chung, 2002).

[1] This reflects the ability of compressible fluids to support sound waves.

[2] In fact, $\sqrt{\beta_c}$ is the artificial speed of sound

There are different methods that can be used to solve compressible flow equations. Most of them can also be used for solving artificially compressible equations.

3.2.4 SIMPLE-Family Methods

The SIMPLE family of methods employ iterative solution strategies to solve the pressure-velocity issue with incompressible problems. There are three main methods in this category: The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm of Patankar and Spalding (1972), the SIMPLER (SIMPLE-Revised) algorithm of Patankar (1980), and the SIMPLEC (SIMPLE-Consistent) algorithm of Van Doormal and Raithby (1984). They are kinds of guess-and-correct procedures to calculate pressure and mostly deploy a staggered grid. The original SIMPLE method was restricted to steady state problems. Then its unsteady version was introduced.

The main procedure of SIMPLE method is as follows (Versteeg and Malalasekera, 1995; Olsen, 2004):

1- Guess the pressure and velocity fields,

2- Solve discretised momentum equations to define a new guess for velocity field

3- Solve a pressure correction equation to define pressure-correction values,

4- Correct pressure and velocity fields (under-relaxation is needed for stability),

5- Check for convergence, if not gained yet, use the latest values as new guess and go back to step 1.

In step 4 an additional equation is needed for velocity correction. It comes from the momentum equation. Discretised momentum equation links velocity correction at each point to pressure and velocity corrections around it. The velocity correction values at the neighbouring points are not known; therefore in SIMPLE method it is assumed that velocity corrections are negligible at this stage.

Similarly in step 3 an additional equation is needed for pressure correction. It comes from the continuity equation. If the corrected velocities (using the simplification described) are inserted into the continuity equation, the only remaining unknowns in the resulting equation will be the pressure correction at the neighbouring points (it is an implicit equation). This equation involves the guessed values of velocity at the neighbouring points as source terms.

The pressure-correction equation is satisfactory for correcting velocities, but it is not so useful for correcting the pressure field (because of the simplifications made). The improved procedure SIMPLER is introduced to overcome this shortcoming. The SIMPLER method uses two different pressure correction equations. One of them, which is the same as for the SIMPLE method, is used just to obtain velocity corrections. A separate, more effective, pressure equation is solved to yield the correct pressure field. No terms are omitted in deriving this pressure equation. This one uses the so-called pseudo-velocities which are again calculated on the base of guessed velocity values, but via the complete momentum equations without neglecting any term.

The procedure of SIMPLER method is as follows (Versteeg and Malalasekera, 1995; Olsen, 2004):

1- Guess the pressure and velocity fields,

2- Calculate the so-called pseudo-velocities,

3- Solve the enhanced pressure correction equation

4- Calculate the corrected pressure field (under-relaxation needed for stability),

5- Set the calculated pressure field as the new guess,

6- Solve discretised momentum equations to have a better guess from velocity field,

7- Solve the simplified pressure correction equation (like SIMPLE) to have a more accurate pressure corrector,

8- Correct velocity field as in step 4 of the SIMPLE method,

9- Check for convergence, if not gained yet, use the latest pressure and velocities as the new guess and go back to step 1.

The SIMPLEC (SIMPLE-Consistent) algorithm of Van Doormal and Raithby (1984) follows the same steps as the SIMPLE algorithm, with the difference being that in deriving the velocity correction equations it omits some terms that are less significant than those omitted by the SIMPLE algorithm.

Under-relaxation is necessary at each stage of pressure or velocity correction due to stability problems. The selection of relaxation factors is dependent on the flow conditions. The values are between 0 and 1 (1 means no under-relaxation and 0 means no correction). A general guide is 0.2 for pressure and 0.8 for velocity.

The steady and unsteady formulations of simple family methods are not very different. The unsteady version has an additional transient term in each discretised momentum equation and also an additional term due to time derivative as source terms in the pressure correction equations. For unsteady problems, iterations are needed at each time step.

The iterative methods used in SIMPLER or SIMPLEC method have robust convergence characteristics in strongly coupled problems (Versteeg and Malalasekera, 1995). The SIMPLE method and its extensions are used in most CFD calculations in the world (Olsen, 2000). More details can be found in text books like Chung, 2002; Olsen, 2004; Versteeg and Malalasekera, 1995 and many others.

3.2.5 PISO Method

The PISO algorithm (Pressure Implicit with Splitting of Operators) of Issa (1986) is a pressure-velocity calculation procedure developed originally for the non-iterative computation of unsteady compressible flows. It obtains solutions without iterations, and with larger time steps and less computing effort (Versteeg and Malalasekera, 1995; Chung, 2002).

In this scheme, the conservation of mass is designed to be satisfied within the predictor-corrector steps. It involves one prediction step and two correction steps. The procedure is as follows (Versteeg and Malalasekera, 1995):

1- Perform steps 1 to 4 of SIMPLE algorithm. It solves the discretised momentum equations and pressure correction equations and then corrects pressure and velocities,

2- Solve the second pressure correction equation,

3- Correct pressure and velocities,

4- Check convergence, if not gained yet, use the latest values as the new guess and go back to step 1

The second pressure correction equation in PISO method (step 2) is different from SIMPLE family correspondents. It contains the difference between the two consecutive guessed velocity fields as source terms. The pressure and velocity correctors (step 3) are also different. They contain two successive pressure correction values and two successive guessed velocity fields.

Issa (1986) showed that the temporal accuracy achieved by the predictor-corrector process is of order 3 (Δt^3) for pressure and 4 (Δt^4) for momentum. Therefore, the pressure and velocity fields obtained at the end of the PISO process with a suitably small time step are considered to be accurate enough to proceed to the next time step immediately and the algorithm is non-iterative. The PISO method may be seen as an extension of SIMPLE, with a further corrector step to enhance it. Still it is very similar to the fractional-step method (Versteeg and Malalasekera, 1995; Ferziger and Peric, 2002). More details can be found in text books like Chung, 2002; Olsen, 2004; Versteeg and Malalasekera, 1995 among many others.

3.2.6 Projection Method

The projection method (first introduced by Chorin, 1968) uses operator splitting in order to decouple velocity and pressure computations. It is based on a projecting operator which projects a vector field onto the space of divergence-free field with appropriate boundary conditions. It is a time splitting method in which the momentum equations are solved without the pressure term for the first stage, then the continuity and the remaining part of the momentum equation are solved in the second stage. This method is based on the fractional steps algorithm, which has been used widely in the present study for 2D/3D modelling and for integration of the models.

The main idea of projection method is that any vector field can be uniquely decomposed into divergence-free and rotation-free components where the rotation-free portion may be denoted by the gradient of a potential:

$$\vec{u}=\vec{u}_d+\nabla\phi \tag{3.4}$$

where $\vec{u}_d$ is divergence-free ($\nabla\cdot\vec{u}_d=0$) and $\nabla\phi$ is rotation-free (ϕ is a scalar function). This approach may be used for the hydrodynamic velocity field with the real fluid velocity as the divergence-free vector field and the pressure as the scalar function. One can start with estimating an intermediate velocity field by neglecting the pressure gradient terms in the N.S. equations. This intermediate velocity field is not divergence-free. However, it can be decomposed later into divergence-free (vector) and rotation-free (scalar) components in such a way that the scalar component is the pressure gradient term. Therefore, the divergence-free component will be the solution of the complete N.S. equations.

The N.S. equations may be written in the following way:

$$\frac{\partial\vec{u}}{\partial t}+\frac{1}{\rho}\nabla p=S(\vec{u})=g+\nu\nabla^2\vec{u}-\vec{u}\cdot\nabla\vec{u} \tag{3.5}$$

where the term $\partial\vec{u}/\partial t$ is divergence free because of incompressibility. The term ∇p is rotation-free. $S(\vec{u})$ is generally neither divergence-free nor rotation-free. When $\vec{u}$ is given, $S(\vec{u})$ can be projected into divergence-free and rotation-free components.

Equation (3.5) may be fractioned as follows:

$$\frac{\vec{u}_{est}-\vec{u}^n}{\Delta t}=S(\vec{u}) \tag{3.6}$$

$$\frac{\vec{u}^{n+1}-\vec{u}_{est}}{\Delta t}=-\frac{1}{\rho}\nabla p \tag{3.7}$$

In order to define the intermediate velocity field, equation (3.6) may be solved without knowing the pressure gradient. Equation (3.7) includes two unknowns: ∇p and $\vec{u}^{n+1}$. However, it is known that $\vec{u}^{n+1}$ is divergence-free. Therefore its divergence is zero. Applying the divergence operator on equation (3.7) leads to:

$$\nabla^2 p=\frac{\rho}{\Delta t}(\nabla\vec{u}_{est}-\overbrace{\nabla\vec{u}^{n+1}}^{0})=\frac{\rho}{\Delta t}\nabla\vec{u}_{est} \tag{3.8}$$

Equation (3.7) is the Poisson equation for pressure. It need be solved in order to define the pressure (and pressure gradient) field. Then equation (3.8) may be solved to define $\vec{u}^{n+1}$.

The major steps of the projection method are as follows (two steps):

1- Intermediate velocity step

The estimated velocity field is computed using an approximate version of the momentum equation. This version of momentum equation does not include unknown pressure values. The resulting intermediate velocity is not divergence-free.

2- Projection step

2-a- Solve the pressure Poisson equation (3.8).

2-b- Determine the projected velocity, i.e. the final one:

$$\vec{u}_d = \vec{u}_{est} - \nabla p \tag{3.9}$$

There are two versions of the projection method. In the original version (projection-I method) the intermediate velocity $\vec{u}_{est}$ is computed using the momentum equations without taking the pressure gradients into account. The scalar function in this case is the physical global pressure. Equation (3.5) follows this approach. In the second version (projection-II method) the influence of the pressure gradients is not neglected completely in estimating the intermediate velocity. Instead, the pressure gradient from the previous time step ∇p^n is applied in the intermediate velocity step. In the projection-I method, the total pressure is computed in the projection step (step 2-a) while a pressure correction is computed in the projection step of the projection-II method.

When the projection method is applied for predicting free surface flows, there is an opportunity to use the hydrostatic part of pressure in the momentum equation in the first step in order to define the intermediate velocity. The hydrostatic pressure can be easily found. Therefore the pressure may be decomposed into a sum of the hydrostatic pressure and hydrodynamic pressure. Using this idea, the governing equations will be split in such a way that the hydrodynamic pressure is treated as a correction to the hydrostatic component. It is also possible to apply in the first stag not only the hydrostatic pressure gradients, but also the gradients of the hydrodynamic pressure from the previous time step. This approach is suitable especially for situations where the hydrodynamic pressure is relatively large in comparison to the hydrostatic one.

3.3 Moving Free Surface Boundary

3.3.1 The Problem

For most of the hydro-environmental problems, the upper boundary of water domain is free to move. In general, the position of the free surface is neither steady nor known a priori and has to be determined during the computation. This fact leads to some difficulties in 3D and 2D-vertical free surface flow problems, where the governing equations must be discretised in the vertical direction too.

Specially, there are two specific difficulties. The first one is that one of the domain boundaries is unknown and must be solved as part of the solution, in spite of that the solution procedure itself needs to know the boundary position. Upper boundary conditions can be applied only when the free surface position is known. The barotropic pressure gradients caused by free surface slopes act as driving forces on the flow (especially in shallow waters). It makes the exact reproduction of free surface movements a crucial point in the modelling (Jankowsky, 1999).

The second problem is that a variable domain must be discretised, gridded and handled through the numerical solution. Handling unsteady grid or unsteady boundary positions has its own difficulties. The variable free surface not only defines the new computational geometry, but also influences the velocity and pressure fields through the boundary conditions. The kinematic and dynamic boundary conditions at the free surface have been discussed in chapter (2). The implementation of free surface boundary conditions is not as trivial as it would appear. If the position of the free surface were known, there would be little problem remains. The problem is that the location of the free surface must be computed as part of the solution and is not usually known in advance (Ferziger and Peric, 2002).

It must be noted that sometimes considerable non-linearity and instability problems can arise because of the effect of unsteady free surface on solution procedure. Therefore, it is a good idea to use the implicit values (the values of the new time step) of water surface elevation in momentum equations[1]. This is the reason why in many numerical models the water surface is calculated before momentum equations (or implicitly with them) at each time step.

The problems described above do not exist with 1D and 2D horizontal problems, because they do not disretize the vertical dimension. For them, the position of water surface is one of the normal unknowns in the horizontal mesh to be solved.

3.3.2 Allocating the Free Surface Position

There are several algorithms that can be used to allocate water surface in hydro-environmental problems with vertical discretisation (3D and 2DV). Some of them use boundary fitted grids and some other use a fixed grid, which extends beyond the free surface.

The most common algorithms are as follows:

- Marker and cell (MAC) method,

[1] Even if the water surface elevation, itself, has been calculated explicitly.

- Volume of fluid (VOF) method,
- Level-set formulation method,
- Solving the kinematic boundary condition,
- Using depth integrated continuity equation, and
- Solving the shallow water equations.

Below is a brief review of each algorithm.

3.3.2.1 Marker and Cell (MAC) Method

The idea of Marker-and-Cell (or Marker-in-Cell) scheme is to introduce mass-less particles at the free surface or in the whole fluid volume at the initial time and following their motion in a fixed grid, which extends beyond the free surface. The shape of the free surface is determined by computing the fraction of each near-interface cell that is partially filled. This approach has been introduced by Harlow and Welch (1965). It can treat relatively complex situations. However, the computing effort is large, especially in three dimensions. In addition to solving the equations governing fluid flow, one has to follow the motion of a large number of marker particles (Ferziger and Peric, 2002). The MAC algorithm is a Lagrangian method. It is necessary to redistribute the particles evenly in all fluid cells whenever they lead to spread disproportionately (Jankowski, 1999).

A great advantage of MAC method is the ability to handle complicated, general and arbitrary free surface problems: breaking surfaces; dam breaking or splash of a falling column of water; fluid detachment or coalescence (droplets); or to simulate filling complex moulding shapes (Jankowski, 1999). However, it is a method for small scale problems. For large scale problems, such as common hydro-environmental problems, the resultant limitation in time step becomes a problem (Ferziger and Peric, 2002). A stability restriction, relating the time step to the spatial discretisation and to the free surface wave speed, inhibits this method from being applied to three-dimensional geophysical flows with a sufficiently fine grid to resolve the small scale non-hydrostatic component of the flow (Casulli and Stelling, 1998). In addition, the required number of particles is too much to be reasonable for large scale problems. Figure (3.1b) illustrates the approach of MAC method.

3.3.2.2 Volume of Fluid (VOF) Method

With the VOF method, in addition to the conservation equations for mass and momentum, one solves an extra equation for the filled fraction of each control volume. The shape of the free surface is determined by checking the fraction of each near-interface cell that is partially filled. This approach has been introduced by Hirt and Nichols (1981). The computation is performed on a fixed grid, which extends beyond the free surface. This method is more efficient than the MAC scheme and may be applied to complex free surface shapes including breaking waves. However, the free surface profile is not sharply defined; it is usually smeared over one to three cells. Local grid refinement is necessary for accurate resolution of the free surface (Ferziger and Peric, 2002).

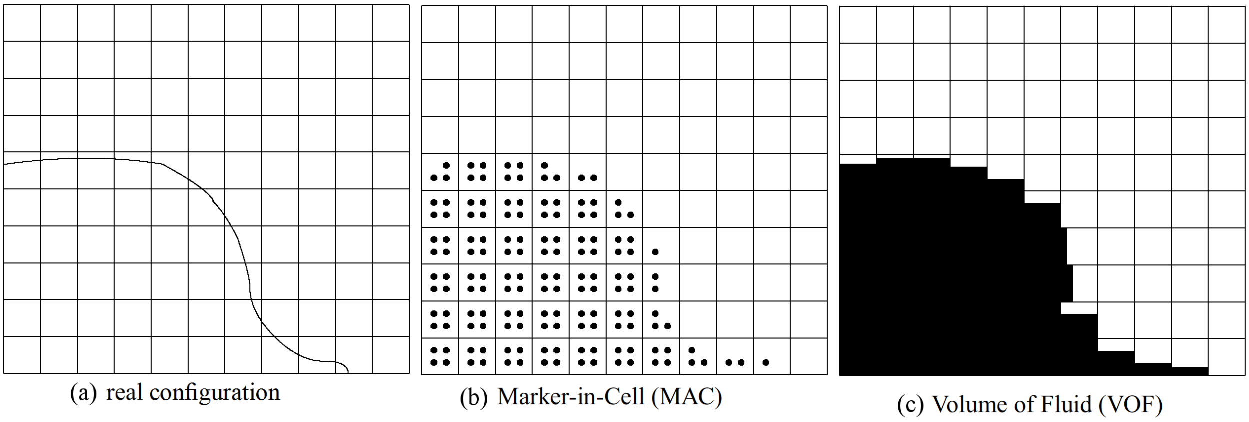
From the continuity equation, one can show that the evolution of the filled fraction of a cell is governed by the transport equation:

$$\frac{\partial c_{cell}}{\partial t}+\operatorname{div}(c_{cell}\vec{u})=0 \tag{3.10}$$

where c_{cell} is the filled fraction of each cell and $\vec{u}$ is the velocity field.

The VOF algorithm is an Eulerian approach. The critical issue in this type of method is the discretisation of the convective term in equation (3.10). Low-order schemes smear the interface and introduce artificial mixing of water and air. Therefore higher-order schemes are preferred. The VOF method is very attractive for small scale problems, but it enforces too small time step limitations in hydro-environmental engineering problems (Ferziger and Peric, 2002).

Figure (3.1c) illustrates the approach of VOF method. There are several variants of the VOF method. The water surface line inside each cell is not simple to specify for all of them.


(a) real configuration

(b) Marker-in-Cell (MAC)

(c) Volume of Fluid (VOF)

Figure 3.1. MAC and VOF methods (Jankowski, 1999)

3.3.2.3 Level-Set Formulation Method

Another class of interface-capturing methods is based on the level-set formulation, introduced by Osher and Sethian (1988). The surface is defined as the one, on which a level-set function equals to zero. Other values of this function have no significance. To make it a smooth function, it is typically initialized as the signed distance from the interface. This function is then allowed to evolve as a solution of the transport equation (Ferziger and Peric, 2002):

$$\frac{\partial \phi}{\partial t}+\operatorname{div}(\phi \vec{u})=0 \tag{3.11}$$

where ϕ is the mentioned function and $\vec{u}$ is the fluid velocity field.

The advantage of this approach relative to the VOF method is that ϕ varies smoothly across the interface while the volume fraction c is discontinuous there. Since ϕ does not explicitly occur in any of the conservation equations, the original level-set method did not exactly conserve mass. There are special treatment methods to enforce mass conservative (see Zhang et al., 1998) (Ferziger and Peric, 2002).

3.3.2.4 Solving the Kinematic Boundary Condition

It is possible to use the kinematic boundary condition at the free surface to trace the water surface level. The condition is as follows:

$$\frac{\partial \eta}{\partial t}+u_s\frac{\partial \eta}{\partial x}+v_s\frac{\partial \eta}{\partial y}-w_s=0 \tag{3.12}$$

The suffix s indicates the velocity components at the free surface.

This approach needs the surface velocity components to be known. It may be obtained by extrapolation from the interior or by using the dynamic free surface boundary condition. The main limitation of this method, like any other surface-tracking algorithm, is due to the mathematical requirement for the height function; it must be a single-valued function. These kind of methods cannot simulate complex cases such as wave breaking. Equation (3.12) is a pure advection equation. Any suitable advection solver can be used for it. For example the method of characteristics can be used.

Equation (3.12) does not include any dissipative terms which may make the solution smooth. Therefore the solution is over-sensitive to small errors or imperfectly set boundary conditions. To overcome this problem, stabilising techniques for the free surface are needed[1] (Jankowski, 1999).

3.3.2.5 Using Depth Integrated Continuity Equation

The most common approach for defining the elevation of water surface in hydro-environmental modelling is to solve the depth integrated form of continuity equation. The kinematic boundary conditions at the water surface and bed are involved in this manner. This

[1] For example some kind of filtering method.

procedure leads to an equation which includes the surface position and the gradient of horizontal velocities integral on the water column, giving as:

$$\frac{\partial \eta}{\partial t} = \int_{z_b}^{\eta} \left(\frac{\partial u}{\partial x} \right) dz + \int_{z_b}^{\eta} \left(\frac{\partial v}{\partial y} \right) dz = \frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} u.dz \right) + \frac{\partial}{\partial y} \left(\int_{z_b}^{\eta} v.dz \right) \tag{3.13}$$

This approach offers a simple and robust method for simulating free surface flows. It restricts to a single valued function for the water level. Thus a wide class of flows, e.g. breaking surfaces, bubbles, and drops cannot be simulated via this method.

Equation (3.13) is a conservative form of the free surface equation (the kinematic BC). It brings a method of finding the free surface while automatically satisfying the mass conservation criterion (Jankowski, 1999).

There are several ways to solve equation (3.13), including explicit, semi-implicit and implicit[1]. Following is a brief description of these methods.

a) The simplest way is an explicit solution, which is straightforward, but restricts the maximum value of time step.

b) It is also possible to solve equation (3.13) semi-implicitly. This method is based on computing [u_k] and [v_k] velocity vectors (for water column) from x- and y-momentum equations, then inserting them into the discretised form of equation (3.13). The momentum equations are discretised in such a way that only water level (η) appears implicitly (of time step n+1) at the left hand side of the resulting equation. All other terms are explicit (from time step n) and known. They are advection terms, hydrodynamic component of pressure, etc. For this to be achieved, the following conditions need to be satisfied:

1- The momentum equations must be written in the vector form (for each water column),

2- The water level gradient[2] and all vertical gradient terms must be discretised implicitly, and

3- All the other terms must be discretised explicitly.

The resulting equation for water level is solved implicitly. The semi implicit method mentioned above has been introduced by Stansby (1997) and Stansby and Zhou (1998) for 2DV modelling. They used a tri-diagonal solver for the resulting implicit equation. Casulli and Stelling (1998) used the same method for 3D modelling. They used a preconditioned conjugate gradient method for solving the resulting implicit equation.

[1] Anyhow, the values of η and vertical position of the σ-coordinate mesh points is taken from the previous time step.

[2] What is doing this term (water level gradient) in a 3D x- or y-momentum equation? It is there because the whole pressure has been split into hydrostatic and hydrodynamic parts. The hydrostatic part has been written as a function of distance from the water level. Thus the horizontal gradient of the whole pressure produces horizontal gradient of water level.

c) It is also possible to define water level implicitly. This approach is based on the fractional step method to simplify momentum equations before combining them with equation (3.13). There are also some other interesting algorithms, which use equation (3.13) among others to eliminate velocity or pressure and derive a system of equations purely for pressure or velocity. They called pressure-elimination or velocity-elimination methods. The unknowns in the resulting equations are not scalar values. They are vectors of pressures or horizontal velocities. Each vector involves the pressure or velocity values for all the computational cells in a water column. The resulting system of equations is in a block tri-diagonal form for the 2DV problems (Namin, 2003). For more information, see Namin et al. (2001) and Namin (2003).

3.3.2.6 Solving Shallow Water Equations

Another way to define water surface, is to solve the 2D shallow water equations at the start of each time step. This approach is simple and suitable for those 3D (or 2DV) models that assume the pressure field to be hydrostatic. For each shallow water simulation, the 3D velocity components are integrated numerically along the water column to define depth averaged velocities at the recent time step. There are many efficient numerical methods for modelling shallow water flows, which can be used in this regard. It is not necessary for the 2D model to have the same time step, as the 3D model. This approach is not suitable for complex flow problems.

3.3.3 Handling Variability of Solution Domain

Three main approaches are used to deal with the problem of variable solution domain:

- Eulerian approach,
- Lagrangian approach,
- Arbitrary Lagrangian-Eulerian approach (ALE).

With Eulerian approach, the numerical model grids are fixed and steady, except the top layer. The thickness of a cell at the top layer is generally variable in time and space in order to define the unsteadiness of the free surface. Some numerical models (for example MIKE3-Structured) use the Eulerian approach. In places where water level variation is considerably large relative to the water depth, this approach does not produce accurate vertical velocity profiles. This is especially true for the situation when wind induced currents are not negligible. Shallow coastal bays (or channels) at windy areas with high tidal variations are beyond the scope of such models.

With the Lagrangian approach, the computational points move with the flow. The numerical formulations of governing equations thus become different from what was described in chapter (2). Specifically, the advection terms disappear.

A pure Lagrangian approach is an attractive method for particle tracking within a defined velocity field. However, it may lead to distorted grids when it is used for solving the Navier Stokes equations. Because it is the velocity field that controls the grid shape.

The arbitrary Lagrangian-Eulerian (ALE) approach is the method of choice for most hydro-environmental free surface models. With the ALE, the numerical mesh moves with an arbitrary velocity which is different from the fluid velocity. The equations are formulated and solved in a transformed co-ordinate system. The mesh must be adapted to the new domain geometry at each new time step. There are several kinds of ALE methods for hydro-environmental free surface flow problems. Most of them restrict grid movement to the vertical direction. They track the spatial variations of bed and water surface and adapt themselves to free surface variations (and gradual variations in bottom topography). In this way, the solution domain remains constant and rectangular while the real physical domain is variable. The ALE method for vertical grid may be used jointly with either a structured or an unstructured horizontal mesh. There are different kinds of vertical moving ALE systems. With geophysical flows, the main categories are:

- Sigma (σ) coordinate grid system,
- Non-uniform extensions of σ system (for example Gamma grid system),
- Gradient-Adaptive-Sigma grid system.

The σ-coordinate is one of the most common coordinate systems for 3D and 2DV free surface flow problems. In this coordinate system, the vertical position of the computational nodes in real domain change (with the time-variable free surface) in such a way that the relative vertical distances between them remain constant. This method was first presented by Phillips (1957).

The σ-coordinate system is sufficient for many engineering problems. However, due to its uniform grid distribution, it has shortcomings when the vertical profile of the unknown quantity involves sharp changes. Two such examples are non-cohesive suspended sediment concentration and wind induced velocity profile. One solution is to use a vertical distributor function to refine the mesh at specific vertical positions. γ-coordinate system is such an example which has been developed to increase vertical resolution near the water surface (Huang and Spaulding, 1995).

Figure (3.2) illustrates σ- and γ-coordinate system of grids in comparison with some Eulerian meshes.

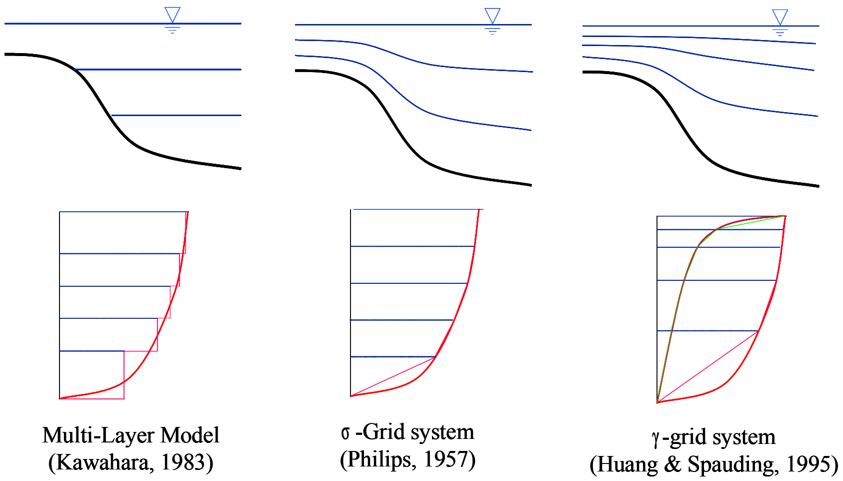


Figure 3.2. Stretches from σ-, γ- and Eulerian meshes (after Lu, 2003)

Using the vertical distributor function extends the applicability of σ-coordinate systems to more complex problems. But still the grid mapping is fixed in time and space. It means that the vertical position of sharp gradients must be known or to be estimated before a simulation starts. In addition, it must remain fixed during the solution and be the same everywhere. A more flexible approach is a vertical gradient adaptive σ-coordinate system (GAS), which has been proposed by Wai and Lu (1999).

The GAS method takes into account the gradient variation of a selected variable and uses it as an additional controlling factor in the σ-coordinate transformation. The transformed grid spacing automatically adjusts in time and space, according to the local solution gradient of the selected variable, and converges in the high gradient regions for better resolution. The transformation rules for GAS coordinate system are as follows (Wai and Lu, 1999):

$$\begin{aligned} &t_{GAS}=t \\ &x_{GAS}=x \\ &y_{GAS}=y \\ &z_{GAS}=\frac{(z-z_b)+\mu}{(\eta-z_b)+M} \qquad z_{GAS}\in[0,1] \end{aligned} \tag{3.14}$$

Where:

$$\mu=\int_{z_b}^{z}\left(\beta\left|\frac{\partial\lambda}{\partial z}\right|+\gamma\left|\frac{\partial^2\lambda}{\partial z^2}\right|\right)dz=f(t,x,y,z) \tag{3.15}$$

$$M=\int_{z_b}^{\eta}\left(\beta\left|\frac{\partial\lambda}{\partial z}\right|+\gamma\left|\frac{\partial^2\lambda}{\partial z^2}\right|\right)dz=f(t,x,y) \tag{3.16}$$

The transformation rules for the GAS system are similar to their correspondents for the σ-coordinate system, except the coefficients μ and M. The transformed form of the governing equations keeps the same as for σ transformation. The coefficients μ and M control the vertical distribution of the grid nodes. They are functions of the first and second vertical derivatives of an interested quantity, λ, which can be sediment concentration, velocity, temperature or anything else. Coefficients b and g are weighted parameters, which affect the grid distribution. The GAS transformation returns to σ transformation when these parameters are zero (Wai and Lu, 1999).

Figure (3.3) illustrates a snapshot of a GAS grid generated in an unsteady and complicated suspended sediment transport study, where two sediment sources exist (inflow and bed).

Any grid system can be used directly or as a transformation from the real domain into the solution domain. In this way, the governing equations are transformed from the real domain into a fixed solution domain before discretisation and numerical solution. By direct use, the governing equations are discretised and solved directly in the real domain. A combination of these to choices is also possible (for example see Stansby and Zhou, 1998).

Applying the σ-coordinate transformation to diffusion terms leads to lengthy and complicated expressions, while advection terms transform without overhead. The following relationship illustrate the transformation rules for the first and second gradients:

$$\frac{\partial}{\partial x}=\frac{\partial}{\partial\tilde{x}}+\frac{\partial\tilde{z}}{\partial x}\frac{\partial}{\partial\tilde{z}} \tag{3.17}$$

$$\frac{\partial^2}{\partial x^2}=\frac{\partial^2}{\partial\tilde{x}^2}+2\frac{\partial\tilde{z}}{\partial x}\frac{\partial^2}{\partial\tilde{x}\,\partial\tilde{z}}+\frac{\partial^2\tilde{z}}{\partial\tilde{x}\,\partial x}\frac{\partial}{\partial\tilde{z}}+\frac{\partial\tilde{z}}{\partial x}\frac{\partial^2\tilde{z}}{\partial\tilde{z}\,\partial x}\frac{\partial}{\partial\tilde{z}}+\left(\frac{\partial\tilde{z}}{\partial x}\right)^2\frac{\partial^2}{\partial\tilde{z}^2} \tag{3.18}$$

Equations (3.17) and (3.18) show that the transformed form of the first gradient includes only two first gradient terms and does not include cross-derivatives, while the transformed form of the second gradient includes several composite terms with complex cross-derivatives.

In several models the horizontal gradient of non-hydrostatic pressure and the horizontal diffusion are not transformed into the σ coordinate system. They are discretised in the Cartesian frame to avoid large errors, particularly near a steep bed. These errors result from the summation of large terms of opposite sign in the σ-coordinate transformation. The resulting truncation errors could cause spurious flows, particularly near the bed (Stansby and Zhou, 1998).

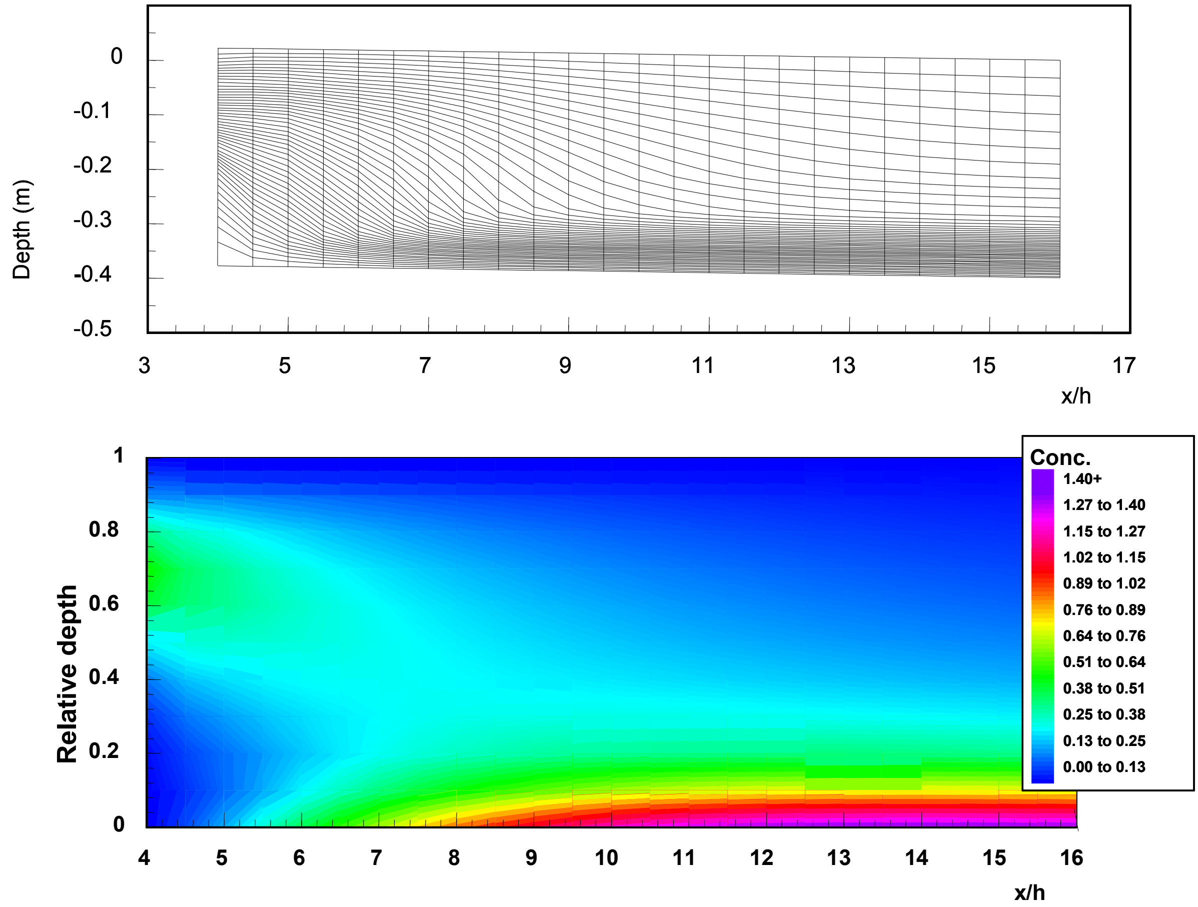


Figure 3.3. Example of GAS mesh (Wai and Lu, 1999)

One disadvantage of the σ-, γ- and GAS-mesh systems is that the number of vertical layers is the same for deep and shallow parts of the solution area. Thus the vertical layers near the bed may be less than necessary while at the shallowest parts, they are too much. Special strategies are also needed for flooding and drying areas.

3.4 Pressure Solution

As mentioned earlier there are several approaches for pressure treatment, such as the hydrostatic assumption, artificial compressibility, and pressure-correction relationships. The last group are based on combining momentum and incompressibility equations. Projection methods, the choice of this research, also need to define pressure or some kind of pseudo-pressure field through combining momentum and incompressibility relationships. This procedure leads to a 3D Poisson equation for pressure which must be solved implicitly at each time step. Solving such a relationship in a 3D domain is time-consuming[1]. Therefore, it is important to be aware of efficient solution algorithms.

3.4.1 The Pressure Equation

The Poisson equation for the pressure has been derived in section (3.2.6) as part of the projection method. However, it may be derived directly too. Assuming that the flow velocity is known, it is possible to define the pressure field from the velocity field. The basic procedure for deriving such a relationship is as follows:

1- Differentiate the u momentum equation with respect to x, the v equation with respect to y, and the w equation with respect to z.

2- Change the position of time and space derivations in the first term of each new equation.

For example: $$\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\right)\rightarrow\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right) \tag{3.19}$$

3- Add the three equations to each other.

4- Use the continuity equation to simplify the resulting equation.

For example $$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)=\frac{\partial}{\partial t}(0)=0 \tag{3.20}$$

Another way is to discretise the time derivatives in the momentum equations, derive new relationships for u^{n+1}, v^{n+1} , and w^{n+1} and then insert them in the discretised form of the continuity equation.

The resulting relationship is as follows (Ferziger and Peric, 2002):

$$\frac{\partial}{\partial x_i}\left(\frac{\partial p}{\partial x_i}\right)=-\frac{\partial}{\partial x_i}\left[\frac{\partial}{\partial x_j}\left(\rho u_i u_j-\tau_{ij}\right)\right]+\frac{\partial F_i}{\partial x_i}+\frac{\partial^2\rho}{\partial t^2} \tag{3.21}$$

For the case of constant density and viscosity this equation can be simplified further; as the viscous and unsteady terms disappear by virtue of the continuity equation, leaving:

$$\frac{\partial}{\partial x_i}\left(\frac{\partial p}{\partial x_i}\right)=-\frac{\partial}{\partial x_i}\left[\frac{\partial\left(\rho u_i u_j\right)}{\partial x_j}\right] \tag{3.22}$$

The Laplacian operator in the Poisson pressure equation is the product of the divergence operator originating from the continuity equation and the gradient operator that comes from the momentum equations. The outer derivative in equation (3.21) stems from the continuity

[1] Solving Poisson's equation for the pressure correction takes most of the time in computing non-stationary incompressible viscous flows (Wesseling, 2001). The main source of computational expense in simulations for incompressible flows is due to the solution of a Poisson equation for pressure (Yang and Voke, 2000). ...

equation while the inner derivative arises from the momentum equations. In a numerical approximation, the approximation of the Poisson equations must be defined as the product of the divergence and gradient approximations used in the basic equations. A violation of this constraint would lead to dissatisfaction of the continuity equation. Hence the outer and inner derivatives may be discretised using different schemes. To maintain consistency among the approximations used, it is best to derive the equation for the pressure term from the discretised momentum and continuity equations rather than by approximating the Poisson equation directly. Similarly, the right hand side of the pressure equation must be approximated in a manner consistent with their treatment in the momentum equations they are derived from (Ferziger and Peric, 2002).

Equation (3.21) shows the normal Poisson equation which relates pressure to other flow parameters. The right hand side of Poisson equation may be different for different pressure-treatment algorithms. In particular, there are different projection methods with each one using its own set of estimations, procedures and fractioning methods. Therefore pressure-updating formulas are not always the same. Sometimes other names such as pseudo-pressure are used. The common fact for different projection methods is that some kind of 3D Poisson equation must be solved to define the pressure field.

3.4.2 Poisson Solvers

The Poisson equation is a linear elliptic equation. It is linear because the flow variables in the right hand side are known from time- or fraction-step calculations. The discretised form of the equation is applied for each computational point. It leads to a linear system of equations. There are different direct and iterative solution algorithms for such a system on structured and unstructured grids. Direct matrix solvers such as the LU decomposition and tri-diagonal algorithm (Thomas Algorithm, TDMA) are useful and very efficient (TDMA) for 1D problems. There are also 5- or n- or even block-diagonal matrix solvers which are efficient for 2DV problems on structured grids. However there is not any direct matrix solver suitable for real engineering problems in 3D[1]. The algebraic systems for unstructured-mesh 3D engineering problems are usually very large and extremely sparse. Iterative matrix solvers are the only efficient algorithms for such systems[2].

[1] Sometimes it is possible to break-down a 3D problem on structured mesh into several smaller 2D problems.

[2] With one exception: the equation for the pressure correction can often be solved very efficiently by so-called fast Poisson solvers, based on Fourier transformation and/or cyclic reduction (Wesseling, 2001). This algorithm does not work with unstructured grids.

Several iterative solvers have been proposed by researchers. Here is a list of some of the most widely used ones:

- Jacobi’s method,
- Gauss–Seidel method,
- Successive over relaxation (SOR) and Symmetric SOR,
- Incomplete LU decomposition (Stone's method),
- Adding time derivative and use unsteady problem algorithms (like ADI),
- Newton and quasi-Newton methods,
- Conjugate gradient method (CG),
- Bi-conjugate gradient method,
- Preconditioned conjugate gradient method (PCG),
- Nested iteration method,
- Multi Grid Methods,
- Generalized minimal residual method (GMRES)
- Other less famous methods or sub-methods.

Some of these algorithms are restricted to symmetric matrices and some are more general. Most of the methods need the matrix to be positive definite. The efficiency of iterative methods varies. Rapid convergence of an algorithm is the key for effectiveness. This behaviour measures by the computing work (W) that each algorithm needed for convergence. Generally, iterative methods have W=Order(N^a) where a> 1 and N is the number of grid points. The optimal efficiency for an iterative algorithm is W=Order(N). Table 3.1 shows the efficiency of some algorithms for a steady state 2D Poisson problem. Unsteady problems are considered simpler, because they often come with a good first approximation from the previous time step (Trottenberg et al., 2001).

The CG method is a fast and efficient iterative solver (Kelley, 1995), but it is applicable only to symmetric and positive definite matrices. Non-symmetric system solvers are also needed in hydrodynamic modelling. There are extended versions of the CG method applicable to non-symmetric systems, such as the bi-conjugate gradient (BI-CG) method (Fletcher, 1976) and the bi-conjugate gradient stabilised (BI-CGSTAB) method (van der Vorst, 1992). The BI-CGSTAB method is a variant of the BI-CG method that improves irregular convergence rates so that residuals can be reduced smoothly.

A brief description about conjugate gradient, preconditioned conjugate gradient and multi-grid methods is given .

Table 3.1. Effectiveness of some iterative solvers (2D Poisson), after Trottenberg et al, 2001

Method	Number of operations
Gaussian elimination (band version)	Order (N^2)
Jacobi iteration	Order (N^2 log(ε))
Gauss-Seidel iteration	Order (N^2 log(ε))
Successive over-relaxation (SOR)	Order ($N^{3/2}$ log(ε))
Conjugate gradient (CG)	Order ($N^{3/2}$ log(ε))
ADI	Order (N log(N) log(ε))
Multigrid (iterative)	Order (N log(ε))
Multigrid (FMG)	Order (N)

3.4.3 Conjugate Gradient Method

CG is the most popular iterative method for solving large systems of linear equations with a symmetric, positive-definite sparse matrix. The main advantages of conjugate gradient method are that it uses relatively little memory for large scale problems and fast at each time step. It typically converges more slowly than Newton or quasi-Newton methods, but convergence is guaranteed.

Like many other global solvers, the conjugate gradient method is based on converting the original system of equations into a minimization problem. It is an improvement to the steepest descents algorithm, which is a natural way to find the minimum value via following the opposite to the gradient from an initial guess (similar to following the steepest downward path as a starting point on a curved surface). Then search for the lowest point on that direction and use the new position as the starting point for the next iteration until convergence. In this way, convergence is guaranteed, but for a real problem it is slow (Ferziger and Peric, 2002). The more ill-conditioned the matrix (that is, the larger its condition number[1]), the slower the convergence of the method (Shewchuk, 1994). The idea of CG is to minimize a function with respect to several directions simultaneously while searching in one direction at a time. Each new search direction is required to be conjugate to all the preceding ones (Ferziger and Peric, 2002). This algorithm always produces a new linearly independent search direction until convergence. The convergence of CG is much

[1] Matrix condition number = its maximum eigenvalue / its minimum eigenvalue.

quicker than that of steepest descent (Shewchuk, 1994). Nevertheless, it depends on the condition number of the matrix concerned.

The residual, which must be determined during each iteration. It may be calculated via the normal and accurate relationship or through a fast estimating formula. The fast recursive formula is usually used to speed up the procedure. After every selected number of iterations, the exact residual may be recalculated in order to remove accumulated floating point error (Shewchuk, 1994).

3.4.4 Preconditioned Conjugate Gradient Method

The CG method can be improved by replacing the problem by one with the same solution but a smaller condition number. This is called preconditioning. One way to precondition the problem is to pre-multiply the equation by another carefully chosen matrix. Therefore, the matrix equation ($M^{-1} A x = M^{-1} b$) will be solved instead of ($A x = b$).

The pre-conditioner matrix must be chosen carefully in such a way that:

1- The condition number of ($M^{-1} A$) is much smaller than the condition number of (A) itself.

2- The new matrix ($M^{-1} A$) remains symmetric.

Finding such a matrix is possible for instance, by the so called incomplete Cholesky factorization or incomplete LU decomposition (Stone's method) (Wesseling, 2001; Ferziger and Peric, 2002; Shewchuk, 1994). For most of the iteration methods (Krylov subspace types), the choice of a good pre-conditioner seems to be more crucial for efficiency than the choice of a particular iterative method (Wesseling, 2001).

3.4.5 Multigrid Method

The first studies investigating multigrid method in a strict sense were those by Fedorenko (1964) and that of Bakhvalov (1966). Multigrid is more a strategy than a particular method. The basis for this strategy is an observation about iterative methods, whose rate of convergence depends on the eigenvalues of the iteration matrix. Some iteration methods produce errors that are smooth functions of the spatial coordinates, because after a few iterations the rapidly varying components of the error may be removed. When the error field is smooth, the update can be computed on a coarser grid. In this way, the dimension of the problem is greatly reduced and each iteration costs considerably less time. Furthermore, iterative methods converge much faster on coarser grids. It is possible to use more than one grid coarsening level to gain more benefit from the multi-grid method.

A two-grid multigrid algorithm involves the following steps (Ferziger and Peric, 2002):

- On the fine grid, perform iterations with a method that gives a smooth error,
- Compute the residual on the fine grid,
- Restrict the residual to the coarse grid (smoothing, filtering, restricting),
- Perform iterations of the correction equation on the coarse grid,
- Interpolate the correction to the fine grid (prolonging),
- Update the solution on the fine grid,
- Repeat the entire procedure until the residual is reduced to the desired level.

Different methods are possible for restricting and prolonging. It is also possible to use more than one coarsening steps, which opens different opportunities for going back and forth between the grids. Therefore several versions of multi-grid are possible. Figure (3.4) illustrates some possible multigrid compositions for two and three level coarsening.

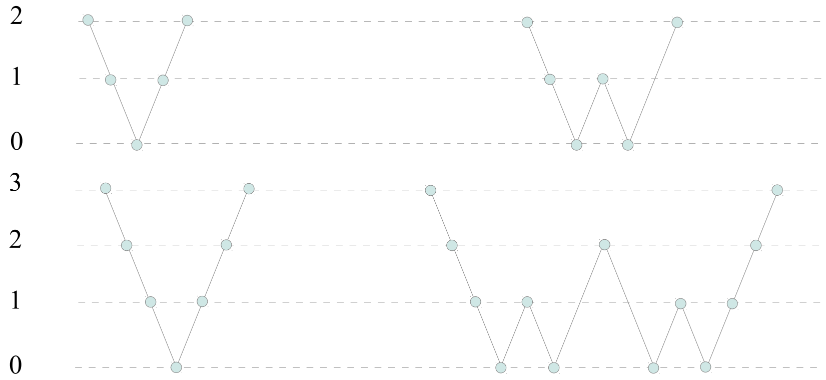


Figure 3.4. Grid levels for the V and W cycles for 2 and 3 levels
(after Knabner and Angermann 2003)

3.5 Summary

The major difficulties of 3D numerical modelling of free surface flows and their solution algorithms have been presented in this chapter. The problem of pressure treatment has been described and several widely used solutions have been presented. The projection method has been selected as the solution for the present study and is described in more details. The projection method is based on the fractional steps algorithm, which has been used widely in the present study for 3D and 2D hydrodynamic modelling and also for implicit integration of the 2D/3D models. Some existing algorithms for allocating the free surface position have been reviewed. The depth integrated continuity equation will be used in the present study which is a simple and robust method for defining the free surface.

The sigma-coordinate and its vertical gradient adaptive version GAS (Wai and Lu 1999) have been reviewed in this chapter in order to address the variability of the solution domain. These coordinate systems allow the mesh nodes to move vertically while they are fixed in the horizontal plane. The GAS coordinate system has been used in the present study. Using this system, the transformed grid spacing automatically adjusts in time and space according to the local solution gradient of the selected variable.

The Poisson pressure equation and some large matrix solvers have been reviewed in the rest of the chapter. Efficient matrix solvers are needed not only for the Poisson pressure system of equations, but also for the implicit free-surface solution procedure. In the present study, the preconditioned conjugate gradient method and the preconditioned bi-conjugate gradient stabilised method will be used for the hydrodynamic pressure and the free surface treatment respectively. The generalized minimal residual method is used for situations where the mentioned algorithms are not successful.

4 Review of Discretisation Algorithms

4.1 Introduction

The governing equations for the flow and the advection diffusion phenomenon generally do not have an analytic solution. Thus they are discretised into a system of algebraic equations with a finite number of unknowns as the solution quantities at the finite number of locations in space and time (for example at nodes, elements or cells). The system of equations may be solved numerically by computer. There are several discretisation frameworks and algorithms in the literature. The stability, accuracy, conservativeness, shape preservation and computational efficiency are important properties that a discretisation scheme ought to satisfy to a sufficient degree. The positivity of the method is also essential in the case of mass transport problems. This chapter presents a general review of the discretisation frameworks and algorithms.

The governing equations may be fractioned into distinguished parts in order to discretise each fraction with the most efficient algorithm. Therefore the splitting approach will be described before the discretisation methods. Practical hydro-environmental problems are usually advection dominated. Traditional discretisation methods are not difficulty-free for advection dominated flow. There are many advection dominated discretisation algorithms in the literature. This chapter includes a general review of these algorithms.

Interpolation is an essential part for most of the numerical algorithms. When it comes to the semi-Lagrangian algorithm, the interpolation accuracy and stability become more important. Linear interpolation is not sufficient for the advection dominated problems while traditional higher order schemes are at a risk of instability. On the other hand, sophisticated interpolation procedures are relatively straightforward on a structured grid. However, this is not the case when an unstructured grid is used. This chapter includes a short review on several unstructured interpolation procedures that have been proposed in the literature.

Wetting or drying may happen at part of the solution area. Numerical simulation of such an area needs special attention. This chapter includes a short review of several handling methods in this regard.

4.2 Splitting Approach

The governing equations of fluid flow and mass and solute transport involve different physical processes. The main processes are the advection, diffusion, pressure influence, Coriolis effect and input (or output) as source terms. Each process has its own behaviour, length-scale and time-scale. For some hydro-environmental problems a process may even have different scales in the vertical and the horizontal directions. The numerical characteristics and the optimum solution methods for the mentioned processes are not the same. The method of fractional steps (introduced by Chorin, 1968 and Yanenko, 1971) makes it possible to decouple a problem into consecutive smaller and easier stages and to solve each stage via a most efficient numerical algorithm.

The idea of the fractional steps method may be used in several ways. The algorithm of projection method, introduced by Chorin (1968) for solving the 3D incompressible flow problem, is an example of the fractional steps method. The directional splitting algorithm is another example which breaks a 2D or 3D problems into two or three 1D problems over a structured grid. The fractional steps method is also named the operator splitting algorithm when it is used for separating the physical processes such as advection, diffusion and so on. Equation (4.1) illustrates a typical operator splitting algorithm:

$$\frac{\partial f}{\partial t}=\underbrace{\frac{f^{n+1}-f^{diffusion}}{\Delta t}}_{remaining\ part}+\underbrace{\frac{f^{diffusion}-f^{advection}}{\Delta t}}_{diffusion\ part}+\underbrace{\frac{f^{advection}-f^{n}}{\Delta t}}_{advection\ part} \tag{4.1}$$

where f^n stands for the solution at the previous time step, $f^{advection}$ stands for the solution of the advection part of the governing equation, $f^{diffusion}$ stands for the solution of the diffusion part, and f^{n+1} stand for the solution at the new time step. Some parts of the source terms may be involved in the diffusion and advection steps.

The fractional steps method may be used along with the semi-implicit time marching (for example see the difference between the versions I and II of the projection method). The fractional steps algorithm used for 3D incompressible modelling is sometimes a combination of operator splitting (decoupling the pressure from the velocity) and directional splitting (decoupling the vertical diffusion from the horizontal diffusion) methods.

The order of time accuracy for a fractional step algorithm depends on how it moves from the known time step up to the new time step. For example the following algorithm is of the first order:

$$\frac{\partial f}{\partial t} = \underbrace{\frac{f^{n+1} - f^{flux\ terms}}{\Delta t}}_{source\ terms\ part} + \underbrace{\frac{f^{flux\ terms} - f^{n}}{\Delta t}}_{advection + diffusion\ part} \tag{4.2}$$

and the following algorithm is of the second order:

$$\frac{\partial f}{\partial t} = \underbrace{\frac{f^{n+1} - f^{flux\ terms}}{\Delta t/2}}_{source\ terms\ part\ repeated} + \underbrace{\frac{f^{flux\ terms} - f^{source\ terms\ *}}{\Delta t}}_{advection + diffusion\ part} + \underbrace{\frac{f^{source\ terms\ *} - f^{n}}{\Delta t/2}}_{source\ terms\ part} \tag{4.3}$$

There are several choices for splitting the governing equations into smaller stages. However the robustness of the overall behaviour must be taken into account. The optimum algorithm depends on the behaviour of the problem in hand.

Some of the fractional steps algorithms define the horizontal and vertical velocities jointly and some others compute the vertical velocity in a separate stage. Some of them handle the advection and diffusion terms jointly and some others breakdown them into different stages. It is common to handle all of the source terms in the diffusion phase and leave the advection phase to be free of force, in order to make a characteristic method more robust at the advection stage. The barotropic part of the hydrostatic pressure (the free surface slope) may be included in the diffusion step. It is usually more effective on the horizontal velocity than the diffusion effect, except for the baroclinic currents and the wind induced currents where the vertical diffusion dominates. An implicit discretisation of the vertical diffusion and the barotropic pressure terms is crucial in order to use a reasonable and practical time step. Some of the fractional steps algorithms split the pressure term into the hydrostatic and hydrodynamic parts, with the hydrostatic part being solved in the first step and the hydrodynamic part later. Some others involve also the explicit values of the hydrodynamic pressure in the first step. The splitting algorithm has been used extensively in the present study.

4.3 Discretisation Frameworks

The most common discretisation frameworks are the finite difference method (FD), the finite volume method (FV), the finite element method (FE) and the semi-Lagrangian characteristic method for advection. There are also other approaches such as the boundary element and the spectral method that are less common in the hydro-environmental modelling field. The aim of each method is to discretise the governing equations (the differential form or the integral form) relevant to a continuous problem into a discrete problem which can be solved using a computer.

The finite difference method (see Abbott and Basco, 1989) is usually based on the structured grid and is most efficient when the spacing is equidistant with quadrangle computational cells. The problem domain is covered by a grid of nodal points and the components of the governing equations are approximated using Taylor series expansions The FD method does not guaranty the mass conservation naturally. The goal is to satisfy the differential form of the governing equations at the grid points to some degree. It is possible but not straightforward and maybe optimum to extend the FD method to the unstructured 2D/3D grids. Other numerical algorithms usually use the FD method for the time discretisation.

In the the FE method (see Zienkiewicz and Taylor, 2000), the local approximation is often substituted into an appropriately weighted integral of the governing equations (the weak formulation). The integrals are evaluated in such a way to minimise a residue parameter. The domain is covered by a mesh of elements and the terms in the governing equations estimated in terms of functions that interpolate the nodal values over the elements. The FE method usually is based on the unstructured computational grid. The mass conservation in the FE method can be secured just in a global system.

The aim of the finite volume discretisation method (see Versteeg and Malalasekera 1995) is to balance the fluxes between arbitrarily chosen control volumes through their boundaries. The idea is to satisfy the integral form of the conservation law(s) to some degree of approximation for each control volume. This method can be adapted efficiently to the unstructured grids which is suitable for complex geometries. The FV method is commonly treated as a natural method for the fluid mechanics, because it is based directly on the flow properties. This method can perform very good conservative properties in the flow field. The FV method is usually based on the unstructured computational grid. The FV method can guarantees mass conservation at the local control volume level. Most of the FV methods use the divergence theorem in order to convert the over-cell gradient integration into over-face integrations. Several estimation algorithms have been proposed in the literature in order to define the cell-face values from the cell-averaged or the cell centred ones. Some of the FV methods use finite element shape functions in this regard. They are called hybrid finite volume finite element algorithm or sometimes control volume finite element (CVFE) algorithms. The last name had been used in some early papers also as the finite volume method on the unstructured-grid. The hybrid finite volume finite element algorithm has been used in the present study for the Eulerian parts of the solution algorithm.

The governing equations are mostly solved under the Eulerian framework in which the computational grid is fixed. Even if the grid points are allowed to move vertically (the σ coordinate approach) the governing equations may be transformed into the σ coordinate system and discretised on the fixed grid. There are many sophisticated Eulerian algorithms for simulating the hydrodynamic processes. However, the time step is restricted to certain values of Courant number for such models, even for the implicit algorithms. For the explicit schemes schemes Courant number is generally restricted to be less than one (because of the stability condition). On the other hand, a large Courant number results in a decrease in the accuracy of the numerical schemes. The Lagrangian approach is another framework to solve the governing equations. In this approach, the computational grid moves along the fluid trajectories or the characteristic lines (they are the same for the pure advection). This approach is a natural idea for the pure advection problem. However the movement of the computational points along with the fluid particles distorts the grid. Another approach is the Lagrangian-Eulerian or the semi-Lagrangian framework (Donea 1983, Healy and Russell 1993) in which the computational points of an Eulerian grid are tracked backward along the fluid particle trajectories or the characteristic lines just for one time step in order to form a Lagrangian grid. Then the information obtained from the previous time step is projected from the Eulerian grid to the Lagrangian one. This procedure is repeated for each time step. A forward-tracking version of the approach is also possible. By using the semi-Lagrangian method, the Courant number restriction is eliminated, as is the case for the Lagrangian method, while the risk of grid-distortion disappears. However, some kind of interpolation or data-reconstruction is needed at each time step. The interpolation procedure must be handled with such an accuracy that eliminates the cumulative damping effect as much as possible. When the simulation is long, the conservativeness of the scheme becomes crucial. There are several conservative semi Lagrangian methods in the literature. Section (4.4) gives more description in this regard.

The present model uses the semi-Lagrangian method for the advection stage and the FV method (with the finite element shape functions) for the diffusion step and the source terms. Indeed the FE method is at least as good as FV for the pure diffusion problem. Additionally, some highly effective source terms such as the barotropic pressure term, are also handled in the diffusion step in the present algorithm. Therefore the local mass conservation property of the FV method is an advantage.

Based on the geometric shape of control volumes and the position of variables, a FV model grid may be cell-centred, cell-vertex, or mixed. The later one means that some quantities are cell-centred and some others are cell-vertex. The most important subjects with the FV method are the approximating scheme of the boundary integrals (numerical fluxes), the distribution function of each quantity within the cell or the numerical estimation procedure for the quantity gradients over the cells, the method of dealing with discontinuities and the oscillations, the numerical estimation of the source terms and the time discretisation.

Most of the numerical models use the finite difference method for the time discretisation. There are also mixed space-time algorithms which are less common. In many existing models the time step is the same for all of the grid points at each time advance (global time step). There are also time discretisation algorithms based on the local time stepping approach. The time discretisation may be explicit, implicit or semi-implicit. The explicit schemes do not need a matrix solver to solve a system of equations. However their time-step is restricted with the Courant number less than one because of stability problems (except for the semi-Lagrangian methods). The time accuracy of the explicit schemes may be of the first order or higher. The Runge–Kutta algorithm is an example of the higher order explicit schemes. The implicit schemes are stable with Courant numbers larger than one. However a large system of equations must be solved at each time step. The time step of a fully implicit scheme is controlled by the accuracy instead of the stability. A compromise between the efficiency and the time step is the semi-implicit (mixed implicit and explicit) algorithm in which the fast evolving terms are handled implicitly while the other terms are handled explicitly.

The local time step algorithm uses different time steps for different cells or blocks of cells. There are different algorithms of this type of algorithms, including: the adaptive time refinement method, the domain decomposition method, the variable time-stepping method, the adaptive time space discretisation method and the self-adaptive flux-conserving local time stepping method. All of these algorithms are explicit (for details see Omelchenko and Karimabadi, 2006). The implicit domain decomposition algorithm using the structured nested grid is also available via the ADI method and double sweep tridiagonal matrix solver (Namin et al. 2001).

The present model is largely based on the global time step in order to use implicit schemes easily. However, the advection step is based on local time stepping in order to define the characteristic path of each computational point.

4.4 Advection Dominated Discretisation Algorithms

Practical hydro-environmental problems are usually advection dominated. When applied to such problems, simple Eulerian discretisation methods of any type (FD, FE or FV) are likely to fail unless very small time steps are used. They produce numerical oscillations or high numerical diffusion in regions with high gradients. On the other hand, the semi-Lagrangian methods have their own challenges both due to the damping effect caused by repetitive interpolations and the concerns of mass balance when long period of time is considered.

Several shock capturing algorithms have been developed in order to eliminate the numerical oscillation and minimise numerical diffusion. The characteristic behaviour of the advection process plays an essential role in deriving shock capturing algorithms. It is not always easy to use the shock capturing algorithms with unstructured 2D and 3D grids, because many of them were introduced in one dimensional space.

Some of the well known advection dominated discretisation schemes are the hybrid (central/upwind) scheme, the exponential scheme, the Godunov's FD algorithm (1959), the Fromm's scheme (Fromm, 1968), the Monotonic Upstream Schemes for Conservation Laws (MUSCL) (van Leer, 1973), the Lax-Wendroff scheme, the Holly-Preissmann characteristic method (1977), the QUICK (Quadratic Upstream Interpolation for Convective Kinematics) method (Leonard, 1979), the QUICKEST (Quadratic Upstream Interpolation for Convective Kinematics with Estimated Streaming Terms) method (Van Leer, 1977; Leonard, 1979 ;Leonard, 1988), the Roe's method (1981), the HLL method (Harten et al., 1983), the cubic-spline characteristic method (Branski and Holley, 1986) and it's later improvements, the six-point characteristic method of Komatsu et al. (1985, 1989), the UNO (Uniformly Non-Oscillatory) scheme (Harten and Osher, 1987), the HLLE method (Einfeldt, 1988), the ULTIMATE-QUICKEST algorithm (Leonard, 1991), the SOUCUP (Composite Second-Order Upwind/Central Difference/First-Order Upwind) scheme (Zhu and Rodi, 1991), the HLPA (Hybrid Linear/Parabolic Approximation) scheme (Zhu, 1991), the UTOPIA (Uniformly Third Order Polynomial Interpolation Algorithm) method (Leonard, 1993), the NIRVANA (Non-oscillatory Integrally Reconstructed Volume-Averaged Numerical Advection) scheme (Leonard, 1995), the PPM scheme (Colella and Woodward, 1984; Savic and Holly 1993), the ENO (Essentially Non-Oscillatory) scheme (Shu and Osher, 1988; Nujic, 1995), the HLLC method (Fraccallo and Toro, 1995), the WAF method (Fraccarollo and Toro, 1995), the WENO (Weighted Essentially Non-Oscillatory) scheme (Levy et al., 2002), the HLLEC method (Váchal et al., 2004) and different types of the Total Variational Diminishing (TVD) methods.

For the finite element method in particular, the most known advection dominated algorithms are the generalised Galerkin method (GGM), the Taylor-Galerkin method (Donea, 1984), the streamline upwind Petrov–Galerkin method (Heinrich et al., 1977; Brooks and Hughes, 1982), the Lagrange–Galerkin method, the least squares-Galerkin method (Hughes et al, 1989), the characteristic Galerkin method (Zienkiewicz and Codina, 1995), and the discontinuous Galerkin method (Oden et al, 1996).

The present model solves the pure advection process using the fractional splitting algorithm. The semi-Lagrangian approach has been used in this regard. It is an explicit approach which is unconditionally stable with no time step restriction. With this approach, interpolation is carried out at each time step in order to transform data from Eulerian grid to Lagrangian grid (or vice versa for the forward tracking algorithm). Low order interpolation results in high damping effect after several time steps. It is necessary to use some kind of high-order interpolation. Therefore some kind of limiter is also needed (monotonicity constraint) in order to prevent numerical oscillations. The basic form of the SL (semi-Lagrangian) method is not conservative. This behaviour does not cause practical difficulties for short period simulations because the SL methods, when high-order interpolation procedures are used, are highly accurate. However mass conservation is critical for long period simulations.

Several conservative SL methods have been introduced by researchers. The key idea is to transform the computational cells instead of the grid points alone. The whole mass (of any conservative material) inside each cell remains constant for pure advection during the Lagrangian movement at each time step. Then the Lagrangian cell masses may be distributed conservatively over the joint Eulerian cells. The interpolation functions may be integrated on each shared area or volume (between the Eulerian and Lagrangian cells) separately to enable conservative data transformation. The Gauss theorem may also be used, in order to change the over cell area (or volume) integration into over cell boundary integration. Calculating the nodal values on the base of the cell averaged (or integrated) values is an essential step in order to construct a fully conservative and stable SL scheme (Mohamadian, 2006). The conservative SL schemes may be classified in two categories: a) corrective schemes and b) inherently conservative schemes. For most of the second type, mass conservation is obtained by imposing a constraint at the polynomial interpolation step. The inherently conservative schemes themselves may be of the exactly conservative methods formulated in a non-conservative form (CIP-CSL) (for example see Tanaka et al. 2000, Yabe et al. 2001, Xiao and Yabe 2001, Xiao 2002) or may be one of the SL integrated mass (SLIM) methods (see Laprise and Plante 1995, Zerroukat et al. 2002, Mohamadian and Le Roux 2006).

The mentioned above methods are based on some kind of "piecewise exact" approach in order to project the information from the Eulerian grid into the Lagrangian one. This approach, while works well, is difficult and computationally expensive to extend to unstructured grid models. Most of these methods have been studied on 1D or structured 2D/3D grids. Extending such methods to unstructured 3D grids with high order interpolation functions is neither easy nor efficient. An alternative approach has been introduced by Mofrad (1999) which is based on tracking the Gaussian quadrature points of each cell instead of its boundaries. In this way, the numerical integration over each cell is computed using the associated Gaussian points. A reasonable accuracy through this algorithm needs 7 Gaussian points for each 2D cell and 11 Gaussian points per each 3D cell. Extending this approach to an unstructured 3D grid model is relatively simple and straightforward.

The conservative versions of the SL methods are considerably time consuming in comparison with the non-conservative ones, especially if an unstructured grid is used.

3D hydro-environmental models are usually not used for problems requiring a long period simulation. Accurate interpolation SL schemes are sufficient for short period problems in many cases. The present model is based on an accurate but not conservative semi Lagrangian scheme for the 3D model. This approach has been selected in order to make the model perform more efficiently.

Some of the high resolution numerical schemes, both for discretisation and interpolation, have been introduced firstly in the one dimensional space. There is a generic way to extend a 1D scheme to 2D/3D schemes with structured grids, based on the splitting approach. The 2D (or 3D) problem is divided into two (or three) one dimensional problems. Then it is possible to use all of the scheme developed for 1D problems. For example, the two dimensional advection problem:

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=S \tag{4.4}$$

may be split as follows:

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}=S_x \quad , \quad \frac{\partial U}{\partial t}+\frac{\partial G}{\partial y}=S_y \tag{4.5}$$

However, this method is not useful when the computational grid is unstructured. It is not always possible or efficient to extend higher order interpolation functions such as the cubic splines or the third-degree polynomials into the unstructured 2D/3D grids. There are some studies that use one dimensional scheme along the direction orthogonal to the cell interfaces (Namin 2003, Mohamadian et al. 2006). However, they are not as accurate as their 1D counterparts, because of the lower-order interpolations used to transform data from the computational points to other needed locations (for example from the cell centres to the vertices or to define the mean values on the cell edges). There are also interpolation/re-construction procedures in the literature for unstructured grids with higher orders of accuracy.

4.5 Interpolation on Unstructured Grid

Interpolation is an essential method for most of the numerical methods. It is used in the FE method for computing the shape functions, in the FV method for reconstructing data at the cell edges, and in the semi-Lagrangian method for defining data at the departure (or arrival) points. The accuracy of the first order interpolation scheme is usually not sufficient. Therefore, higher order interpolation schemes are needed. The interpolation procedure is relatively straightforward on a structured grid. However, it is not the case when an unstructured grid is used.

Several unstructured interpolation procedures have been proposed in the literature. There are plenty of shape functions for the FE method in this regard. Several FV methods are also exist, with most of them being based on the cell-averaged gradients of the solution quantity. Some schemes to be listed in this regard are the Pan and Cheng method (1993), the large stencil method by Jawahar and Kamath (2000), the method of Wang and Liu (2000), the maximum limited gradient method by Brufau and Garcia-Navarro (2000), and the least-squares reconstruction method by Wang (2000). Some of these procedures take the variation of the gradient within a cell into account for defining the cell-averaged gradients. However the data reconstruction is still based on the linear interpolation. It is possible to consider both the cell-gradient and the cell-curve (the second derivatives) into account (Namin, 2003). This method is more accurate than the previous ones.

Barth and Frederickson (1990) introduced a general approach considering high-order polynomials of arbitrary degrees using the neighbouring cell-averaged values and least squares formulation. Agarwal and Halt (1999) introduced a high-order scheme with third order interpolation polynomials by solving extra differential equations over the solution domain for the spatial derivatives of the governing equation or their moments. Harten and Chkravartht (1991) and Abgral (1994) extended the ENO finite volume scheme to the unstructured grid as an unstructured interpolation algorithm. Sonar (1997) studied the 2D polynomial recovery and accuracy over the triangular grid for the ENO algorithm. Friedrich (1997) introduced a weighted ENO algorithm (WENO) on the triangular. All of the cited procedures use some kind of limiter to avoid numerical oscillations except the ENO/WENO schemes that use selected neighbouring cells for polynomial fitting in this regard.

4.6 Horizontal Discretisation with σ-coordinate Mesh

The grid-layers of the sigma-coordinate mesh are not necessarily horizontal. Therefore, every horizontal gradient term in the governing equation is transformed into two terms; an on-layer gradient term and a vertical gradient one. The horizontal diffusion and the pressure gradient terms involve additional derivative terms in the vertical direction in a sigma coordinate system. The sigma coordinate mesh usually follows the bed slope. In such conditions, the horizontal gradient terms may cause significant errors near the steep bed slopes.

The horizontal gradient is numerically equivalent to the difference between the gradient along the sigma-coordinate (called the sigma gradient) and the vertical gradient (called the hydrostatic correction when it comes to the pressure gradient). When topography is steep or when the vertical gradient is large, the horizontal gradient becomes the difference between two large numbers. Therefore, even small errors in the computation of either term can produce relatively large errors in the horizontal gradient. The large truncation errors resulted from the summation of large terms of opposite sign in the σ-coordinate system could cause spurious flows, particularly near the bed (Stansby and Zhou, 1998). The problem becomes particularly significant when some kind of vertical layering/stratification is happened in the solution quantity and the mesh is compressed near the bed. For density induced currents, Stelling and van Kester (1994) have shown that artificial circulation can be produced due to numerical errors in the baroclinic terms.

There are primarily four approaches developed to avoid the problem mentioned above. The first approach is using higher order numerical schemes (Auclair et al. 2000, Shchepetkin and McWilliams 2003). The second approach is based on mixing the sigma coordinate mesh with the z-level mesh (for example see Mike3, 2009). This method is called eta-coordinate. In this way, the upper part of the domain is discretised using the sigma coordinate approach while the lower part and near the bed is discretised using the Cartesian z-level approach (stepwise topography). Therefore, the variable water level is handled with the sigma transformation while the near bed region, the area where the combination of steep slope and stratification is probable, is handled with real horizontal layers. This approach needs further enhancements for simulating the process of down-slope spreading of a dense plume density current. Because in sigma coordinates the process is dominated by down-slope along-sigma advection driven by pressure gradients. However, in stepwise topography it is a combination of horizontal advection and diffusion that results in intense vertical mixing and thicker boundary layers.

The third technique is to decompose the solution quantity into a vertical varying base state and a perturbation from the base state. The horizontal gradient can then be calculated from the perturbation rather than from the total quantity. This approach is suitable in particular for the pressure gradient because the pressure gradient and the hydrostatic correction are several orders of magnitude larger than their difference. Some references in this regard can be found from (Corby et al. 1972, Gary 1973, Mellor et al. 1994, Mellor et al. 1998, Batteen 1988, Song 1998, Robertson et al. 2000, Bernardet 2005).

The fourth approach is to calculate the horizontal gradients on a true horizontal plane instead of using just the neighbouring nodes/cells/elements on the same sigma plain. In this way, the horizontal discretisation terms are not transformed into the sigma coordinate system. Interpolation in the vertical direction (between the layers) is needed before computing the horizontal gradients. There are some enhancements for this method close to the bed in order to avoid intersecting the topography. Some references in this regard are (Simmons and Burridge 1981, Mellor and Blumberg 1985, Haney 1991, Stelling and Kester 1994, Stansby 1997, Stansby and Zhou 1998, Ye et al. 1998, Li and Atkinson 1999, Mahrer 1984, Fortunato and Baptista1996, Song 1998, Zängl et al. 2004).

The fourth approach has been used in the present model both for computing the pressure gradient and horizontal diffusion. On the other hand, using the projection method the pressure has been de-composited into the hydrostatic and hydrodynamic parts, which is in harmony with the third technique.

4.7 Discretisation in Flooding/Drying Area

Wetting and drying may happen at part of the computational domain of coastal areas and reservoirs. Some coastal domains involve tidal-flats that are influenced by the wetting/drying process. The upstream parts of many reservoirs are also affected by the wetting/drying process because of water level variation. It is not always possible to ignore the wetting/drying area in the numerical simulation.

Special attention is needed in numerically simulating the wetting/drying process. Both the variability of the solution domain and the severely shallowness of the nearby area must be addressed in this regard. The landlocked elements (or cells) may also be a source of difficulty. They are wet elements surrounded by non-active (dried) elements. It is usual to choose limiter depth values (a few decimetres) for both the wet-to-dry and dry-to-wet processes. These two values may be different in order to prevent oscillation.

In some numerical models the computational grid is allowed to change with the water boundaries (for example see Gopalakrishnan 1989, Roig and Evans 1993, Lin et al. 2004). This approach works well in smooth topography areas. However, it is computationally intensive and may lead to ill-shaped elements when the topography is complex. Many numerical models use fixed computational grid and let the elements to be wet and dry during the solution process. It is possible to turn the elements active and inactive or on and off (for example see Jia and Wang 1999, Dietrich et al. 2006, Kruger and Rutschmann 2006). Another approach is to remove selected terms from the governing equations (Falconer and Chen 1991). (Roig 1995) uses variable marsh porosity in order to lower the ability of a dried element to hold water (RMA2 manual 2006). Bradford and Sanders (2002) let the element to be half wet and half dry (also in Lu 2003). An efficient approach for the last scheme is to set velocities equal to zero for partly-dried elements and ignore the momentum solution while the mass conservation is solved (Zhao et al. 1994, Sleigh et al, 1998, .Lu 2003). The outward flux, the bed slope and the friction values may be set to zero for the dry elements while inward fluxes remains active as a mechanism for re-wetting the dry elements. It means setting the momentum fluxes to zero and only taking the mass fluxes into consideration (Zhao et al. 1994, Sleigh et al. 1998, Mohammadian and Roux 2006).

Another approach is to handle all of the wet and dry elements in a unique solution procedure without any on-off or on-partly-off scheme. It is called the thin slot algorithm. The idea of the thin slot algorithm is to lett the dry elements to have a thin layer of water but increasing their bed resistance to a very high value. An intermediate depth-zone for smooth change of roughness between the wet and dry elements is necessary (MacArthur et al., 1990; Bates and Hervouet, 1999; Namin, 2003; Nielsen and Apelt, 2003). The bed roughness terms in this way must be discretised implicitly in order to prevent instability.

Some of the shallow water models, which are based on the approximate Riemann solution, use the analytical solution of the Riemann problem in which the initial depth at one side of the cell-edge is zero in order to simulate the wetting process (for example see Brufau et al. 2002, Titov and Synolakis 1995, Yoon and Kang 2004, and Kim et al. 2004). This is a good approach for predicting dam-breaking flows.

Simulating the wetting/drying process is usually included in the shallow water models. However, many 3D numerical models do not have this capability. Some of the 3D models with the wetting/drying capability are: TRIVAST by Lin and Falconer (1997), RMA10 by King (1998), an extension to ECOM by Zheng et al. (2002), SSIIM by Olsen (2003), FVCOM by Chen et al. (2003), the model of Lu (2003), MIKE3-HD, MIKE3-FM and Telemac3D. Among the above mentioned list, FVCOM and Telemac3D are based on the σ–coordinate grid system. SSIIM uses blocks of structured horizontal grids with unstructured cells in the vertical direction. RMA10 is an unstructured FE model which uses the element removal method and marsh porosity (King, 1988 and 1998). The model of Lu (2003) is a layer integrated model, which handles the wetting/drying problem as a shallow water model. MIKE3-FM is based on the work by Zhao et al. (1994) and Sleigh et al. (1998). When the depths are small the governing equations are reformulated (setting the momentum fluxes to zero and only taking the mass fluxes into consideration) and when the depths are very small the elements/cells are removed from the “active” model domain.

The reason for requiring 3D models to offer wetting/drying simulation capability is the need to simulate a whole model domain which involves such areas. It does not mean that the near-drying or just-flooded areas need 3D simulation. When the water depth is so shallow as to be close to the drying depth, 2D modelling is more reasonable than 3D simulation.

4.8 Summary

A general review on the discretisation techniques has been presented in this chapter. The idea of the fractional splitting has been explained and preferred over the coupled algorithms because of its flexibility. The model developed in this thesis is based on the projection method II and the operator splitting algorithm in order to simulate the advection and diffusion processes separately through an optimised numerical algorithm. Common discretisation frameworks have been reviewed along with the advection dominated schemes. The semi-Lagrangian algorithm has been selected for use in the advection step and the FV scheme for the diffusion part. Several sophisticated interpolation algorithms have been cited as an essential part of the mentioned algorithms for unstructured grid models.

The problem of traditional horizontal discretisation algorithms on the σ-coordinate grid has been considered with exiting solution strategies being described. In this study the horizontal discretisation is performed on the real horizontal plane instead of transforming the horizontal gradient terms into the sigma coordinate system.

Wetting or drying may happen at some parts of the solution area, for which a special treatment is needed. Several handling methods have been reviewed in this regard. The method of Zhao et al. (1994) and Sleigh et al. (1998) has been selected for use in the current study. When the water depth is small the problem is reformulated (setting the momentum fluxes to zero and only taking the mass fluxes into consideration) and when the depth is very small the elements/cells are removed from the active computational domain.

5 Governing Equations to be Solved

5.1 Introduction

This chapter includes the governing equations that must be solved in the present study in order to address the hydrodynamic, water quality and sediment transport characteristics. Both the 3D and the 2D version of the governing equations will be discussed in this regard. Considering that the sigma coordinate system will be used, the transformation procedure and the transformed equations will also be presented.

5.2 Hydrodynamic Equations

The 3D mass and momentum conservation equations in the Cartesian coordinate system are:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=Q_{dis} \tag{5.1}$$

$$\begin{aligned}\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=&\overbrace{\frac{\partial}{\partial x}\left(\nu_h\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu_h\frac{\partial u}{\partial y}\right)}^{1}+\overbrace{\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u}{\partial z}\right)}^{2}\overbrace{-\frac{1}{\rho}\frac{\partial p_a}{\partial x}}^{3}\\&\underbrace{-g\frac{\rho_\eta}{\rho}\frac{\partial\eta}{\partial x}}_{4}\underbrace{-g\frac{1}{\rho}\int_z^\eta\frac{\partial\rho}{\partial x}dz}_{5}\underbrace{-\frac{1}{\rho}\frac{\partial q}{\partial x}}_{6}\underbrace{+2\cdot v\cdot\omega_z-2\cdot w\cdot\omega_y}_{7}+u_{dis}Q_{dis}\end{aligned} \tag{5.2}$$

$$\begin{aligned}\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=&\overbrace{\frac{\partial}{\partial x}\left(\nu_h\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu_h\frac{\partial v}{\partial y}\right)}^{1}+\overbrace{\frac{\partial}{\partial z}\left(\nu_v\frac{\partial v}{\partial z}\right)}^{2}\overbrace{-\frac{1}{\rho}\frac{\partial p_a}{\partial y}}^{3}\\&\underbrace{-g\frac{\rho_\eta}{\rho}\frac{\partial\eta}{\partial y}}_{4}\underbrace{-g\frac{1}{\rho}\int_z^\eta\frac{\partial\rho}{\partial y}dz}_{5}\underbrace{-\frac{1}{\rho}\frac{\partial q}{\partial y}}_{6}\underbrace{-2\cdot u\cdot\omega_z+2\cdot w\cdot\omega_x}_{7}+v_{diss}Q_{dis}\end{aligned} \tag{5.3}$$

$$\begin{aligned}\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=&\overbrace{\frac{\partial}{\partial x}\left(\nu_h\frac{\partial w}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu_h\frac{\partial w}{\partial y}\right)}^{1}+\overbrace{\frac{\partial}{\partial z}\left(\nu_v\frac{\partial w}{\partial z}\right)}^{2}\overbrace{-\frac{1}{\rho}\frac{\partial q}{\partial z}}^{6}\\&\underbrace{+2\cdot u\cdot\omega_y-2\cdot v\cdot\omega_x}_{7}+w_{dis}Q_{dis}\end{aligned} \tag{5.4}$$

where u, v and w are the velocity components in the x-, y-, and z-directions, respectively, ρ is the local water density, ρ_η is the water density at the free surface, p_a is the atmospheric pressure, g is the acceleration of the gravity, η is the water surface level, q is the hydrodynamic pressure, ν_h is the horizontal eddy viscosity, ν_v is the vertical eddy viscosity,

ω_x, ω_y and ω_z are the projections of $\vec{\omega}_E$, the rotational speed of the Earth around the S-N axis, in the x-, y- and z-directions respectively, Q_{dis} is the magnitude of the discharge due to the point sources, and u_{dis}, v_{dis} and w_{dis} are the velocity components representing water discharged into the solution domain.

The terms (1) and (2) in equations (5.2) to (5.4) are relevant to the horizontal and vertical diffusion processes, respectively. The terms (3) to (6) illustrate different parts of the pressure gradient. They stand for the atmospheric pressure gradient, the barotropic pressure effect, the baroclinic pressure effect, and the hydrodynamic pressure gradient respectively. The term (7) stands for the Coriolis effect.

The Earth rotational speed components ω_x, ω_y and ω_z are functions of the geographical latitude of the cell involved and the angle between the x-axis and the east-direction. If the x-axis is in the eastward direction, then ω_x is equal to zero.

The baroclinic pressure term in equations (2.26) and (5.3) involve the ratio of the surface water density to the local water density. It indicates that the effect of the free surface slope on the layers of a stratified flow depends of the layer density. This term may not be negligible in situations when both the density force and the free surface slope influence the current. It is derived from the pressure equation:

$$p = p_a + \int_z^\eta \rho\, g\, dz + q \qquad (5.5)$$

by applying differentiation to equation (5.5) in the x and y directions, respectively. For example in the x-direction we have:

$$\frac{\partial p}{\partial x} = \frac{\partial p_a}{\partial x} + \frac{\partial}{\partial x}\int_z^\eta \rho\, g\, dz + \frac{\partial q}{\partial x} = \frac{\partial p_a}{\partial x} + g\int_z^\eta \frac{\partial \rho}{\partial x} dz + g\,\rho_\eta \frac{\partial \eta}{\partial x} + \frac{\partial q}{\partial x} \qquad (5.6)$$

The Leibniz rule has been used in equation (5.6), because the upper limit of the integral is a function of x; the derivation parameter.

It should be noted that the term $-g$ (or an equivalent term) is not included in equation (5.4), while some versions of the vertical momentum equation in the literature include such a term. In fact, two versions of the vertical momentum equation exist in the literature in this regard. The difference is caused by whether or not the Boussinesq approximation (for ρ) is applied. The vertical body force due to the gravity effect has been cancelled in the present version with the vertical gradient of the hydrostatic pressure. This is the case when the Boussinesq approximation is not used. With the Boussinesq approximation, a small term (compared to the gravity term) remains in the equation which is equal to $-g\cdot((\rho-\rho_o)/\rho_o)$, where ρ_o is the average water density.

The kinematic boundary condition for the hydrodynamic equations at the free surface is given as:

$$\frac{\partial\eta}{\partial t}+u_\eta\frac{\partial\eta}{\partial x}+v_\eta\frac{\partial\eta}{\partial y}=w_\eta \tag{5.7}$$

The dynamic boundary conditions at the free surface are:

$$\nu_v\left(\frac{\partial u_\eta}{\partial z}+\frac{\partial w_\eta}{\partial x}\right)=\frac{\tau_x^s}{\rho_{water}}=\frac{\rho_{air}}{\rho_{water}}.C_w.W.W_x \tag{5.8}$$

and

$$\nu_v\left(\frac{\partial v_\eta}{\partial z}+\frac{\partial w_\eta}{\partial y}\right)=\frac{\tau_y^s}{\rho_{water}}=\frac{\rho_{air}}{\rho_{water}}.C_w.W.W_y \tag{5.9}$$

where ρ_{air} is the air density, ρ_{water} is the water density, W is the wind speed relative to the water velocity at the free surface, and C_w is the wind drag coefficient (see equation 2.26).

The kinematic condition for the hydrodynamic equations at the bottom boundary is given as:

$$\vec{V}_b\cdot\vec{n}_{bed}=n_x u_b+n_y v_b+n_z w_b=0 \tag{5.10}$$

where (n_x, n_y, n_z) is the outward unit vector normal to the bed.

Equation (5.10) is used usually as the boundary condition for the vertical velocity component because the bed slope is usually gentle or even negligible.

The dynamic conditions at the bottom boundary may be derived from the wall function. If the bottom-boundary node is located within an acceptable distance from the bed and the bed surface is rough, the following relationships may be used:

$$\frac{\vec{\tau}_b}{\rho}=\nu_t\frac{\partial\vec{V}_b}{\partial n} \tag{5.11}$$

$$\nu_t=\kappa d_o\frac{\left|\vec{V}_b\right|}{\frac{1}{\kappa}\ln\left(\frac{d_o}{(k_s/30)}\right)} \tag{5.12}$$

$$\frac{\vec{\tau}_b}{\rho}=-\frac{\left|\vec{V}_b\right|\vec{V}_b}{\left(\frac{1}{\kappa}\ln\left(\frac{d_o}{(k_s/30)}\right)\right)^2} \tag{5.13}$$

where d_0 is the normal distance between the real bed and a suitable point above it. It may be the no-slip level or the boundary between the bed slope and the suspended sediment load. $\vec{V}_b$ is the velocity at distance d_0 above the bed, $\vec{n}$ is the unit outward directed vector normal to the bed, $\kappa = 0.41$ is von Karman's constant, and k_s is the bed roughness height.

Applying equations (5.12) and (5.13) to equation (5.11) leads to the following relationship:

$$\frac{\partial \vec{V}_b}{\partial n} = -\frac{\vec{V}_b}{d_o \cdot \ln\left(\frac{d_o}{(k_s/30)}\right)} \tag{5.14}$$

Equation (5.14) is projected into the bed-parallel directions, $\vec{e}_{xn}$ and $\vec{e}_{yn}$, in the xn and yn planes, in order to present the bed stress components needed for the x- and y-momentum equations. If the bed slope is negligible, the planes xz and yz may be used instead of the planes xn and yn. Therefore, the dynamic bottom boundary conditions may be written as follows:

$$\frac{\partial u_b}{\partial z} = \frac{u_b}{d_o \cdot \ln\left(\frac{d_o}{(k_s/30)}\right)} \tag{5.15}$$

$$\frac{\partial v_b}{\partial z} = \frac{v_b}{d_o \cdot \ln\left(\frac{d_o}{(k_s/30)}\right)} \tag{5.16}$$

However, when the bed slope is not gentle or negligible, then the planes xn and yn are different from the planes xz and yz, with the later ones being not necessarily perpendicular to the bed (lateral bed slope). The direction $\vec{n}$ and the distance d_0 in equations (5.11) to (5.14) are considered perpendicular to the bed. They are not necessarily within the planes xz or yz. Even the directions $\vec{e}_{xn}$ and $\vec{e}_{yn}$ are not generally perpendicular to each other. Therefore the bottom boundary condition will be more complex in situations where the bed slope is considerable.

The distance between the bottom boundary point and the bed has to satisfy the following relationship:

$$30\nu < d_o \frac{\left|\overrightarrow{V_{do}}\right|}{\frac{1}{\kappa}\ln\left(\frac{d_o}{(k_s/30)}\right)} < 100\nu \tag{5.17}$$

where ν is the water viscosity.

The velocity or the pressure may be known at the boundary nodes as lateral boundary conditions. The water surface level may be used instead of the pressure term if the hydrostatic pressure assumption is adequate. The horizontal gradient of the vertical velocity component may be assumed to be zero at the lateral open boundaries:

$$\frac{\partial w}{\partial x}=0 \quad \text{and} \quad \frac{\partial w}{\partial y}=0 \tag{5.18}$$

When such simple forms of the outflow boundary conditions influence negatively the interior solution, then the non-reflecting or absorbing boundary conditions may be used.

The boundary conditions for the hydrodynamic pressure correction will be discussed in the next section.

5.2.1 Poisson Equation

The present study is based on the projection-2 method in order to estimate the effect of the hydrodynamic pressure gradient on the velocity field. A Poisson-like equation may be solved in this regard in order to define the hydrodynamic pressure correction. The momentum equations are solved numerically using the explicit value of the hydrodynamic pressure field from the previous time step. However the estimated velocity field is not necessarily convergence free. It is considered just as an intermediate solution. If the hydrodynamic pressure correction field was known, its spatial gradient could be used in order to project the intermediate velocity into the divergence-free velocity field via the following relationships:

$$\frac{u^{n+1}-\acute{u}}{\Delta t}=-\frac{1}{\rho}\frac{\partial \hat{q}}{\partial x} \quad , \quad \frac{v^{n+1}-\acute{v}}{\Delta t}=-\frac{1}{\rho}\frac{\partial \hat{q}}{\partial y} \quad , \quad \frac{w^{n+1}-\acute{w}}{\Delta t}=-\frac{1}{\rho}\frac{\partial \hat{q}}{\partial z} \tag{5.19}$$

where u^{n+1}, v^{n+1}, and w^{n+1} are the convergence-free velocity components, $\acute{u}$, $\acute{v}$ and $\acute{w}$ are the intermediate velocity components, and $\hat{q}\equiv q^{n+1}-q^{n}$ is the correction to the hydrodynamic pressure. Therefore the convergence-free velocity components may be written as follows:

$$u^{n+1}=\acute{u}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial x} \quad , \quad v^{n+1}=\acute{v}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial y} \quad , \quad w^{n+1}=\acute{w}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial z} \tag{5.20}$$

The quantities u^{n+1}, v^{n+1}, and w^{n+1} in the incompressibility equation may be defined from equations (5.20) which results in the following relationship:

$$\frac{\partial}{\partial x}\left(\acute{u}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial x}\right)+\frac{\partial}{\partial y}\left(\acute{v}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial y}\right)+\frac{\partial}{\partial z}\left(\acute{w}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial z}\right)=0 \tag{5.21}$$

and can be re-arranged as follows:

$$\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial \hat{q}}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial \hat{q}}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{1}{\rho}\frac{\partial \hat{q}}{\partial z}\right)=\frac{1}{\Delta t}\left(\frac{\partial \acute{u}}{\partial x}+\frac{\partial \acute{v}}{\partial y}+\frac{\partial \acute{w}}{\partial z}\right) \tag{5.22}$$

and then may be expanded as follows:

$$
\begin{aligned}
&-\frac{1}{\rho^2}\frac{\partial \rho}{\partial x}\frac{\partial \hat{q}}{\partial x}+\frac{1}{\rho}\frac{\partial}{\partial x}\left(\frac{\partial \hat{q}}{\partial x}\right)-\frac{1}{\rho^2}\frac{\partial \rho}{\partial y}\frac{\partial \hat{q}}{\partial y}+\frac{1}{\rho}\frac{\partial}{\partial y}\left(\frac{\partial \hat{q}}{\partial y}\right)+\frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial \hat{q}}{\partial y}\right)\\
&-\frac{1}{\rho^2}\frac{\partial \rho}{\partial z}\frac{\partial \hat{q}}{\partial z}+\frac{1}{\rho}\frac{\partial}{\partial z}\left(\frac{\partial \hat{q}}{\partial z}\right)=\frac{1}{\Delta t}\left(\frac{\partial \acute{u}}{\partial x}+\frac{\partial \acute{v}}{\partial y}+\frac{\partial \acute{w}}{\partial z}\right)
\end{aligned}
\qquad (5.23)
$$

In practical situations, the order of magnitude of the terms, $\partial \rho / \partial x$ and $\partial \rho / \partial z$ are not more than 1 and 10 respectively. Recalling that $\rho \approx 1000$ (kg/m^3), the terms with $(1/\rho^2)$ in the left hand side of equation (5.23) are negligible in comparison with the other terms. Therefore equation (5.23) may be simplified as follows:

$$
\frac{\partial}{\partial x}\left(\frac{\partial \hat{q}}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\partial \hat{q}}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{\partial \hat{q}}{\partial z}\right)=\frac{\rho}{\Delta t}\left(\frac{\partial \acute{u}}{\partial x}+\frac{\partial \acute{v}}{\partial y}+\frac{\partial \acute{w}}{\partial z}\right) \qquad (5.24)
$$

If the atmospheric pressure term and the wind stress are negligible and the free surface slope is sufficiently small, the free surface boundary condition for the hydrodynamic pressure is given as:

$$
q_\eta = \rho \nu_v \frac{\partial w}{\partial z} \qquad (5.25)
$$

If the water level changes are negligible (for example the pure baroclinic conditions), it is usually adequate to assume that the hydrodynamic pressure component at the free surface is negligible (Stansby and Zhou 1998):

$$
q_\eta = 0 \qquad (5.26)
$$

It is also the case when the viscosity is negligible and the flow may be assumed to be inviscid.

The hydrodynamic pressure may be assumed negligible at the lateral open boundaries if the hydrostatic condition is adequate for those areas. If not, the normal gradient of the hydrodynamic pressure may be set to zero.

From the theoretical point of view, the hydrodynamic pressure condition at the closed lateral boundaries and at the bottom boundary may be written as follows:

$$
\frac{\partial q}{\partial \vec{n}} = 0 \qquad (5.27)
$$

where $\vec{n}$ is the direction normal to the boundary, because the velocity field is convergence-free. However, depending on the boundary conditions used for the intermediate velocity field, it may be convergence-free or divergent at the closed boundaries. Therefore equation (5.27) may not be used for the hydrodynamic pressure simulation. The adequate boundary condition for the bed and the lateral closed impermeable boundaries is given as:

$$-\frac{1}{\rho}\frac{\partial q}{\partial n}=\frac{1}{\Delta t}\left(u_{normal}^{n+1}-\grave{u}_{normal}\right)=\frac{1}{\Delta t}\left(0-\grave{u}_{normal}\right) \quad \rightarrow \quad \frac{\partial q}{\partial n}=\frac{\rho}{\Delta t}\grave{u}_{normal} \tag{5.28}$$

where $\grave{u}_{normal}$ is the normal component of the intermediate velocity close to the wall.

When the intermediate velocity field is simulated by the convergence-free restrictions near the walls the right hand side of equation (5.28) is zero and equation (5.27) is true. The hydrodynamic pressure correction may be simulated in many cases by zero Neumann boundary conditions at the solid and open boundaries and zero Dirichlet boundary conditions at the free surface (Gresho 1990, Jankowski 1999).

Equation (5.24) has been solved in the present study in order to define the hydrodynamic pressure correction field.

5.2.2 Salinity Equation

The 3D conservation equation for the salinity follows the general form of the advection-diffusion equation as follows:

$$\frac{\partial S}{\partial t}+u\frac{\partial S}{\partial x}+v\frac{\partial S}{\partial y}+w\frac{\partial S}{\partial z}=\frac{\partial}{\partial x}\left(D_h\frac{\partial S}{\partial x}\right)+\frac{\partial}{\partial y}\left(D_h\frac{\partial S}{\partial y}\right)+\frac{\partial}{\partial z}\left(D_v\frac{\partial S}{\partial z}\right)+S_{dis}Q_{dis} \tag{5.29}$$

where S_{dis} is the salinity of the source and D_h and D_v are the horizontal and vertical diffusion coefficients.

The salinity diffusion coefficients may be linked to the momentum diffusion coefficients as follows:

$$D_h=\frac{\nu_h}{P_r} \quad \text{and} \quad D_v=\frac{\nu_v}{P_r} \tag{5.30}$$

where P_r is the Prandtl number.

The bottom boundary condition for the salinity equation is given as:

$$\frac{\partial S_b}{\partial z}=0 \tag{5.31}$$

where S_b is the near bed salinity of the water.

The surface boundary condition for the salinity equation is (Steinhorn, 1991):

$$D_v\frac{\partial S_\eta}{\partial z}=\frac{1}{\rho}\frac{S_\eta(E_{vap}-R_{pr})}{1-0.001S_s} \tag{5.32}$$

where S_η is the water surface salinity measured in (PSU), E_{vap} is the evaporation rate in $kg\,m^{-2}s^{-1}$, and R_{pr} is the precipitation rate in $kg\,m^{-2}s^{-1}$.

The salinity gradient normal to the boundary assumed to be zero at the closed lateral boundaries and also at the outflow open boundaries. The value of the salinity must be known at the inflow open boundary.

5.2.3 The Temperature Equation

The 3D advection-diffusion equation for the temperature is given as:

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}+w\frac{\partial T}{\partial z}=\frac{\partial}{\partial x}\left(D_h\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(D_h\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(D_v\frac{\partial T}{\partial z}\right)+T_{dis}\,Q_{dis}+A_{heat}^{rad} \tag{5.33}$$

where T_{dis} is the temperature of the source and A_{heat}^{rad} is the heat exchange with the atmosphere (see Eq. 5.55).

The temperature diffusion coefficients may be linked to the momentum diffusion coefficients as follows:

$$D_h=\frac{\nu_h}{S_{ch}} \quad \text{and} \quad D_v=\frac{\nu_v}{S_{ch}} \tag{5.34}$$

where S_{ch} is the Schmidt number.

The bottom boundary condition is:

$$\frac{\partial T_b}{\partial z}=0 \tag{5.35}$$

The surface boundary condition is:

$$D_v\frac{\partial T_\eta}{\partial z}=\frac{Q_{heat}^{surf}}{4217\cdot\rho}+T_p\hat{P}+T_e E_{heat} \tag{5.36}$$

where Q_{heat}^{surf} is the surface net heat flux and E_{heat} is the evaporation effect as follows:

$$E_{heat}=\begin{Bmatrix}\dfrac{q_{heat}^{vap}}{2.5\cdot10^6\cdot\rho} & q_{heat}^{vap}\geqslant0\\ 0 & q_{heat}^{vap}\leqslant0\end{Bmatrix} \tag{5.37}$$

in which, q_{heat}^{vap} is the heat flux due to the water vaporization.

The surface net heat exchange Q_{heat}^{surf} may be estimated as follows:

$$Q_{heat}^{surf}=q_{heat}^{vap}+q_{heat}^{conv}+\beta_{heat}^{infraRed}\,q_{heat}^{shortRad}+q_{heat}^{longRad} \tag{5.38}$$

where q_{heat}^{conv} is the heat flux due to the convection effect (see Eq. 5.40), $q_{heat}^{shortRad}$ is the net heat loss due to the short wave radiation (see Eq. 5.41), $\beta_{heat}^{infraRed}$ is the fraction of infra-red energy which is absorbed near the water surface (usually between 0.2 to 0.6), and $q_{heat}^{longRad}$ is the net heat loss due to the long wave radiation (see Eq. 5.53).

The value of q_{heat}^{vap} in equation (5.38) is defined by the following relationship (Mike3, 2009):

$$q_{heat}^{vap} = -4370\left(0.5+0.9\,W_{2\mathrm{m}}\right) \left(\frac{\exp\left(5418\left(\frac{1}{273.15}-\frac{1}{T_{water}+273.15}\right)\right)}{T_{water}+273.15} - \frac{\exp\left(5418\left(\frac{1}{273.15}-\frac{1}{T_{air}+273.15}\right)\right)}{T_{air}+273.15} \right) \quad (5.39)$$

where $W_{2\mathrm{m}}$ is the wind speed at two meters above the water surface, T_{water} is the water temperature and T_{air} is the air temperature.

The value of q_{heat}^{conv} in equation (5.38) is defined by the following relationship (Kantha Clyson, 2000):

$$q_{heat}^{conv} = \left\{ \begin{array}{ll} \rho_{air}\, c_{air}\, c_{heat}^{conv}\, W_{10\mathrm{m}}\left(T_{water}-T_{air}\right) & T_{air} \geqslant T_{water} \\ \rho_{air}\, c_{water}\, c_{heat}^{conv}\, W_{10\mathrm{m}}\left(T_{water}-T_{air}\right) & T_{air} < T_{water} \end{array} \right\} \quad (5.40)$$

where ρ_{air} is the air density (kg/m^3), c_{air} is the specific heat of the air (1007 J/kg °K), c_{water} is the specific heat of the water (4186 J/kg °K), c_{heat}^{conv} is the convection heat transfer coefficient (1.41x10$^{-3)}$, and $W_{10\mathrm{m}}$ is the wind speed at 10 meters above the water surface.

The value of $q_{heat}^{shortRad}$ (W/m^2) in equation (5.38) is defined by the following relationship:

$$q_{heat}^{shortRad} = \left(1-\alpha_{heat}^{refl}\right) q_{hourly}^{shortRad} \frac{10^6}{360} \quad (5.41)$$

where $q_{hourly}^{shortRad}$ is the hourly-averaged short radiation (see Eq. 5.48) and α_{heat}^{refl} is the reflection coefficient:

$$\alpha_{heat}^{refl} = \left\{ \begin{array}{cc} \frac{altitude}{5} 0.48 & altitude < 5 \\ \frac{30-altitude}{25}\left(0.48-0.05\right) & 5 \leqslant altitude \leqslant 30 \\ 0.05 & altitude > 30 \end{array} \right\} \quad (5.42)$$

The value of $altitude$ in equation (5.42) is given as:

$$altitude = 90 - \left(\frac{180}{\pi} \arccos\left(\sin\left(\delta_{dec}^{ang}\right) \sin\left(\phi_l\right) + \cos\left(\delta_{dec}^{ang}\right) \cos\left(\phi_l\right) \cos\left(\omega_{hr}\right) \right) \right) \quad (5.43)$$

where ϕ_l is the latitude (positive for the northern hemisphere), ω_{hr} is the hour angle, and δ_{dec}^{ang} is the declination angle measured in radian.

The value of ω_{hr} in equation (5.43) is defined by the following relationship:

$$\omega_{hr} = \frac{\pi}{12}\left(12 + \Delta t_{summer} + \frac{4}{60}\left(L_S^{tz} - L_l^{tz}\right) - \frac{E_{time}}{60} - t_{local} \right) \quad (5.44)$$

where Δt_{summer} is the time-shift due to the summer time, L_s^{tz} is the standard longitude for the time zone, L_l^{tz} is the local longitude (in degrees), t_{local} is the local time in hours, and E_{time} is the discrepancy in time due to the solar orbit (Iqbal, 1983; Mike3, 2009):

$$\begin{aligned}E_{time}= & \\ & 229.18(0.000075) \\ & +229.18(0.001868\cos(\Gamma_{day})-0.032077\sin(\Gamma_{day})) \\ & -229.18(0.014615\cos(2\Gamma_{day})+0.04089\sin(2\Gamma_{day}))\end{aligned} \tag{5.45}$$

where Γ_{day} is the day angle measured in radian. It is defined by the following relationship:

$$\Gamma_{day}=\frac{2\pi(d_{Julian}-1)}{360} \tag{5.46}$$

where d_{Julian} is the Julian day of the year.

The value of δ_{dec}^{ang} in equation (5.43) is:

$$\begin{aligned}\delta_{dec}^{ang}= & 0.006918-0.399912\cos(\Gamma_{day})+0.07257\sin(\Gamma_{day})-0.006758\cos(2\Gamma_{day}) \\ & +0.000907\sin(2\Gamma_{day})-0.002697\cos(3\Gamma_{day})+0.00148\sin(3\Gamma_{day})\end{aligned} \tag{5.47}$$

The value of $q_{hourly}^{shortRad}$ in equation (5.41) is the amount of the average hourly short radiation. It is defined by the following relationship:

$$\begin{aligned}q_{hourly}^{shortRad}= & 4.9212\ E_{ec}\left(\sin(\phi_l)\sin(\delta_{dec}^{ang})+\frac{24}{\pi}\cos(\phi_l)\cos(\delta_{dec}^{ang})\cos(\omega_{hr})\right) \\ & \left(a_2+b_2\frac{n_{sunRise}}{n_{sunRise}^{max}}\right)(a_3+b_3\cos(\omega_{hr}))\end{aligned} \tag{5.48}$$

where $n_{sunRise}$ is the number of sunrise hours, $n_{sunRise}^{max}$ is the maximum number of sunrise hours, a_2 and b_2 are calibration factors for the daily radiation under the cloudy conditions (their default values are 0.295 and 0.371 respectively). E_{ec} is the eccentricity in the solar orbit:

$$\begin{aligned}E_{ec}= & 1.000110+0.034221\cos(\Gamma_{day}) \\ & +0.001280\sin(\Gamma_{day})+0.000719\cos(2\Gamma_{day})+0.000077\sin(2\Gamma_{day})\end{aligned} \tag{5.49}$$

The values of a_3 and b_3 in equation (5.48) are defined as follows:

$$a_3=0.4090+0.5016\ \sin\left(\omega_{sr}-\frac{\pi}{3}\right) \tag{5.50}$$

$$b_3=0.6609+0.4767\ \sin\left(\omega_{sr}-\frac{\pi}{3}\right) \tag{5.51}$$

where the sunrise angle $\omega_{ang}^{sunRise}$ (rad) is given by:

$$\omega_{ang}^{sunRise}=\arccos(-\tan(\phi_l)\tan(\delta_{dec}^{ang})) \tag{5.52}$$

The value of $q_{heat}^{longRad}$ in equation (5.38) is defined as follows:

$$q_{heat}^{longRad} = -5.6697\ 10^{-8}\left(T_{air}+273.15\right)^{4}\left(0.56-0.077\sqrt{10\,R_h\cdot e_{sat}^{vap}}\right)\left(0.1+0.9\frac{n_{sunRise}}{n_{sunRise}^{max}}\right) \quad (5.53)$$

where R_h is the relative humidity and e_{sat}^{vap} is the saturated vapour pressure (kPa). The saturated vapour pressure is estimated by the following relationship:

$$e_{sat}^{vap} = 3.38639\left(\left(7.38\ 10^{-3}\,T_{air}+0.8072\right)^{8}-1.9\ 10^{-5}\left(\left|1.8\ T_{air}+48\right|\right)+1.316\ 10^{-3}\right) \quad (5.54)$$

The heat exchange with the atmosphere A_{heat}^{rad} in equation (5.33) is defined as follows:

$$A_{heat}^{rad} = \frac{q_{sr,net}\left(1-\beta_{heat}^{infraRed}\right)\dfrac{e^{-\lambda_l(\eta-z)}}{\lambda_l}}{4217\ \rho} \quad (5.55)$$

where λ_l is the light extinction coefficient (usually between 0.5 to 1.4 m^{-1})

The temperature gradient normal to the closed lateral boundary and the bed may be set to zero. The same condition may be used for the outflow open boundaries. The value of the temperature must be known at the inflow open boundaries.

5.2.4 Suspended Sediment Equation

The 3D version of the suspended sediment concentration equation is:

$$\frac{\partial c}{\partial t}+u\frac{\partial c}{\partial x}+v\frac{\partial c}{\partial y}+w\frac{\partial c}{\partial z}-\frac{\partial w_s c}{\partial z}=\frac{\partial}{\partial x}\left(D_h\frac{\partial c}{\partial x}\right)+\frac{\partial}{\partial y}\left(D_h\frac{\partial c}{\partial y}\right)+\frac{\partial}{\partial z}\left(D_v\frac{\partial c}{\partial z}\right)+\text{S.T.} \quad (5.56)$$

where w_s is the sediment settling velocity and "S.T." includes the source terms.

Equation (5.56) is solved numerically for one or more fractions of sediment and/or mud classes. Each fraction has specific settling velocity and source terms. The classification of the suspended sediment fractions is based on the grain size. The mud itself involves silt (grain diameter ranges from 4 to 63 μm) and clay (grain diameter is less than 4 μm) grain sizes.

The settling velocity w_s is defined differently for sand and mud materials. For the grain size greater than or equal to 63 μm (sand), the following relationship may be used (van Rijn, 1987):

$$w_{s,i}=\left\{\begin{array}{cc} \dfrac{\rho_s-\rho_w}{\rho_w}\dfrac{g d_i^2}{18\nu} & d_i<100\,\mu m \\ \dfrac{10\nu}{d_i}\left\{\left[1+\dfrac{\rho_s-\rho_w}{\rho_w}\dfrac{0.01 g d_i^3}{\nu^2}\right]^{0.5}-1\right\} & 100<d_i\leq 1000\,\mu m \\ 1.1\left(\dfrac{\rho_s-\rho_w}{\rho_w} g d_i\right)^{0.5} & d_i>1000\,\mu m \end{array}\right\} \qquad (5.57)$$

where i is the fraction index, d is the grain size, and ν is the water viscosity.

For grain size less than 63 μm (mud), the settling velocity is affected by flocculation, hindered settling, and the gelling effects. Therefore it is a function of the suspended sediment concentration. The following relationship may be used for the salt water:

$$w_{s,i}=\left\{\begin{array}{cc} w_{s,i,cons} & \left(\sum_i c_i\right)\leq c_{floc} \\ w_{s,i,cons}\dfrac{\left(\sum_i c_i\right)}{C_{floc}} & c_{floc}<\left(\sum_i c_i\right)\leq c_{hind} \\ w_{s,i,const}\dfrac{c_{hind}c_{gel}}{c_{floc}\left(c_{gel}-c_{hind}\right)}\left(1-min\left(1.0,\dfrac{\left(\sum_i c_i\right)}{c_{gel}}\right)\right) & c_{hind}<\left(\sum_i c_i\right)\leq c_{gel} \end{array}\right\} \qquad (5.58)$$

where c_{floc} is the concentration at which flocculation starts, c_{hind} is the concentration at which hindered settling starts, c_{gel} is the concentration at which gelling process occurs, c_i is the concentration of the ith sediment fraction, and $w_{s,i,const}$ is the constant settling velocity for the ith sediment fraction when the concentration is sufficiently low and no one of the mentioned processes are effective.

The recommended values for c_{floc}, c_{hind} and c_{gel} are 0.01, 10, and 50 kg/m^3 respectively. These parameters are relevant to the total suspended sediment concentration (the sum of all suspended sediment fractions). The value of $w_{s,i,const}$ may be defined by field or laboratory measurements. It may also be considered as a calibration coefficient. Equation (5.57) may be used as the first estimation for $w_{s,i,const}$ when there is not any measurement in hand.

The flocculation process for the mud material is effective both in the fresh water due to the organic matter content and in the saline water where the salt flocculation process is also active.

In situations where the fresh water meets the salt water, the salinity depression slows down the flocculation process and the settling velocity. The formulation is (Krone 1962, Mike3MT 2005):

$$w_{s,i} = w_{s,i,salt\,water}\left(1 - 0.5\,e^{-0.33\,S}\right) \tag{5.59}$$

where $w_{s,i,saline\,water}$ is the output of equation (5.58). Equation (5.59) is applicable for situations where the salinity is less than 10 PSU. The flocculation process is not influenced by the salinity value when it is more than 10 PSU.

Equation (5.56) needs boundary conditions. The free surface boundary condition for the ith sediment fraction is:

$$w_{s,i}\,c_{i,\eta} + D_v\left(\frac{\partial c_i}{\partial z}\right)_{z=\eta} = 0 \tag{5.60}$$

which means that the net vertical sediment transport at the water surface is zero.

The bottom boundary condition for the sediment fractions with grain sizes less than 63 μm (mud) is (Han and He, 1997; Fang and Wang, 2000):

$$w_{s,i}\,c_{i,b} + D_v\left(\frac{\partial c_i}{\partial z}\right)_b = Deposition_i - Erosion_i \tag{5.61}$$

which means that the net vertical sediment transport at the bed is equal to the deposition rate minus the bed-erosion rate. The horizontal dispersion effect has been neglected in this regard.

Positive values are not realistic for the vertical gradient of the suspended sediment concentration near the bed in usual cases. In these cases if for any numerical reason (for example using an explicit algorithm for a rapidly varying flow field) the bottom boundary condition results in the positive gradient, then equation (5.61) has to be solved again by $\partial c_i / \partial z = 0$. A simpler approach for the cohesive sediment transport that may be applied in simple cases is to assume that $\partial c_i / \partial z = 0$ as the bottom boundary condition and handle the deposition and the bed erosion quantities as the sinks and sources for the bottom boundary nodes.

The deposition rate for each fraction of the suspended sediment in equation (5.61) may be defined by the following relationship (Krone, 1962):

$$Deposition_i = w_{s,i}\,c_{i,b}\,p_{D,i} \tag{5.62}$$

where the index i is the fraction number, $c_{i,b}$ is the suspended sediment concentration for the ith fraction close to the bed, $w_{s,i}$ is the settling velocity for the ith fraction, and $p_{dep,i}$ is the probability function.

Te probability function may be defined as follows:

$$p_{dep,i}=max\left(0,min\left(1,1-\frac{\tau_b}{\tau_{cr,d,i}}\right)\right) \tag{5.63}$$

where τ_b is the bed shear stress and $\tau_{cr,d,i}$ is the critical bed shear stress for deposition of the ith fraction. No deposition occurs when τ_b is more than the critical value $\tau_{cr,d,i}$.

The bed material generally involves different grain sizes. The eroded material includes all of the sediment fractions. The total erosion rate depends on the bed density and the bed consolidation. The bed may include more than one layer, each one with specific density, consolidation and critical shear stress. Erosion occurs at the active layer on the top. When it is eroded completely, the next layer becomes active. The erosion rate is different for the consolidated (hard) bed and the un-consolidated (soft) bed conditions. The water or moisture content in the hard bed is below 100% while it is more than 100% in the soft bed material.

The total erosion rate for the hard bed layer is defined by the following relationship (Mehta et al, 1989):

$$E_{j,total}=\alpha_{h,j}^{erosion}\ p_{E,j} \tag{5.64}$$

where j is the active bed layer, $E_{j,total}$ is the erosion rate of the jth bed layer, $\alpha_{h,j}^{erosion}$ is the calibration factor for the hard bed erosion, and $p_{erosion,j}$ is the probability function.

The probability function is defined as follows:

$$p_{erosion,j}=max\left(0,\left(\frac{\tau_b}{\tau_{c,er,j}}-1\right)\right) \tag{5.65}$$

where $\tau_{c,er,j}$ is the critical shear stress for erosion for the jth bed layer.

The total erosion rate for the soft bed layer is defined as follows (Parchure and Mehta, 1985; Mehta et al, 1989):

$$E_{j,total}=E_{0,j}\exp\left(\alpha_s^{erosion}\sqrt{\tau_b-\tau_{c,er,j}}\right) \tag{5.66}$$

where $E_{0,j}$ and $\alpha_s^{erosion}$ are calibration factors.

The relationship between the bed erosion rate ($E_{j,total}$) and the erosion rate of each sediment size fraction ($Erosion_i$) is based on the percentage of each suspended sediment fraction (grain size) in the bed material. This data must be known at the start of the simulation as part of the initial condition. The variation of the bed gradation with time is calculated through the simulation.

The bottom boundary condition for the suspended sediment concentration is different from equation (5.61) if the suspended sediment grain size is 63 to 125 μm (fine sand). The following empirical relationship may be used in this regard (Zyserman and Fredsoe, 1994):

$$c_b = \frac{0.331\left(\theta^{sh} - \theta_c^{sh}\right)^{1.75}}{1 + \frac{0.331}{0.46}\left(\theta^{sh} - \theta_c^{sh}\right)^{1.75}} \qquad (5.67)$$

where c_b is the near-bed suspended sediment concentration, θ^{sh} is the shields parameter and θ_c^{sh} is the critical shields parameter.

The shields parameter is:

$$\theta^{sh} = \frac{\tau_b}{\rho_w g d_{50} \frac{\rho_s - \rho_w}{\rho_w}} \qquad (5.68)$$

The bottom boundary condition for the fine sand (63 to 125 μm) is more complicated in situations where the bed material includes both mud and fine sand. This condition has not been considered in the present study.

The suspended sediment concentration gradient normal to the lateral closed boundary may be set to zero. The same assumption may be used for the outflow open boundary. The concentration must be known at the inflow open boundary.

5.2.5 Turbulence Equations

The eddy viscosity in the hydrodynamic equations stands for the turbulence effect and any other sub-grid scale mixing phenomena. The behaviour of the small eddies (compared to the water depth) is isotropic in the 3D space while the behaviour of the larger eddies are anisotropic and restricted to the horizontal plane. The 3D small eddies mainly influence the vertical mixing while the 2D large eddies mainly influence the horizontal diffusion. The 3D small eddies have not considerable effect on the horizontal diffusion in the momentum equations because the horizontal grid space is usually some order of magnitudes larger than the vertical grid space and the water depth.

The 3D $k-\epsilon$ equations are solved in the present study for the small-scale isotropic turbulence in order to define the vertical eddy viscosity. It is also possible to choose the log-law relationship which gives the standard parabolic profile (see Eq. 2.67). The log-law relationship is sufficient and more efficient in many practical cases.

The $k-\epsilon$ model for the vertical eddy viscosity is different from the general form of the 3D $k-\epsilon$ model. The production terms in the k and ε equations involve just the vertical gradient of the horizontal velocities without taking the horizontal gradients into account. The 3D $k-\epsilon$ equations for the vertical eddy viscosity are:

$$\frac{\partial k}{\partial t}+\overbrace{u\frac{\partial k}{\partial x}+v\frac{\partial k}{\partial y}+w\frac{\partial k}{\partial z}}^{1}=\overbrace{\frac{\partial}{\partial x}\left(\frac{\nu_h}{\sigma_k}\frac{\partial k}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\nu_h}{\sigma_k}\frac{\partial k}{\partial y}\right)}^{2}+\overbrace{\frac{\partial}{\partial z}\left(\frac{\nu_v}{\sigma_k}\frac{\partial k}{\partial z}\right)}^{3}$$
$$\underbrace{+\nu_v\left(\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2\right)}_{4}+\underbrace{\frac{g}{\rho}\frac{\nu_v}{\sigma_t}\frac{\partial \rho}{\partial z}}_{5}-\varepsilon \tag{5.69}$$

$$\frac{\partial \varepsilon}{\partial t}+\overbrace{u\frac{\partial \varepsilon}{\partial x}+v\frac{\partial \varepsilon}{\partial y}+w\frac{\partial \varepsilon}{\partial z}}^{1}=\overbrace{\frac{\partial}{\partial x}\left(\frac{\nu_h}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\nu_h}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial y}\right)}^{2}+\overbrace{\frac{\partial}{\partial z}\left(\frac{\nu_v}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial z}\right)}^{3} \tag{5.70}$$
$$+c_{1\varepsilon}\frac{\varepsilon}{k}\underbrace{\nu_v\left(\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2\right)}_{4}+c_{3\varepsilon}\frac{\varepsilon}{k}\underbrace{\frac{g}{\rho}\frac{\nu_v}{\sigma_t}\frac{\partial \rho}{\partial z}}_{5}-c_{2\varepsilon}\frac{\varepsilon^2}{k}$$

The values of k and ϵ in equations (5.69) and (5.70) are the turbulence kinetic energy and its dissipation rate respectively. The terms (1) to (5) are the advection, the horizontal diffusion, the vertical diffusion, the production and the buoyancy effect respectively.

The bottom boundary-conditions for k and ϵ are based on the wall function theory (Rodi, 1984):

$$k_b=\frac{\frac{\tau_b}{\rho}}{\sqrt{c_\mu}}=\frac{1}{\sqrt{c_\mu}}\frac{\left|\overrightarrow{V_{do}}\right|^2}{\left(\frac{1}{\kappa}\ln\left(\frac{d_o}{(k_s/30)}\right)\right)^2} \tag{5.71}$$

$$\epsilon_b=\frac{\left(\frac{\tau_b}{\rho}\right)^{3/2}}{\kappa d_o}=\frac{1}{\kappa d_o}\left(\frac{\left|\overrightarrow{V_{do}}\right|}{\left(\frac{1}{\kappa}\ln\left(\frac{d_o}{(k_s/30)}\right)\right)}\right)^{3/2} \tag{5.72}$$

where $\kappa=0.41$ is the von Karman's constant, k_s is the bed roughness height, d_o is the normal distance between the boundary point and the bed, and $\left|\overrightarrow{V_{do}}\right|$ is the magnitude of the velocity at the boundary point.

The free surface boundary conditions for k and ϵ depend on the wind shear stress:

$$k_\eta=\frac{1}{\sqrt{c_\mu}}\frac{\tau_s}{\rho}=\frac{1}{\sqrt{c_\mu}}\frac{\rho_{air}C_w W^2}{\rho} \tag{5.73}$$

$$\epsilon_\eta=\frac{\left(\frac{\tau_b}{\rho}\right)^{3/2}}{\kappa d_{o,\eta}}=\frac{1}{\kappa d_{o,\eta}}\left(\frac{\rho_{air}C_w W^2}{\rho}\right)^{3/2} \tag{5.74}$$

where W is the wind speed relative to the water velocity at the free surface, C_w is the wind drag coefficient as illustrated in equation (2.26), and $d_{o,\eta}$ is the distance between the top layer and the real free surface.

In situations where the wind shear stress is negligible, the following relationship may be used (Rodi, 1984):

$$\left(\frac{\partial k}{\partial z}\right)_\eta=0 \tag{5.75}$$

$$\epsilon_\eta=\frac{\left(k_\eta\sqrt{c_\mu}\right)^{3/2}}{0.07\kappa\left(\eta-z_b\right)} \tag{5.76}$$

where $\left(\eta-z_b\right)$ is the water depth.

The horizontal eddy viscosity usually needs calibration in the practical problems. The Smagorinsky formulation has been used for the horizontal eddy viscosity. The relationship is:

$$\nu_h=c_{sm}l_{sm}^2\sqrt{\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2+\frac{1}{2}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2} \tag{5.77}$$

where c_{sm} is the calibration factor and l_{sm} is the length scale. The area of the triangular elements has been used for l_{sm}^2 in the present study.

5.2.6 Water Density

The value of the water density ρ depends on the temperature, the salinity and the suspended sediment concentration. The UNESCO equation of state (EOS–80) determines the effect of the salinity and the temperature in this regard. It is valid for $T\in[-2,40]$ ºC and $S\in[0,42]$ psu. The present model calculates the value of ρ in three steps as follows:

Step 1- The reference density of the pure water is computed as a function of the temperature:

$$\rho_{pw}(T)=a_0+a_1T+a_2T^2+a_3T^3+a_4T^4+a_5T^5 \tag{5.78}$$

where ρ_{pw} is the pure water density in (kg/m^3) and T is the water temperature in ºC. The relevant coefficients are:

a_0 = + 999.842594

a_1 = + 6.793952e-2

a_2 = - 9.095290e-3

a_3 = + 1.001685e-4

a_4 = - 1.120083e-6

a_5 = + 6.536332e-9

Step 2- The effect of the salinity is added to the density by the following relationship:

$$\rho(T,S)=\rho_{pw}(T)+(b_0+b_1T+b_2T^2+b_3T^3+b_4T^4)S+(c_0+c_1T+c_2T^2)S^{3/2}+d_0S^2 \quad (5.79)$$

where $\rho(T,S)$ is the clean water density measured in (kg/m^3) and S is the salinity measured in (psu). The unit (psu) is almost identical with (ppt) for practical purposes (parts per thousand). The relevant coefficients in equation (5.79) are:

b_0 = + 8.24493e-1

b_1 = - 4.0899e-3

b_2 = + 7.6438e-5

b_3 = - 8.2467e-7

b_4 = + 5.3875e-9

c_0 = - 5.72466e-3

c_1 = + 1.0227e-4

c_2 = - 1.6546e-6

d_0 = + 4.8314e-4

Equation (5.79) is adequate for the standard atmospheric pressure condition at the free surface.

Step 3- The effect of the suspended sediment concentration on the water density is taken into account by the following relationship:

$$\rho(T,S,c)=\rho(T,S)+\frac{\rho_{sed}-\rho(T,S)}{\rho_{sed}}c \quad (5.80)$$

where ρ_{sed} is the sediment constituent density in (kg/m^3) and c is the dry mass suspended sediment concentration in (kg/m^3).

When there are more than one suspended sediment fractions to be solved, a more general relationship may be used instead of equation (5.80). It is:

$$\rho(T,S,c)=\rho(T,S)+\sum_{i=1}^{N_f}\frac{\rho_{sed}-\rho(T,S)}{\rho_{sed}}c_i \quad (5.81)$$

where i is the fraction number and N_f is the total number of the sediment fractions.

5.3 Transformation and the Transformed Equations

5.3.1 Transformation

The 3D governing equations may be transformed from the Cartesian coordinate system (t, x, y, z) into the vertical boundary fitted coordinate system $(\tilde{t}, \tilde{x}, \tilde{y}, \tilde{z})$ in order to simplify parts of the solution procedure including the water level variability. Several transformation approaches are available in this regard. The present study is based on an extension of the σ-coordinate system called the vertical gradient adaptive σ-coordinate transformation system (GAS). This method was proposed by Wai and Lu, 1999. The GAS transformation system takes into account the vertical gradient of a selected variable (for example the sediment concentration or the temperature) as a control factor in order to adjust the vertical grid spacing automatically during the simulation procedure.

The transformation rules for the GAS coordinate system are (Wai and Lu, 1999):

$$\begin{aligned}
&\tilde{t}=t\\
&\tilde{x}=x\\
&\tilde{y}=y\\
&\tilde{z}=\frac{(z-z_b)+a_{GAS}}{(\eta-z_b)+b_{GAS}} \qquad z_{GAS}\in[0,1]\\
&a_{GAS}=\int_{z_b}^{z}\left(\beta_{GAS}\left|\frac{\partial\phi}{\partial z}\right|+\gamma_{GAS}\left|\frac{\partial^2\phi}{\partial z^2}\right|\right)dz\\
&b_{GAS}=\int_{z_b}^{\eta}\left(\beta_{GAS}\left|\frac{\partial\phi}{\partial z}\right|+\gamma_{GAS}\left|\frac{\partial^2\phi}{\partial z^2}\right|\right)dz
\end{aligned} \tag{5.82}$$

where β_{GAS} and γ_{GAS} are user specified coefficients which control the influence of the variable ϕ on the vertical grid distribution. The control variable ϕ can be the suspended sediment concentration, the velocity, the temperature, the salinity, or any other interested quantity.

The rules for the GAS transformation (5.82) are similar to their counterparts for the standard σ-coordinate system, except for the coefficients a_{GAS} and b_{GAS}. The transformed equations in the GAS-coordinate system are also similar to the transformed equations in the σ-coordinate system. The GAS transformation is exactly identical to the σ transformation if the constants β_{GAS} and γ_{GAS} are set to zero.

The factors β_{GAS} and γ_{GAS} must be selected carefully. The grid skewness becomes too significant if one of the factors is set unreasonably large. This may cause numerical instability. For estuary and coastal water simulation, the optimal value for β_{GAS}, subject to a settling term, may be chosen between 1 and 5. The optimal value for β_{GAS}, subject to dissolved substances, should be in the range of 0.1 to 1. The value of γ_{GAS} should be 1/10 of β_{GAS}.

The partial derivatives must be transformed from the Cartesian coordinate system (x,y,z,t) to the GAS coordinate system $(\tilde{x},\tilde{y},\tilde{z},\tilde{t})$. The transformation rules are:

$$\frac{\partial}{\partial t}=\frac{\partial}{\partial \tilde{t}}+\frac{\partial \tilde{z}}{\partial t}\frac{\partial}{\partial \tilde{z}} \tag{5.83}$$

$$\frac{\partial}{\partial x}=\frac{\partial}{\partial \tilde{x}}+\frac{\partial \tilde{z}}{\partial x}\frac{\partial}{\partial \tilde{z}} \quad , \quad \frac{\partial}{\partial y}=\frac{\partial}{\partial \tilde{y}}+\frac{\partial \tilde{z}}{\partial y}\frac{\partial}{\partial \tilde{z}} \tag{5.84}$$

$$\frac{\partial}{\partial z}=\frac{\partial \tilde{z}}{\partial z}\frac{\partial}{\partial \tilde{z}} \tag{5.85}$$

The time and z derivatives of z_{GAS} in equations (5.83) to (5.85) are defined as follows:

$$\frac{\partial \tilde{z}}{\partial t}=\frac{1}{H_{GAS}}\left(\frac{\partial a_{GAS}}{\partial t}-\tilde{z}\frac{\partial \eta}{\partial t}-\tilde{z}\frac{\partial b_{GAS}}{\partial t}\right) \tag{5.86}$$

$$\frac{\partial \tilde{z}}{\partial z}=\frac{1}{H_{GAS}} \tag{5.87}$$

where the value H_{GAS} is defined by the following relationship:

$$H_{GAS}=\frac{\eta-z_b+b_{GAS}}{1+\beta_{GAS}\left|\frac{\partial \phi}{\partial z}\right|+\gamma_{GAS}\left|\frac{\partial^2 \phi}{\partial z^2}\right|} \tag{5.88}$$

The horizontal diffusion and the horizontal non-hydrostatic pressure gradient terms are not transformed into the GAS co-ordinate system. The reason has been described in section (4.6).

The vertical velocity component w may be transformed into a new variable $\widetilde{w}$ in order to have simple transformed equations similar to their original counterparts. Two kinds of transformations are common in this regard. One of them is given as (Wai and Lu 1999, Luyten et al. 1999):

$$\widetilde{w}=\frac{D\tilde{z}}{Dt}=\frac{\partial \tilde{z}}{\partial t}+u\frac{\partial \tilde{z}}{\partial x}+v\frac{\partial \tilde{z}}{\partial y}+w\frac{\partial \tilde{z}}{\partial z} \tag{5.89}$$

which links w to $\widetilde{w}$ as follows:

$$w=H_{GAS}\widetilde{w}+\tilde{z}\frac{\partial \eta}{\partial t}+u\left(\tilde{z}\frac{\partial \eta}{\partial x}+(1-\tilde{z})\frac{\partial z_b}{\partial x}\right)+v\left(\tilde{z}\frac{\partial \eta}{\partial y}+(1-\tilde{z})\frac{\partial z_b}{\partial y}\right) \tag{5.90}$$

The other one is the multiplication of equation (5.89) by H_{GAS} which is the water depth in the σ-coordinate system (Stansby 1997, Stansby and Zhou 1998). The first choice has been selected in the present study. The transformed vertical velocity in the governing equations appears in a similar form as the horizontal velocity components when the first choice is used.

5.3.2 Flow Equations

The hydrodynamic equations in the transformed coordinate system are:

$$\frac{\partial H_{GAS}}{\partial t}+\frac{\partial(u\,H_{GAS})}{\partial\tilde{x}}+\frac{\partial(v\,H_{GAS})}{\partial\tilde{y}}+\frac{\partial(\tilde{w}\,H_{GAS})}{\partial\tilde{z}}=H_{GAS}\cdot Q_{dis} \tag{5.91}$$

$$\begin{aligned}
&\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial\tilde{x}}+v\frac{\partial u}{\partial\tilde{y}}+\tilde{w}\frac{\partial u}{\partial\tilde{z}}=\\
&\frac{\partial}{\partial\tilde{x}}\left(\nu_h\frac{\partial u}{\partial x}\right)+\frac{\partial\tilde{z}}{\partial x}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial\tilde{y}}\left(\nu_h\frac{\partial u}{\partial y}\right)+\frac{\partial\tilde{z}}{\partial y}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u}{\partial z}\right)\\
&-\frac{1}{\rho}\frac{\partial p_a}{\partial\tilde{x}}-g\frac{\rho_\eta}{\rho}\frac{\partial\eta}{\partial\tilde{x}}-g\frac{1}{\rho}\int_z^\eta\frac{\partial\rho}{\partial x}dz-\frac{1}{\rho}\left(\frac{\partial q}{\partial\tilde{x}}+\frac{\partial\tilde{z}}{\partial x}\frac{\partial q}{\partial\tilde{z}}\right)\\
&+2\cdot v\cdot\omega_z-2\cdot w\cdot\omega_y+u_{dis}Q_{dis}
\end{aligned} \tag{5.92}$$

$$\begin{aligned}
&\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial\tilde{x}}+v\frac{\partial v}{\partial\tilde{y}}+\tilde{w}\frac{\partial v}{\partial\tilde{z}}=\\
&\frac{\partial}{\partial\tilde{x}}\left(\nu_h\frac{\partial v}{\partial x}\right)+\frac{\partial\tilde{z}}{\partial x}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial\tilde{y}}\left(\nu_h\frac{\partial v}{\partial y}\right)+\frac{\partial\tilde{z}}{\partial y}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial v}{\partial y}\right)+\frac{\partial}{\partial z}\left(\nu_v\frac{\partial v}{\partial z}\right)\\
&-\frac{1}{\rho}\frac{\partial p_a}{\partial\tilde{y}}-g\frac{\rho_\eta}{\rho}\frac{\partial\eta}{\partial\tilde{y}}-g\frac{1}{\rho}\int_z^\eta\frac{\partial\rho}{\partial y}dz-\frac{1}{\rho}\left(\frac{\partial q}{\partial\tilde{y}}+\frac{\partial\tilde{z}}{\partial y}\frac{\partial q}{\partial\tilde{z}}\right)\\
&-2\cdot u\cdot\omega_z+2\cdot w\cdot\omega_x+v_{dis}Q_{dis}
\end{aligned} \tag{5.93}$$

$$\begin{aligned}
&\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial\tilde{x}}+v\frac{\partial w}{\partial\tilde{y}}+\tilde{w}\frac{\partial w}{\partial\tilde{z}}=\\
&\frac{\partial}{\partial\tilde{x}}\left(\nu_h\frac{\partial w}{\partial x}\right)+\frac{\partial\tilde{z}}{\partial x}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial w}{\partial x}\right)+\frac{\partial}{\partial\tilde{y}}\left(\nu_h\frac{\partial w}{\partial y}\right)+\frac{\partial\tilde{z}}{\partial y}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial w}{\partial y}\right)+\frac{\partial}{\partial z}\left(\nu_v\frac{\partial w}{\partial z}\right)\\
&-\frac{1}{H_{GAS}}\frac{1}{\rho}\frac{\partial q}{\partial\tilde{z}}+2\cdot u\cdot\omega_y-2\cdot v\cdot\omega_x+w_{dis}Q_{dis}
\end{aligned} \tag{5.94}$$

The horizontal gradient terms in equations (5.92) to (5.94) have not been transformed completely into the σ-coordinate system. The reason for such transformation was described in section (4.6). The aim is to avoid the artificial velocity and diffusion that could form near the steep bed slope in highly stratified flow regions such as the density current in the reservoir.

The present study has introduced a new approach for discretisation of the horizontal diffusion terms. It computes the shear stresses in the Cartesian coordinate system, and then computes the gradients of the shear stresses in the σ-coordinate system.

The vertical diffusion terms in equations (5.92) to (5.94) have been discretised in the Cartesian coordinate system in order to guaranty the shear stress continuity at the boundary of the vertical elements.

5.3.3 Hydrodynamic Pressure Equation

The relationship for the hydrodynamic pressure correction (5.24) is transformed to the σ-coordinate system as follows:

$$\begin{aligned}&\frac{\partial}{\partial\tilde{x}}\left(\frac{\partial\hat{q}}{\partial\tilde{x}}+\frac{\partial\tilde{z}}{\partial x}\frac{\partial\hat{q}}{\partial\tilde{z}}\right)+\frac{\partial\tilde{z}}{\partial x}\frac{\partial}{\partial\tilde{z}}\left(\frac{\partial\hat{q}}{\partial\tilde{x}}+\frac{\partial\tilde{z}}{\partial x}\frac{\partial\hat{q}}{\partial\tilde{z}}\right)\\&+\frac{\partial}{\partial\tilde{y}}\left(\frac{\partial\hat{q}}{\partial\tilde{y}}+\frac{\partial\tilde{z}}{\partial y}\frac{\partial\hat{q}}{\partial\tilde{z}}\right)+\frac{\partial\tilde{z}}{\partial y}\frac{\partial}{\partial\tilde{z}}\left(\frac{\partial\hat{q}}{\partial\tilde{y}}+\frac{\partial\tilde{z}}{\partial y}\frac{\partial\hat{q}}{\partial\tilde{z}}\right)\\&+\frac{1}{H_{GAS}}\frac{\partial}{\partial\tilde{z}}\left(\frac{1}{H_{GAS}}\frac{\partial\hat{q}}{\partial\tilde{z}}\right)=\frac{\rho}{\Delta t}\left(\frac{\partial\acute{u}}{\partial\tilde{x}}+\frac{\partial\tilde{z}}{\partial x}\frac{\partial\acute{u}}{\partial\tilde{z}}+\frac{\partial\acute{v}}{\partial\tilde{y}}+\frac{\partial\tilde{z}}{\partial y}\frac{\partial\acute{v}}{\partial\tilde{z}}+\frac{1}{H_{GAS}}\frac{\partial\acute{w}}{\partial\tilde{z}}\right)\end{aligned}\tag{5.95}$$

where $\acute{u}$, $\acute{v}$ and $\acute{w}$ are the intermediate non-divergence-free velocity components and $\hat{q}=q^{n+1}-q^{n}$ is the correction to the hydrodynamic pressure.

Equation (5.95) is used to calculate the correction term to the hydrodynamic pressure. It does not provide the total pressure. This approach is consistent with the third technique described in section (4.6) for avoiding the hydrostatic pressure inconsistency when σ-coordinate grid is used.

The second order vertical gradient term in equation (5.95) has been derived as follows:

$$\frac{1}{H_{GAS}}\frac{\partial}{\partial\tilde{z}}\left(\frac{1}{H_{GAS}}\frac{\partial\hat{q}}{\partial\tilde{z}}\right)=\frac{1}{H_{GAS}^{2}}\frac{\partial^{2}\hat{q}}{\partial\tilde{z}^{2}}+\frac{1}{H_{GAS}}\frac{\partial(1/H_{GAS})}{\partial\tilde{z}}\frac{\partial\hat{q}}{\partial\tilde{z}}\tag{5.96}$$

It should be noted that when the σ planes are not distributed homogeneously along the water depth, then:

$$\frac{\partial^{2}}{\partial z^{2}}\neq\frac{1}{H_{GAS}^{2}}\frac{\partial^{2}}{\partial\tilde{z}^{2}}\tag{5.97}$$

5.3.4 Other 3D Governing Equations

The transformed equations for the salinity and temperature are given as:

$$\begin{aligned}&\frac{\partial S}{\partial t}+u\frac{\partial S}{\partial\tilde{x}}+v\frac{\partial S}{\partial\tilde{y}}+\tilde{w}\frac{\partial S}{\partial\tilde{z}}=\\&\frac{\partial}{\partial x}\left(D_{h}\frac{\partial S}{\partial x}\right)+\frac{\partial}{\partial y}\left(D_{h}\frac{\partial S}{\partial y}\right)+\frac{1}{H_{GAS}}\frac{\partial}{\partial\tilde{z}}\left(\frac{D_{v}}{H_{GAS}}\frac{\partial S}{\partial\tilde{z}}\right)+S_{dis}Q_{dis}\end{aligned}\tag{5.98}$$

$$\begin{aligned}&\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial\tilde{x}}+v\frac{\partial T}{\partial\tilde{y}}+\tilde{w}\frac{\partial T}{\partial\tilde{z}}=\\&\frac{\partial}{\partial x}\left(D_{h}\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(D_{h}\frac{\partial T}{\partial y}\right)+\frac{1}{H}\frac{\partial}{\partial\tilde{z}}\left(\frac{D_{v}}{H}\frac{\partial T}{\partial\tilde{z}}\right)+T_{dis}Q_{dis}+A_{heat}^{rad}\end{aligned}\tag{5.99}$$

The transformed equation for the suspended sediment concentration is given as:

$$\begin{aligned}&\frac{\partial c}{\partial t}+u\frac{\partial c}{\partial \tilde{x}}+v\frac{\partial c}{\partial \tilde{y}}+\tilde{w}\frac{\partial c}{\partial \tilde{z}}-\frac{1}{H_{GAS}}\frac{\partial w_s c}{\partial \tilde{z}}=\\&\frac{\partial}{\partial x}\left(D_h\frac{\partial c}{\partial x}\right)+\frac{\partial}{\partial y}\left(D_h\frac{\partial c}{\partial y}\right)+\frac{1}{H_{GAS}}\frac{\partial}{\partial \tilde{z}}\left(\frac{D_v}{H_{GAS}}\frac{\partial c}{\partial \tilde{z}}\right)+\text{S.T.}\end{aligned} \tag{5.100}$$

The transformed equations for the vertical turbulence are given as:

$$\begin{aligned}&\frac{\partial k}{\partial t}+u\frac{\partial k}{\partial x}+v\frac{\partial k}{\partial y}+\tilde{w}\frac{\partial k}{\partial \tilde{z}}=\frac{\partial}{\partial x}\left(\frac{\nu_h}{\sigma_k}\frac{\partial k}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\nu_h}{\sigma_k}\frac{\partial k}{\partial y}\right)\\&+\frac{1}{H_{GAS}}\frac{\partial}{\partial \tilde{z}}\left(\frac{\nu_v}{\sigma_k}\frac{1}{H_{GAS}}\frac{\partial k}{\partial \tilde{z}}\right)+\underbrace{\nu_v\left(\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2\right)}_{production}+\frac{g}{\rho}\frac{\nu_v}{\sigma_t}\frac{\partial \rho}{\partial z}-\varepsilon\end{aligned} \tag{5.101}$$

$$\begin{aligned}&\frac{\partial \varepsilon}{\partial t}+u\frac{\partial \varepsilon}{\partial x}+v\frac{\partial \varepsilon}{\partial y}+\tilde{w}\frac{\partial \varepsilon}{\partial \tilde{z}}=\frac{\partial}{\partial x}\left(\frac{\nu_h}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\nu_h}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial y}\right)\\&+\frac{1}{H_{GAS}}\frac{\partial}{\partial \tilde{z}}\left(\frac{\nu_v}{\sigma_\varepsilon}\frac{1}{H_{GAS}}\frac{\partial \varepsilon}{\partial \tilde{z}}\right)+c_{1\varepsilon}\frac{\varepsilon}{k}\underbrace{\nu_v\left(\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2\right)}_{production}+c_{3\varepsilon}\frac{\varepsilon}{k}\frac{g}{\rho}\frac{\nu_v}{\sigma_t}\frac{\partial \rho}{\partial z}-c_{2\varepsilon}\frac{\varepsilon^2}{k}\end{aligned} \tag{5.102}$$

5.4 2D Governing Equations

The governing equations for the 2D depth averaged hydrodynamic model are:

$$\frac{\partial(\eta-z_b)}{\partial t}+\frac{\partial(DU)}{\partial x}+\frac{\partial(DV)}{\partial y}=DQ_{dis} \tag{5.103}$$

$$\begin{aligned}&\frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+V\frac{\partial U}{\partial y}=fV-g\frac{\partial \eta}{\partial x}+\frac{\tau_{sx}}{\rho D}-\frac{gU\sqrt{U^2+V^2}}{DC_{ch}}\\&-\frac{1}{\rho}\frac{\partial p_a}{\partial x}-\frac{gD}{2\rho}\frac{\partial \rho}{\partial x}+\bar{\nu}_t\left(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}\right)+(u_s-U)Q_{dis}\end{aligned} \tag{5.104}$$

$$\begin{aligned}&\frac{\partial V}{\partial t}+U\frac{\partial V}{\partial x}+V\frac{\partial V}{\partial y}=-fU-g\frac{\partial \eta}{\partial y}+\frac{\tau_{sy}}{\rho D}-\frac{gV\sqrt{U^2+V^2}}{DC_{ch}}\\&-\frac{1}{\rho}\frac{\partial p_a}{\partial y}-\frac{gD}{2\rho}\frac{\partial \rho}{\partial y}+\bar{\nu}_t\left(\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2}\right)+(v_s-V)Q_{dis}\end{aligned} \tag{5.105}$$

where $D=\eta-z_b$ is the total water depth, C_{ch} is the Chezy coefficient for the bed roughness, τ_s is the wind shear stress, ρ is the depth averaged water density, f is the Coriolis factor, and U and V are the depth averaged velocities defined by (5.106):

$$U=\frac{1}{D}\int_{z_b}^{\eta}u\,\mathrm{dz}\ \text{ and }\ V=\frac{1}{D}\int_{z_a}^{\eta}v\,\mathrm{dz} \tag{5.106}$$

The 2D depth averaged advection-diffusion equation for the temperature is given as:

$$\frac{\partial(\bar{T})}{\partial t}+U\frac{\partial(\bar{T})}{\partial x}+V\frac{\partial(\bar{T})}{\partial y}=\bar{D}_h\left(\frac{\partial^2\bar{T}}{\partial x^2}+\frac{\partial^2\bar{T}}{\partial y^2}\right)+A_{heat}^{rad}+(T_s-\bar{T})Q_{dis} \quad (5.107)$$

where $\bar{T}$ is the depth averaged water temperature.

The 2D depth averaged advection-diffusion equation for the salinity is:

$$\frac{\partial(\bar{S})}{\partial t}+U\frac{\partial(\bar{S})}{\partial x}+V\frac{\partial(\bar{S})}{\partial y}=\bar{D}_t\left(\frac{\partial^2\bar{S}}{\partial x^2}+\frac{\partial^2\bar{S}}{\partial y^2}\right)+(S_s-\bar{S})Q_{dis} \quad (5.108)$$

where $\bar{S}$ is the depth averaged salinity level.

The applicability of the depth averaged equations for the salinity and temperature are restricted to well-mixed conditions. The vertical gradient of these parameters and the suspended sediment concentration is usually not negligible even when the flow field can be defined by the shallow water equations. The safe approach in this regard is to use the 3D equations of these parameters in joint with the 2D depth averaged flow equations. The depth averaged form of the suspended sediment concentration has not been included in the present study in order to force the users to use the 3D version.

5.5 Summary

This chapter presented the governing equations that are to be solved in the present study. The 3D hydrodynamic equations were illustrated with descriptions of the relevant terms, the variable effect of the free surface slope on different layers of the stratified flow, and the effect of the Boussinesq approximation on the vertical momentum equation. Then the kinematic and dynamic boundary conditions were reviewed.

The Poisson equation for the hydrodynamic pressure and its boundary conditions were presented. Then the 3D conservation equations for the salinity and temperature were illustrated together with their boundary conditions. Several heat exchange relationships, necessary for the boundary conditions and the source terms of the temperature equation, were listed. Then the suspended sediment equation was presented together with relationships for describing the fall velocity, boundary condition and erosion/deposition processes in the cohesive and non-cohesive environments.

The log-law relationship and the 3D $k-\epsilon$ equations for the vertical eddy viscosity were considered with the Smagorinsky formulation for the horizontal eddy viscosity. The UNESCO equation of state (EOS–80) was illustrated for the water density involving the effect of temperature and salinity. Then the effect of the suspended sediment concentration (of sediment fractions) on the water density was considered into account.

The vertical gradient adaptive σ-coordinate transformation system (GAS, Wai and Lu 1999) and the associated transformation rules were employed in the current study in order to transform the 3D governing equations from the Cartesian coordinate system into a vertical boundary fitted coordinate system giving flexibility and control on the vertical the vertical grid spacing along the solution area. The transformed forms of the 3D flow, hydrodynamic pressure correction, temperature, salinity and suspended sediment concentration were listed. A new approach for the discretisation of the horizontal diffusion terms in the momentum equations has been proposed in order to avoid the difficulties that have been described in section (4.6). The depth averaged equations for the flow, temperature and salinity have been cited at the end of this chapter.

6 Discretisation Algorithms in the Model

6.1 Introduction

This chapter addresses the computational grid, the discretisation schemes and the interpolation elements that have been used as tools in the solution procedures of the present study. It does not involve the solution procedures of the numerical model, but rather includes the methods and schemes that have been used as discretisation tools in the solution procedures. The solution procedure itself is addressed in the next chapter (7) which uses the present chapter as a reference.

Section (6.2) demonstrates the computational grid. Section (6.3) reviews the general discretisation algorithm of the numerical model in order to show the role of each discretisation scheme in the overall picture. Section (6.4) describes the horizontal interpolation elements and shape functions. The vertical interpolation elements and shape functions are described in section (6.5). Section (6.6) addresses the complete three-dimensional interpolation procedure. This procedure is used for the 3D semi-Lagrangian advection algorithm. Section (6.7) describes the horizontal finite volume discretisation scheme. It has been used for the horizontal diffusion solver, the free surface solver and part of the hydrodynamic pressure correction step. The divergence theorem has an essential role in the finite volume algorithm. This theorem and the analytic derivation of the cell-boundary integrals through the interpolation functions have been described in section (6.8). The lateral boundary conditions are handled in a new way in order to satisfy the Neumann boundary conditions in the semi-Lagrangian advection algorithm and the FV discretisation procedure. Section (6.9) describes the handling method of the lateral boundary conditions. Section (6.10) addresses the numerical estimation of the spatial gradients at the computational nodes of the unstructured grid. Section (6.11) describes the vertical finite volume discretisation algorithm. It has been used in the vertical diffusion step of the 3D model.

6.2 Computational Grid

The present model uses the unstructured triangular grid in order to discretise the horizontal plane. The horizontal grid is projected on several layers to discretise the vertical direction. In this way, the three dimensional grid is unstructured in the horizontal plane (sigma-layers) while it is structured in the vertical direction. Figure (6.1) illustrates a sample computational grid in the three dimensional space. The discretisation and interpolation schemes that have been used for the vertical direction are different from their counterparts for the horizontal plane. The two dimensional part of the model uses the same horizontal grid.

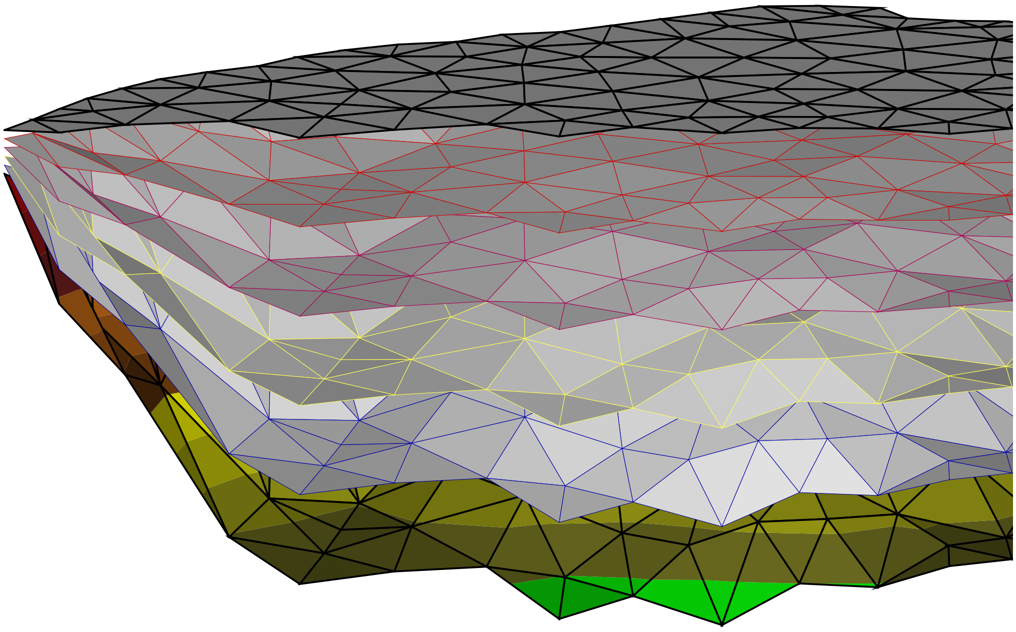


Figure 6.1. Sample computational grid

6.3 General Algorithm

The present model is based on the operator splitting approach. The 3D flow equations are solved through the following steps: the advection step, the horizontal diffusion step, the vertical diffusion step, the free surface step, the hydrodynamic pressure correction step and the velocity projection step. The advection step is solved by the semi-Lagrangian method which needs an interpolation algorithm in order to transform data from the computational nodes to other locations in the elements. There are several interpolation schemes in the literature (see section 4.5). The present model uses the two dimensional finite element shape functions in the horizontal plane in combination with the one dimensional finite element shape functions in the vertical direction in order to interpolate data in the 3D space. The FE interpolation shape functions have been used in order to ensure the consistency of the advection step with other steps that use the same shape functions for finite volume discretisation.

The horizontal diffusion step is solved layer by layer through the 2D finite volume approach. It is also the case for the free surface step. The vertical diffusion step is solved by the 1D finite volume approach in the vertical direction. The finite volume procedures in the model are based on the analytical integration of the FE-type shape functions over the FV cell areas and/or boundaries (faces). The discretisation algorithm for the 3D Poisson step has been developed through combining the 2D horizontal finite volume approach and the 1D vertical finite difference scheme. The 2D model is based on semi-Lagrangian scheme for the advection step and the horizontal FV scheme for the remaining steps.

6.4 The Horizontal Interpolation Elements

Two interpolation procedures have been used in the model: quadratic and linear. Complete interpolation functions of any order may be constructed for triangular elements using the so-called Pascal's triangle of numbers. Pascal's triangle shows both the terms needed in the interpolation polynomial and the position of the nodes on a triangular cell. Figure (6.2) illustrates the Pascal's triangle of numbers. The resulted polynomial is identical to the Taylor series of the same order.

The six-node (quadratic) and three-node (linear) triangular elements have been used with the Lagrangian shape functions in order to interpolate data horizontally between the computational nodes. In this way, all of the variables remain continuous and consistent not only at the nodes but also along the element boundary lines. Continuity along the element boundaries is not satisfied in the traditional FV cells. However it is crucial for the semi-Lagrangian advection algorithm.

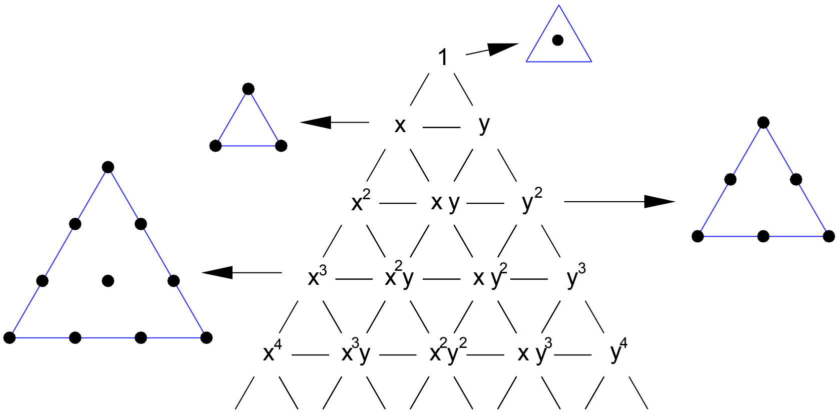


Figure 6.2. Triangular interpolation element construction by Pascal's triangle of numbers

Figure (6.3) illustrates a quadratic triangular element with the nodal points and the local numbering order. The unknown quantity may be interpolated inside the element and on the boundaries through the following relationship:

$$\phi(\tilde{x},\tilde{y}) = \sum_{j=1}^{6} \psi_j^h(\tilde{x},\tilde{y}) \cdot \phi_j \qquad (6.1)$$

where ϕ is a generic quantity at the point $(\tilde{x},\tilde{y})$ in the element, ϕ_j is the value of ϕ at the jth node of the element, and $\psi_j^h(\tilde{x},\tilde{y})$ is the shape function of the jth node.

The interpolation function (6.1) is second-order accurate because the nodal shape functions $\psi_j^h(\tilde{x},\tilde{y})$ are second-order polynomials. There are six shape functions in a quadratic interpolation element. The relationship for the shape functions is given as:

$$\psi^h(\tilde{x},\tilde{y}) = a_1 + a_2(\tilde{x}-\tilde{x}_o) + a_3(\tilde{y}-\tilde{y}_o) + a_4(\tilde{x}-\tilde{x}_o)^2 + a_5(\tilde{x}-\tilde{x}_o)(\tilde{y}-\tilde{y}_o) + a_6(\tilde{y}-\tilde{y}_o)^2 \quad (6.2)$$

where $(\tilde{x}_o,\tilde{y}_o)$ is the origin point of the local coordinate system. The polynomial coefficients, a_1 to a_6, must be defined for each node.

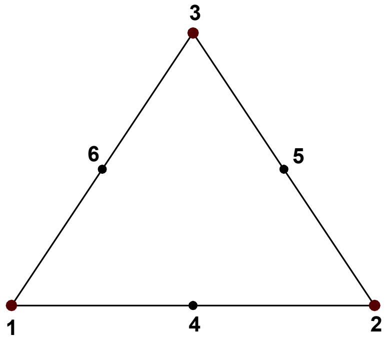


Figure 6.3. Quadratic triangular cell with six nodes

The polynomial coefficients relevant to the jth shape function ψ_j^h may be defined by solving six equations:

a) The value of ψ_j^h is equal to 1 at the jth node $(\tilde{x}_j,\tilde{y}_j)$, and

b) The value of ψ_j^h is equal to zero at all of the other nodes of the element.

The resulting shape functions are independent from the nodal values ϕ_j. They depend solely on the geometry of the triangles and the position of the nodes. The equations mentioned above have been solved analytically for the present study.

If the node number 1 is selected as the local origin point, the resulted coefficients for the shape function of the node number 1 (ψ_1^h) are:

$$\begin{array}{lll} a_1=1 & a_2=\frac{3}{2}\frac{\tilde{y}_2-\tilde{y}_3}{A_e} & a_3=\frac{3}{2}\frac{\tilde{x}_3-\tilde{x}_2}{A_e} \\ a_4=\frac{1}{2}\left(\frac{\tilde{y}_2-\tilde{y}_3}{A_e}\right)^2 & a_5=\frac{(\tilde{x}_3-\tilde{x}_2)(\tilde{y}_2-\tilde{y}_3)}{A_e^2} & a_6=\frac{1}{2}\left(\frac{\tilde{x}_3-\tilde{x}_2}{A_e}\right)^2 \end{array} \tag{6.3}$$

where A_e is the triangle's area.

The coefficients for the shape function of the node number 2 (ψ_2^h) are:

$$\begin{array}{lll} a_1=0 & a_2=\frac{\tilde{y}_1-\tilde{y}_3}{2A_e} & a_3=\frac{\tilde{x}_3-\tilde{x}_1}{2A_e} \\ a_4=\frac{1}{2}\left(\frac{\tilde{y}_1-\tilde{y}_3}{A_e}\right)^2 & a_5=\frac{(\tilde{x}_3-\tilde{x}_1)(\tilde{y}_1-\tilde{y}_3)}{A_e^2} & a_6=\frac{1}{2}\left(\frac{\tilde{x}_3-\tilde{x}_1}{A_e}\right)^2 \end{array} \tag{6.4}$$

The coefficients for the shape function of the node number 3 (ψ_3^h) are:

$$\begin{array}{lll} a_1=0 & a_2=\frac{\tilde{y}_2-\tilde{y}_1}{2A_e} & a_3=\frac{\tilde{x}_1-\tilde{x}_2}{2A_e} \\ a_4=\frac{1}{2}\left(\frac{\tilde{y}_2-\tilde{y}_1}{A_e}\right)^2 & a_5=\frac{(\tilde{x}_1-\tilde{x}_2)(\tilde{y}_2-\tilde{y}_1)}{A_e^2} & a_6=\frac{1}{2}\left(\frac{\tilde{x}_1-\tilde{x}_2}{A_e}\right)^2 \end{array} \tag{6.5}$$

The coefficients for the shape function of the node number 4 (ψ_4^h) are:

$$\begin{array}{ll} a_1=0 & a_2=2\frac{\tilde{y}_3-\tilde{y}_1}{A_e} \\ a_3=2\frac{\tilde{x}_1-\tilde{x}_3}{A_e} & a_4=\frac{(\tilde{y}_2-\tilde{y}_3)(\tilde{y}_3-\tilde{y}_1)}{A_e^2} \\ a_5=\frac{(\tilde{x}_1-\tilde{x}_3)(\tilde{y}_2-\tilde{y}_3)+(\tilde{x}_3-\tilde{x}_2)(\tilde{y}_3-\tilde{y}_1)}{A_e^2} & a_6=\frac{(\tilde{x}_3-\tilde{x}_2)(\tilde{x}_1-\tilde{x}_3)}{A_e^2} \end{array} \tag{6.6}$$

The coefficients for the shape function of the node number 5 (ψ_5^h) are:

$$\begin{array}{ll} a_1=0 & a_2=0 \\ a_3=0 & a_4=\frac{(\tilde{y}_1-\tilde{y}_2)(\tilde{y}_3-\tilde{y}_1)}{A_e^2} \\ a_5=\frac{(\tilde{x}_2-\tilde{x}_1)(\tilde{y}_3-\tilde{y}_1)+(\tilde{x}_1-\tilde{x}_3)(\tilde{y}_1-\tilde{y}_2)}{A_e^2} & a_6=\frac{(\tilde{x}_1-\tilde{x}_3)(\tilde{x}_2-\tilde{x}_1)}{A_e^2} \end{array} \tag{6.7}$$

The coefficients for the shape function of the node number 6 (ψ_6^h) are:

$$
\begin{array}{ll}
a_1=0 & a_2=2\dfrac{\tilde{y}_1-\tilde{y}_2}{A_e} \\
a_3=2\dfrac{\tilde{x}_2-\tilde{x}_1}{A_e} & a_4=\dfrac{(\tilde{y}_1-\tilde{y}_2)(\tilde{y}_2-\tilde{y}_3)}{A_e^2} \\
a_5=\dfrac{(\tilde{x}_2-\tilde{x}_1)(\tilde{y}_2-\tilde{y}_3)+(\tilde{x}_3-\tilde{x}_2)(\tilde{y}_1-\tilde{y}_2)}{A_e^2} & a_6=\dfrac{(\tilde{x}_2-\tilde{x}_1)(\tilde{x}_3-\tilde{x}_2)}{A_e^2}
\end{array}
\tag{6.8}
$$

The area of the triangular element may be defined from the following relationship:

$$
2\,A_e=\begin{vmatrix} 1 & \tilde{x}_1 & \tilde{y}_1 \\ 1 & \tilde{x}_2 & \tilde{y}_2 \\ 1 & \tilde{x}_3 & \tilde{y}_3 \end{vmatrix}=(\tilde{x}_2-\tilde{x}_3)(\tilde{y}_3-\tilde{y}_1)+(\tilde{x}_1-\tilde{x}_3)(\tilde{y}_2-\tilde{y}_3) \tag{6.9}
$$

The present model is based on the fixed computational grid. Therefore the shape functions are constant during the simulation procedure. The shape functions for a specific computational grid are calculated only once.

The finite element models usually use a master iso-parametric element with already known shape functions and surface integrals. However they compute and store the Jacobian matrices for all of the real elements in order to transform the real elements into the master element. The present model has not used the concept of master element. There is no benefit from using the master element when the shape functions and the integrals are computed analytically.

Figure (6.4) illustrates the linear elements inside each quadratic element. Linear shape functions are used in the FV algorithm for specific quantities/conditions in order to prevent numerical oscillation and instability in parts of the model (see chapter 7 for details). It must be noted that separating the advection part and handling it with the semi-Lagrangian algorithm is another arrangement to prevent numerical oscillation.

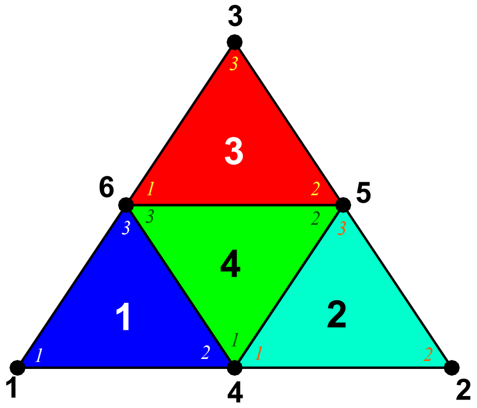


Figure 6.4. Linear interpolation triangles inside a quadratic triangle

The interpolation function for the linear element is given as:

$$\phi(\tilde{x},\tilde{y})=\sum_{j=1}^{3} {}^{l}\psi_j^h(\tilde{x},\tilde{y})\cdot\phi_j \tag{6.10}$$

in which the linear shape function ${}^{l}\psi^h$ is:

$${}^{l}\psi^h(\tilde{x},\tilde{y})=a_1+a_2(\tilde{x}-\tilde{x}_o)+a_3(\tilde{y}-\tilde{y}_o) \tag{6.11}$$

where $(\tilde{x}_o,\tilde{y}_o)$ is the local origin point for the linear interpolation sub-element.

The coefficients have been defined analytically for the present study. If the node number 1 of the linear element is selected as the local origin point, then the coefficients for the shape function of the node number 1 (${}^{l}\psi_1^h$) are:

$$a_1=1 \quad , \quad a_2=-\frac{y_3-y_2}{2A_e} \quad , \quad a_3=-\frac{x_2-x_3}{2A_e} \tag{6.12}$$

The coefficients for the shape function of the node number 2 (${}^{l}\psi_2^h$) are:

$$a_1=0 \quad , \quad a_2=\frac{y_3-y_1}{2A_e} \quad , \quad a_3=\frac{x_1-x_3}{2A_e} \tag{6.13}$$

The coefficients for the shape function of the node number 3 (${}^{l}\psi_3^h$) are:

$$a_1=0 \quad , \quad a_2=\frac{y_1-y_2}{2A_e} \quad , \quad a_3=\frac{x_2-x_1}{2A_e} \tag{6.14}$$

where A_e is the area of the linear triangular element.

The node numbers mentioned in equations (6.3) to (6.14) are different from the global numbering system of the computational grid. The numbering system in the elements is local.

6.5 The Vertical Interpolation elements

The one dimensional cubic polynomial (third degree) has been used in order to interpolate data in the vertical direction. There is an interpolation element between each two consecutive nodes in the column. The interpolation functions is used to ensure the consistency of to the solution parameters and their vertical derivatives at the nodes. In this way, both of the interpolating variables and their vertical gradients will be continuous at the element boundaries.

The interpolation scheme is used for the Cartesian and for the σ-coordinate systems. The resulting curves in the two coordinate systems are not exactly the same. The lengths of the vertical elements are not necessarily equivalent in the Cartesian coordinate system. The vertical gradient at a node is estimated by fitting a second-degree polynomial to that node and its two neighbouring nodes. This algorithm results in the following relationship for the spatial derivative of ϕ at the node number l in the Cartesian coordinate system:

$$\left(\frac{\partial\phi}{\partial z}\right)_l = A_l\phi_{l-1} + B_l\phi_l + C_l\phi_{l+1} \tag{6.15}$$

where:

$$A_l = -\frac{z_{l+1}-z_l}{(z_l-z_{l-1})(z_{l+1}-z_{l-1})} \ , \ B_l = \frac{z_{l+1}-2z_l+z_{l-1}}{(z_l-z_{l-1})(z_{l+1}-z_l)} \ , \ C_l = \frac{z_l-z_{l-1}}{(z_{l+1}-z_l)(z_{l+1}-z_{l-1})} \tag{6.16}$$

The interpolation function relevant to a vertical element in the Cartesian coordinate system is given as:

$$\phi(z) = \sum_{k=l-1}^{l+2} \psi_k^v(z)\cdot\phi_k \tag{6.17}$$

where ϕ is the interpolating quantity at the arbitrary position z in the element, ϕ_k is the nodal value of ϕ at the node number k, $\psi_k^v(z)$ is the shape function of the kth node. The global numbering system has been used in the vertical direction.

The vertical shape functions are cubic polynomials of the form:

$$\psi^v(z) = b_1 + b_2(z-z_l) + b_3(z-z_l)^2 + b_4(z-z_l)^3 \tag{6.18}$$

where z_l is the z value of the computational node below the point and b_1 to b_4 are polynomial coefficients.

The polynomial coefficients relevant to the l^{th} shape function ψ_k^v may be defined by solving four equations:

a) The value of $\phi(z)$ is equal to ϕ_k at the nodes l and $l+1$, and

b) the value of $\partial\phi(z)/\partial z$ is equal to equation (6.15) at the nodes l and $l+1$.

The resulting shape functions are independent from the nodal values ϕ_k. They depend solely on the vertical positions of the nodes $l-1$, l, $l+1$ and $l+2$.

The equations mentioned above have been solved analytically for the present study. The resulting coefficients for the shape function of node number $l-1$, ψ_{l-1}^{v}, are:

$$b_1=0 \ , \ b_2=A_l \ , \ b_3=-\frac{2A_l}{(\Delta z)} \ , \ b_4=\frac{A_l}{(\Delta z)^2} \tag{6.19}$$

The coefficients for the shape function of node number l, ψ_l^{v}, are:

$$b_1=1 \ , \ b_2=B_l \ , \ b_3=-\frac{3}{(\Delta z)^2}-\frac{2B_l}{(\Delta z)}-\frac{A_{l+1}}{(\Delta z)} \ , \ b_4=\frac{A_{l+1}}{(\Delta z)^2}+\frac{B_l}{(\Delta z)^2}+\frac{2}{(\Delta z)^3} \tag{6.20}$$

The coefficients for the shape function of node number $l+1$, ψ_{l+1}^{v}, are:

$$b_1=0 \ , \ b_2=C_l \ , \ b_3=\frac{3}{(\Delta z)^2}-\frac{2C_l}{(\Delta z)}-\frac{B_{l+1}}{(\Delta z)} \ , \ b_4=\frac{B_{l+1}}{(\Delta z)^2}+\frac{C_l}{(\Delta z)^2}-\frac{2}{(\Delta z)^3} \tag{6.21}$$

The coefficients for the shape function of the node number $l+2$, ψ_{l+2}^{v}, are:

$$b_1=0 \ , \ b_2=0 \ , \ b_3=-\frac{c_{l+1}}{(\Delta z)} \ , \ b_4=\frac{c_{l+1}}{(\Delta z)^2} \tag{6.22}$$

The quantity Δz in equations (6.19) to (6.22) is $\Delta z = z_{l+1} - z_l$.

The node number $l-1$ does not exist for the area between the bed layer and the next one (the node numbers 1 and 2). Thus, another condition is needed for calculating the coefficients in equation (2.18). For the Neumann boundary conditions, the near-bed vertical gradient is known and may be used as the fourth condition. A virtual layer is used under the bed layer in order to handle the boundary condition without changing the discretisation scheme. This approach lets the program to use a unique interpolation procedure for all of the vertical elements. However an extra relationship is needed for the virtual node. This relationship is defined in such a way that the vertical gradient at the bed layer becomes equivalent to the Neumann boundary condition:

$$z_0=z_1-(z_2-z_1) \ , \ \phi_2-\phi_0=2(z_2-z_1)\left(\frac{\partial\phi}{\partial z}\right)_1 \tag{6.23}$$

where the subscripts 0 and 1 stand for the virtual layer and the bed layer respectively.

For the explicit interpolation procedures equation (6.23) is solved directly in order to define the nodal value at the under-bed virtual layer. For the implicit numerical schemes equation (6.23) is added to the linear set of equations that must be solved numerically.

Sometimes the bed boundary condition is a linear relationship between the bed value and the vertical gradient. In this case the corresponding equation for the virtual node may be derived from equation (6.23) as follows:

$$\phi_2 - 2\alpha(z_2 - z_1)\phi_1 - \phi_0 = 2\beta(z_2 - z_1) \quad \text{where} \quad \left(\frac{\partial \phi}{\partial z}\right)_b = \alpha\phi_1 + \beta \tag{6.24}$$

A similar approach may be used for the water surface by adding a virtual node above the water level with the following characteristics:

$$z_{k+1} = z_k + (z_k - z_{k-1}) \ , \ \phi_{k+1} - \phi_{k-1} = 2(z_k - z_{k-1})\left(\frac{\partial \phi}{\partial z}\right)_k \tag{6.25}$$

where the subscripts k and $k+1$ stand for the free surface layer and the virtual layer respectively.

If the vertical gradient at the boundary is linked to the boundary value itself, the following equation may be used:

$$\phi_{k+1} - 2\alpha(z_k - z_{k-1})\phi_k - \phi_{k-1} = 2\beta(z_k - z_{k-1}) \quad \text{where} \quad \left(\frac{\partial \phi}{\partial z}\right)_b = \alpha\phi_1 + \beta \tag{6.26}$$

The vertical interpolation shape functions may be simplified further when the procedure is needed directly in the σ-coordinate system instead of the Cartesian one. Considering that $\Delta\tilde{z}$ is fixed and unique for all of the vertical elements, equations (6.17) to (6.22) may be simplified as follows:

$$\phi(\tilde{z}) = \sum_{k=l-1}^{l+2} \psi_k^v(\tilde{z}) \cdot \phi_k \tag{6.27}$$

$$\psi^v(\tilde{z}) = b_1 + b_2(\tilde{z} - \tilde{z}_l) + b_3(\tilde{z} - \tilde{z}_l)^2 + b_4(\tilde{z} - \tilde{z}_l)^3 \tag{6.28}$$

For ψ_{l-1}^v:

$$b_1 = 0,\ b_2 = -\frac{1}{2(\Delta\tilde{z})},\ b_3 = \frac{1}{(\Delta\tilde{z})^2},\ b_4 = -\frac{1}{2(\Delta\tilde{z})^3} \tag{6.29}$$

For ψ_l^v:

$$b_1 = 1,\ b_2 = 0,\ b_3 = -\frac{5}{2(\Delta\tilde{z})^2},\ b_4 = \frac{3}{2(\Delta\tilde{z})^3} \tag{6.30}$$

For ψ_{l+1}^v:

$$b_1 = 0,\ b_2 = \frac{1}{2(\Delta\tilde{z})},\ b_3 = \frac{2}{(\Delta\tilde{z})^2},\ b_4 = -\frac{3}{2(\Delta\tilde{z})^3} \tag{6.31}$$

For ψ_{l+2}^v:

$$b_1 = 0,\ b_2 = 0,\ b_3 = -\frac{1}{2(\Delta\tilde{z})^2},\ b_4 = \frac{1}{2(\Delta\tilde{z})^3} \tag{6.32}$$

The integral of the interpolation function (6.17) between the layers l and $l+1$ is written as:

$$\int_{l}^{l+1} \phi(z)\,dz = \int_{l}^{l+1} \sum_{k=l-1}^{l+2} \psi_k^v(z)\cdot\phi_k\,dz = \sum_{k=l-1}^{l+2} \phi_k \int_{l}^{l+1} \psi_k^v(z)\,dz \tag{6.33}$$

in which the integral for each shape function may be defined analytically as follows:

$$\begin{aligned}\int_{l}^{l+1} \psi_k^v(z)\,dz &= \int_{l}^{l+1} \left[b_1 + b_2(z-z_l) + b_3(z-z_l)^2 + b_4(z-z_l)^3\right] d\tilde{z}\\ &= b_1(\Delta z) + b_2(\Delta z)^2 + b_3(\Delta z)^3 + b_4(\Delta z)^4\end{aligned} \tag{6.34}$$

where $\Delta z = (z_{l+1} - z_l)$.

Equation (6.33) may be used for numerical depth integration.

6.6 The 3D Interpolation procedure

The horizontal and vertical interpolation equations (6.1 and 6.27) may be mixed together in order to link the unknown quantity at an arbitrary point $(\tilde{x}, \tilde{y}, \tilde{z})$ to the surrounding nodes. The relationship for the 3D interpolation is given as:

$$\phi(\tilde{x},\tilde{y},\tilde{z}) = \sum_{j=1}^{6}\left(\psi_j^h(\tilde{x},\tilde{y})\cdot \sum_{k=l-1}^{l+2}\left(\psi_k^v(\tilde{z})\cdot\phi_{j,k}\right)\right) \tag{6.35}$$

Figure (6.5) illustrates the affecting nodes for data interpolation at an arbitrary point in the 3D space. Equation (6.35) determines the weight of each node in the interpolation procedure.

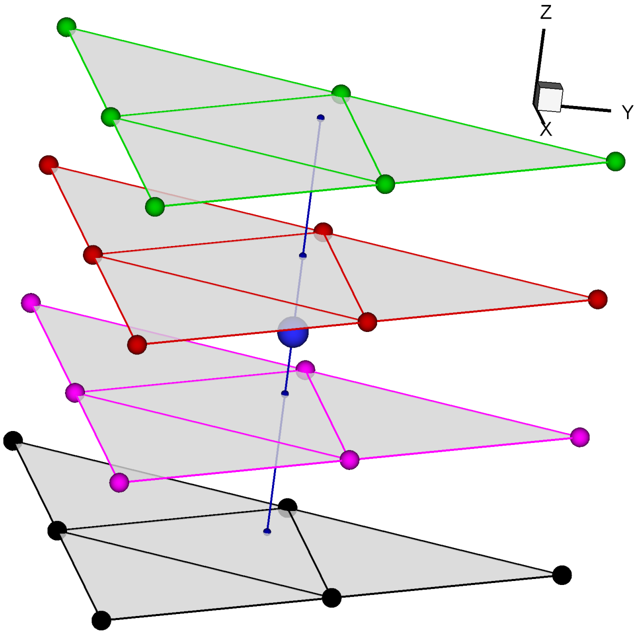


Figure 6.5. Affecting nodes for data interpolation at an arbitrary point in 3D space

6.7 The Horizontal Discretisation Algorithm

The finite volume approach has been used as the horizontal discretisation scheme in the sigma coordinate planes except for the semi-Lagrangian advection step. The finite volume schemes are usually based on the cell-averaged values or numerical integration. The present scheme makes use of the interpolating shape functions described in section (6.4) in order to integrate the equations over the cell area and the cell boundaries. This type of finite volume scheme may be considered as a hybrid finite volume finite element scheme in the literature.

Figure (6.6) illustrates the finite volume cells. Each cell belongs to a computational node (vertex-cell). The computational nodes are the vertices of the interpolation triangles and the mid-points of their edges. There is an independent cell for each node. Therefore each unknown variable has an independent equation which means that the number of unknowns is equal to the number of equations.

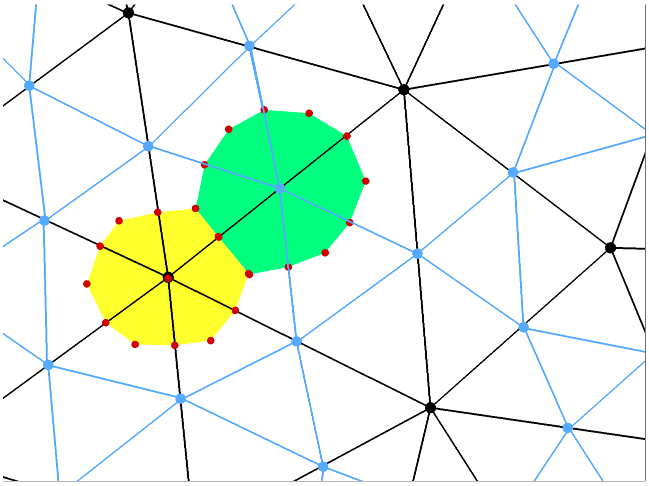


Figure 6.6. Two sample finite volume cells over the interpolating triangular elements

The finite volume discretisation scheme is based on the integration over the cell area. The Gauss' divergence theorem is commonly used in order to convert the cell area integration of the divergence terms into the cell boundary integration of the quantities. The area and boundary integrals can be determined via analytical integration of the interpolating shape functions. It can be shown that when the analytical shape function and the analytical integration is used, there is no difference between the result of the boundary integral and the cell-area integration of the divergence terms. The proof for this claim is exactly the proof of the Gauss' divergence theorem. The integration procedure has been described below.

Figure (6.6) shows that each FV cell-area contains parts from two or more interpolation elements. Each interpolation element has its specific shape functions. Therefore the cell-area integral is the sum of two or more sub-cell integrals. On the other hand the analytical integration is easier on triangle than other shapes. Therefore each FV cell is partitioned into several sub-cell triangular parts as illustrated in figure (6.7). Considering the quadratic interpolation elements, equation (6.1) may be integrated analytically over each part in order to derive the integral of a generic quantity ϕ over a finite volume cell. The result is:

$$\sum_{1}^{m} \iint_{sp} \phi(\tilde{x},\tilde{y})\,dx\,dy=\sum_{1}^{m} \iint_{sp} (\sum_{j=1}^{6} \psi_j^h(\tilde{x},\tilde{y})\cdot\phi_j)\,dx\,dy=\sum_{1}^{m} \sum_{j=1}^{6} \left(\phi_j (\iint_{sp} \psi_j^h(\tilde{x},\tilde{y})\,dx\,dy)\right) \quad (6.36)$$

where the subscript sp stands for the sub-cell triangle parts in a FV cell, m is the number of the sub-cell triangle parts, $\phi(\tilde{x},\tilde{y})$ is the interpolating function to be integrated, ϕ_j is the value of $\phi(\tilde{x},\tilde{y})$ at the nodes of the interpolation element which contains sp, and $\psi_j^h(\tilde{x},\tilde{y})$ is the shape function of the interpolation element nodes.

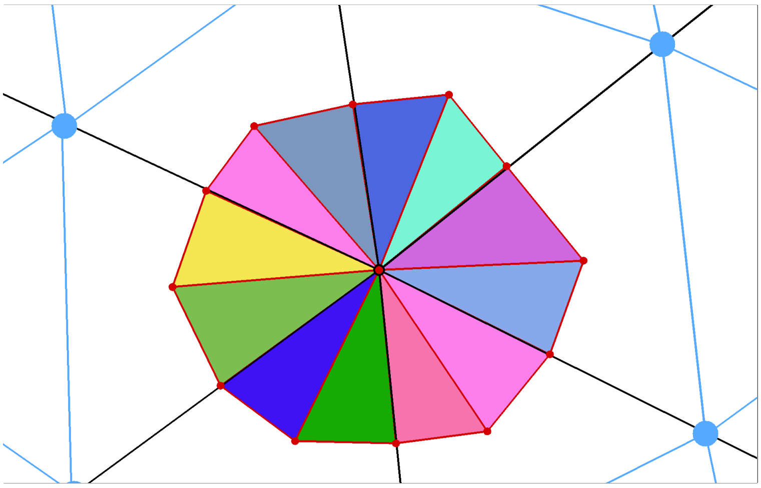


Figure 6.7. Integration sub-cell triangles in a finite volume cell

Equation (6.36) shows that the integration of the variable ϕ on a finite volume cell depends on the surrounding nodal values and the integration of the nodal shape functions. All of the shape functions are polynomials of the same form (see equation 6.2). The analytical interpolation of a polynomial of the form (6.2) on a triangular area sp is written as:

$$\begin{aligned}
&\iint_{sp} \psi^h(\tilde{x},\tilde{y})\,ds = \\
&\quad \iint_{sp} a_1\,ds + \iint_{sp} a_2(\tilde{x}-\tilde{x}_o)\,ds + \iint_{sp} a_3(\tilde{y}-\tilde{y}_o)\,ds \\
&\quad + \iint_{sp} a_4(\tilde{x}-\tilde{x}_o)^2\,ds + \iint_{sp} a_5(\tilde{x}-\tilde{x}_o)(\tilde{y}-\tilde{y}_o)\,ds + \iint_{sp} a_6(\tilde{y}-\tilde{y}_o)^2\,ds = \\
&\quad a_1\iint_{sp} ds + a_2\iint_{sp}(\tilde{x}-\tilde{x}_o)\,ds + a_3\iint_{sp}(\tilde{y}-\tilde{y}_o)\,ds \\
&\quad + a_4\iint_{sp}(\tilde{x}-\tilde{x}_o)^2\,ds + a_5\iint_{sp}(\tilde{x}-\tilde{x}_o)(\tilde{y}-\tilde{y}_o)\,ds + a_6\iint_{sp}(\tilde{y}-\tilde{y}_o)^2\,ds
\end{aligned} \tag{6.37}$$

where the subscript o stands for the node number 1 of the quadratic interpolation triangle which involves sp.

Analytic integration of the terms in equation (6.37) results in the following relationships:

$$\iint_{sp} ds = \frac{1}{2}\left[x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2)\right] \tag{6.38}$$

$$\iint_{sp} x\,ds = \frac{1}{6}\left[(y_1-y_2)(x_3^2-x_1x_2)+(y_2-y_3)(x_1^2-x_2x_3)+(y_3-y_1)(x_2^2-x_1x_3)\right] \tag{6.39}$$

$$\iint_{sp} y\,ds = \frac{1}{6}\left[(x_2-x_1)(y_3^2-y_1y_2)+(x_3-x_2)(y_1^2-y_2y_3)+(x_1-x_3)(y_2^2-y_1y_3)\right] \tag{6.40}$$

$$\begin{aligned}
&\iint_{sp} x^2\,ds = \\
&\quad \frac{1}{12}\left[(x_3^2+x_1^2)(x_3y_1-x_1y_3)+(x_2^2+x_3^2)(x_2y_3-x_3y_2)+(x_1^2+x_2^2)(x_1y_2-x_2y_1)\right] \\
&\quad +\frac{1}{12}\left[(x_3^2-x_2^2)x_1y_1+(x_2^2-x_1^2)x_3y_3+(x_1^2-x_3^2)x_2y_2\right]
\end{aligned} \tag{6.41}$$

$$\begin{aligned}
&\iint_{sp} y^2\,ds = \\
&\quad \frac{1}{12}\left[(y_1^2+y_3^2)(x_3y_1-x_1y_3)+(y_2^2+y_1^2)(x_1y_2-x_2y_1)+(y_3^2+y_2^2)(x_2y_3-x_3y_2)\right] \\
&\quad +\frac{1}{12}\left[(y_1^2-y_2^2)x_3y_3+(y_2^2-y_3^2)x_1y_1+(y_3^2-y_1^2)x_2y_2\right]
\end{aligned} \tag{6.42}$$

$$\begin{aligned}
&\iint_{sp} x\,y\,ds = \\
&\quad \frac{x_3-x_1}{24}\left[(x_3+x_1)(y_3^2+y_1^2)+2(y_3+y_1)(x_3y_3+x_1y_1)\right] \\
&\quad +\frac{x_2-x_3}{24}\left[(x_2+x_3)(y_2^2+y_3^2)+2(y_2+y_3)(x_2y_2+x_3y_3)\right] \\
&\quad +\frac{x_1-x_2}{24}\left[(x_1+x_2)(y_1^2+y_2^2)+2(y_1+y_2)(x_1y_1+x_2y_2)\right]
\end{aligned} \tag{6.43}$$

where the subscript sp stands for a triangle with the vertices $(x_{1,}y_1)$, $(x_{2,}y_2)$ and $(x_{3,}y_3)$ in the local coordinate system ($x\equiv\tilde{x}-\tilde{x}_o$, $y\equiv\tilde{y}-\tilde{y}_o$).

Equations (6.37) to (6.43) show that the integrals in equation (6.36) are only depends on the horizontal mesh-geometry. They are fixed in time and may be computed just once. In this way, the integral of the interpolation function (6.36) over the finite volume cell is linked to the surrounding nodal values and the cell centred node itself.

The integral of the spatial gradient of the interpolation function on the cell area may be determined analytically as follows:

$$\begin{aligned}\sum_1^m \iint_{sp} \frac{\partial \phi(\tilde{x},\tilde{y})}{\partial \tilde{x}} dx\,dy &= \sum_1^m \iint_{sp} \frac{\partial(\sum_{j=1}^{6} \psi_j^h(\tilde{x},\tilde{y})\cdot\phi_j)}{\partial \tilde{x}} dx\,dy \\ &= \sum_1^m \sum_{j=1}^{6} \left(\phi_j \left(\iint_{sp} \frac{\partial \psi_j^h(\tilde{x},\tilde{y})}{\partial \tilde{x}} dx\,dy \right) \right)\end{aligned} \tag{6.44}$$

A similar relationship can be defined for the $\tilde{y}$- gradient.

Equation (6.44) shows that the gradient operator acts on the shape functions. The spatial gradient of a polynomial shape function of the form (6.2) may be determined analytically as follows:

$$\frac{\partial}{\partial \tilde{x}} \psi^h(\tilde{x},\tilde{y}) = a_2 + 2a_4(\tilde{x}-\tilde{x}_o) + a_5(\tilde{y}-\tilde{y}_o) \tag{6.45}$$

$$\frac{\partial}{\partial \tilde{y}} \psi^h(\tilde{x},\tilde{y}) = a_3 + a_5(\tilde{x}-\tilde{x}_o) + 2a_6(\tilde{y}-\tilde{y}_o) \tag{6.46}$$

The integrals of equations (6.45) and (6.46) over sp are derived as follows:

$$\iint_{sp} \frac{\partial \psi^h}{\partial \tilde{x}} dx\,dy = a_2 \iint_{sp} dx\,dy + 2a_4 \iint_{sp} (\tilde{x}-\tilde{x}_o)\,dx\,dy + a_5 \iint_{sp} (\tilde{y}-\tilde{y}_o)\,dx\,dy \tag{6.47}$$

$$\iint_{sp} \frac{\partial \psi^h}{\partial \tilde{y}} dx\,dy = a_3 \iint_{sp} dx\,dy + a_5 \iint_{sp} (\tilde{x}-\tilde{x}_o)\,dx\,dy + 2a_6 \iint_{sp} (\tilde{y}-\tilde{y}_o)\,dx\,dy \tag{6.48}$$

Equations (6.38) to (6.40) can be used in order to define the right-hand side of equations (6.47) and (6.48) analytically. In this way, the integrals of the gradients over a FV cell are linked to its neighbouring nodal values and its own, through constant coefficients.

The analytic derivations of the cell surface integrals are also possible for the linear interpolation sub-elements described in section (6.4). Integration on the linear sub-elements is needed locally in situations where the second-order estimation results in numerical oscillation. The integrals of a generic parameter ϕ and its $\tilde{x}$- and $\tilde{y}$- gradients on a FV cell with the linear interpolation elements are defined as follows:

$$\sum_{1}^{m}\iint_{sp}\phi(\tilde{x},\tilde{y})\,dx\,dy=\sum_{1}^{m}\iint_{sp}\sum_{j=1}^{3}\psi_j^h(\tilde{x},\tilde{y})\cdot\phi_j\,dx\,dy$$
$$=\sum_{1}^{m}\sum_{j=1}^{3}\left(\phi_j(\iint_{sp}\psi_j^h(\tilde{x},\tilde{y})\,dx\,dy)\right) \tag{6.49}$$

$$\sum_{1}^{m}\iint_{sp}\frac{\partial\phi(\tilde{x},\tilde{y})}{\partial\tilde{x}}\,dx\,dy=\sum_{1}^{m}\iint_{sp}\frac{\partial(\sum_{j=1}^{3}\psi_j^h(\tilde{x},\tilde{y})\cdot\phi_j)}{\partial\tilde{x}}\,dx\,dy$$
$$=\sum_{1}^{m}\sum_{j=1}^{3}\left(\phi_j(\iint_{sp}\frac{\partial\psi_j^h(\tilde{x},\tilde{y})}{\partial\tilde{x}}\,dx\,dy)\right) \tag{6.50}$$

$$\sum_{1}^{m}\iint_{sp}\frac{\partial\phi(\tilde{x},\tilde{y})}{\partial\tilde{y}}\,dx\,dy=\sum_{1}^{m}\iint_{sp}\frac{\partial(\sum_{j=1}^{3}\psi_j^h(\tilde{x},\tilde{y})\cdot\phi_j)}{\partial\tilde{y}}\,dx\,dy$$
$$=\sum_{1}^{m}\sum_{j=1}^{3}\left(\phi_j(\iint_{sp}\frac{\partial\psi_j^h(\tilde{x},\tilde{y})}{\partial\tilde{y}}\,dx\,dy)\right) \tag{6.51}$$

where the linear shape function ψ is defined by equations (6.11) to (6.14).

The derivatives of the linear polynomial shape function are:

$$\frac{\partial}{\partial\tilde{x}}\psi^h(\tilde{x},\tilde{y})=a_2 \tag{6.52}$$

$$\frac{\partial}{\partial\tilde{y}}\psi^h(\tilde{x},\tilde{y})=a_3 \tag{6.53}$$

The integrals of equations (6.52) and (6.53) on sp are derived as follows:

$$\iint_{sp}\frac{\partial\psi^h}{\partial\tilde{x}}\,dx\,dy=a_2\iint_{sp}dx\,dy \tag{6.54}$$

$$\iint_{sp}\frac{\partial\psi^h}{\partial\tilde{y}}\,dx\,dy=a_3\iint_{sp}dx\,dy \tag{6.55}$$

6.8 Divergence Theorem

The divergence theorem is an essential part of most finite volume schemes. The plane version of the theorem states that the 2D integral of the divergence of $\vec{\phi}$ ($\nabla\cdot\vec{\phi}$) over the cell area and the 1D integral of $\vec{\phi}$ over the cell boundary are related by:

$$\iint_{S}\nabla\cdot\vec{\phi}\;dA=\int_{\partial S}\vec{\phi}\cdot\vec{n}\;ds \tag{6.56}$$

where ∂S is the boundary of the finite volume cell.

Equation (6.56) is used in the FV algorithm in order to replace the 2D divergence integrals into some boundary integrals. The hybrid finite element finite volume algorithm of the present study can determine the boundary integrals analytically. The 1D integral along the cell-boundary may be partitioned into several boundary segments as illustrated in figure (6.7). The unit vector $\vec{n}$ is constant along each boundary segment line and

$$\vec{\phi}\cdot\vec{n}=\phi_x n_x+\phi_y n_y \tag{6.57}$$

where $\vec{\phi}=(\phi_x,\phi_y)$ and $\vec{n}=(n_x,n_y)$. Therefore:

$$\iint_S \nabla\cdot\vec{\phi}\ dA=\int_{\partial S}\vec{\phi}\cdot\vec{n}\ ds=n_x\int_{\partial S}\phi_x\ ds+n_y\int_{\partial S}\phi_y\ ds \tag{6.58}$$

The line-integral of a generic scalar function ϕ along a cell boundary segment can be derived analytically. Each boundary line segment is inside an interpolation element in which the generic function ϕ may be written as follows:

$$\phi(x,y)=\sum_{j=1}^{6\,\mathrm{or}\,3}\psi_j(x,y)\cdot\phi_j \tag{6.59}$$

where the nodes j correspond to the interpolation element that includes the boundary segments.

Thus we have:

$$\int_{\partial S}\phi\ ds=\sum_{i=1}^{m}\left(\int_{\partial s_i}\sum_{j=1}^{6\,\mathrm{or}\,3}\psi_j(x,y)\cdot\phi_j\ ds\right)=\sum_{i=1}^{m}\sum_{j=1}^{6\,\mathrm{or}\,3}\phi_j\int_{\partial s_i}\psi_j(x,y)\ ds \tag{6.60}$$

where ∂s_i is the cell boundary segment and m is the number of boundary segments. Each boundary segment has its own set of ϕ_j and $\psi_j(x,y)$ in equation (6.60). The cell boundary ∂S is composed from several line-segments ∂s_i each segment in a boundary segment analytically.

Equation (6.58) may be written as follows:

$$\iint_S \nabla\cdot\vec{\phi}\ dA=\sum_{i=1}^{m}\sum_{j=1}^{6\,\mathrm{or}\,3}\left(n_x(\phi_x)_j+n_y(\phi_y)_j\right)\int_{\partial s_i}\psi_j\ ds \tag{6.61}$$

where each boundary segment has its own set of $(\phi_x)_j$, $(\phi_y)_j$ and $\psi_j(x,y)$ relevant to its element-node.

The coefficients $\int_{\partial s_i}\psi_j(x,y)\ ds$ are not dependent on the nodal values of the solution quantity. They are solely functions of the computational grid geometry. These coefficients may be calculated analytically, because the functions $\psi_j(x,y)$ have analytical representations along the boundary segments. The integration procedure is described:

The boundary line segment from the point (x_s, y_s) to the point (x_e, y_e) may be determined by the following parametric relationship:

$$\begin{aligned} x(t) &= x_s + a_l t \\ y(t) &= y_s + b_l t \end{aligned} \qquad (6.62)$$

where:

$$t \in [0, L], \quad L = \sqrt{(x_e - x_s)^2 + (x_e - y_s)^2}, \quad a_l = \frac{x_e - x_s}{L}, \quad b_l = \frac{y_e - y_s}{L} \qquad (6.63)$$

and the subscripts s and e are the start point and the end point of the boundary line segment respectively.

When the divergence theorem is used with the quadratic interpolation element, the shape function $\psi(x, y)$ can be written as follows:

$$\psi(x, y) = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 \qquad (6.64)$$

Equation (6.62) can be inserted in equation (6.64) which results in the following relationship:

$$\psi(x) = A_1 + A_2 t + A_3 t^2 \qquad (6.65)$$

where:

$$\begin{aligned} A_1 &= a_1 + a_2 x_s + a_3 y_s + a_4 x_s^2 + a_5 x_s y_s + a_6 y_s^2 \\ A_2 &= a_2 a_l + a_3 b_l + 2 x_s a_4 a_l + a_5 x_s b_l + a_5 y_s a_l + 2 a_6 y_s b_l \\ A_3 &= a_4 a_l^2 + a_5 a_l b_l + a_6 b_l^2 \end{aligned} \qquad (6.66)$$

Therefore:

$$\int_{\partial s_j} \psi \; ds = \int_0^L \psi \; dt = A_1 L + \frac{1}{2} A_2 L^2 + \frac{1}{3} A_3 L^3 \qquad (6.67)$$

When the divergence theorem is used with the linear interpolation element, the shape function $\psi(x, y)$ in equation (6.61) can be written as follows:

$${}^{l}\psi^{h}(x, y) = a_1 + a_2 x + a_3 y \qquad (6.68)$$

Then equation (6.62) can be inserted into equation (6.63) to give the following relationship:

$$\psi(x) = A_1 + A_2 x \qquad (6.69)$$

where:

$$\begin{aligned} A_1 &= a_1 + a_2 x_s + a_3 y_s \\ A_2 &= a_2 a_l + a_3 b_l \end{aligned} \qquad (6.70)$$

Therefore:

$$\int_{\partial s_j} \psi \; ds = \int_0^L \psi \; dx = A_1 L + \frac{1}{2} A_2 L^2 \qquad (6.71)$$

The coefficients $\int_{\partial s_i} \Psi_j \, ds$ are computed for all of the finite volume cells just once at the beginning of the simulation.

Applying the divergence theorem results in a similar coefficient matrix as the traditional surface integration over the FV cells, when the hybrid finite element finite volume method is used together with the analytic interpolation functions. However, it is easier to handle some of the boundary conditions such as the closed boundary, when the divergence theorem and the resulting cell-boundary flux determination are used. Many of the existing TVD schemes are also applied on the cell boundary fluxes.

6.9 Handling the Lateral Boundary Conditions

The interpolation elements need modification at the boundaries. There is no problem with the normal interpolation elements when the boundary condition is of the Dirichlet type. However, the interpolation elements along the Neumann boundaries need some adjustments, in order to satisfy the Neumann condition without the need to condense the computational grid. Many existing discretisation schemes ignore this requirement. The present study introduces the Neumann interpolation elements with virtual nodes in order to avoid dense grid near the boundary. This procedure is used for interpolation in the advection step and for the finite volume procedure in some of the model components.

Figure (6.8) illustrates a normal computational grid with normal interpolation elements near the boundary. These boundary elements are satisfactory for the Dirichlet boundary condition. Figure (6.9) illustrates the modified computational grid. Each interpolation element with a boundary side has been superseded with two interpolation elements containing virtual node in such a way that the boundary line be their symmetric axis. The Neumann interpolation elements are used exactly as the normal interpolation elements.

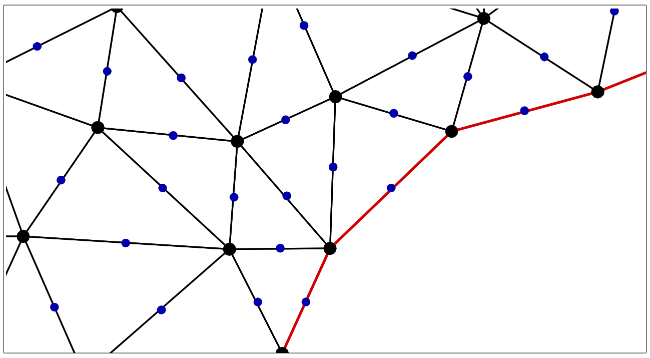


Figure 6.8. Computational grid with normal interpolation elements at the boundary

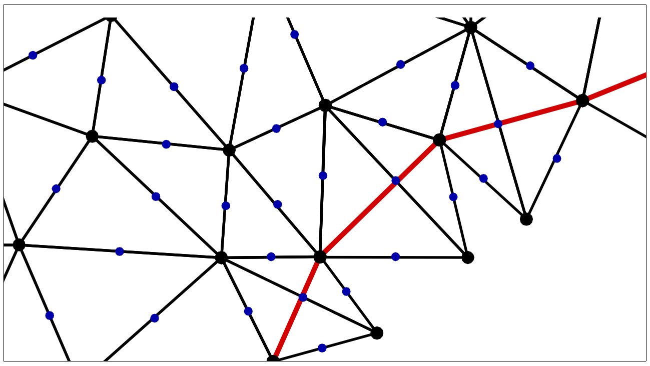


Figure 6.9. Neumann boundary interpolation elements

The virtual nodes in figure (6.9) are proportional and symmetric to real nodes. In this way, the positions of the virtual nodes are determined. The normal gradient of the interpolation function at the boundary may be controlled by linking the nodal values of the virtual nodes to their proportional real nodes. If the nodal values of the virtual nodes are set equal to their real counterparts, then the normal gradient at the boundary will be zero. The following relationship may be used in order to produce any arbitrary normal gradient at the boundary:

$$\begin{aligned} \phi_{n1}^{virtual} - \phi_{n1}^{real} &= l \cdot \phi_{,n} \quad , \quad \textit{for the vertex nodes} \\ \phi_{n2}^{virtual} - \phi_{n2}^{real} &= \frac{l}{2} \cdot \phi_{,n} \quad , \quad \textit{for the mid} - \textit{side nodes} \end{aligned} \qquad (6.72)$$

where l is the distance between the real and the virtual vertex nodes (see figure 6.9), and $\phi_{,n}$ is the normal gradient which is known as the boundary condition. Equation (6.72) can be used for the extra relationships needed because of adding the virtual nodes.

When the second gradient of the solution quantity in the direction normal to the boundary is zero, the following relationship may be used:

$$\begin{aligned} \phi_{n1}^{virtual} - 2\phi_{b1} + \phi_{n1}^{real} &= 0 \quad , \quad \textit{for the vertex nodes} \\ \phi_{n2}^{virtual} - 2\sum_{j=1}^{6} \phi_j \psi_j\left(\frac{b}{2}, 0\right) + \phi_{n2}^{real} &= 0 \quad , \quad \textit{for the mid} - \textit{side nodes} \end{aligned} \qquad (6.73)$$

The coefficient matrix for the global system of equations is fixed in time even when the Neumann boundary values are not constant in time. The Neumann interpolation element may also be used for situations where the boundary condition is given as a linear relationship between the normal gradient and the boundary value itself.

It should be noted that the above procedure does not addresses the Neumann condition completely at the vertex boundary nodes if the boundary line direction changes considerably at these boundary points. This weakness is not due to the procedure itself, but due to the natural behaviour of the discretisation procedure which converts the continuous boundary curve into broken segments. Therefore the technique of condensing the computational grid is nevertheless necessary near the boundaries with severe curvature, except for the corners.

The nodal values at the boundary nodes with the Dirichlet condition are known. There is no need for extra equations to be included in the linear system of equations. However, the boundary nodes with the Neumann boundary condition are unknown. Each Neumann boundary node needs a specific equation per unknown variable. This equation can be based on the finite volume scheme as at any other nodes. The finite volume cells for the boundary nodes do not cover part of the area which is outside the solution domain.

There are practical situations in which some of the variables have Dirichlet condition on a specific boundary segment while some others have Neumann condition at the same location. Therefore the Neumann and Dirichlet interpolation elements will be both used at the same time. However, the mid-side boundary nodes are not necessarily located in the same position. It is a good idea to use semi-length triangular interpolation elements at the boundaries in order to prevent interpolation of data along the boundary line. The mid-side boundary nodes are identical for the two element types when semi-length triangles are used as the Dirichlet boundary elements.

It is possible to use the unstructured finite difference method instead of the finite volume approach for determining the governing equations at the Neumann boundary nodes. Such a scheme estimates the x- and y-gradients of the unknown quantity at the boundary nodes using the position and the nodal values of the surrounding nodes. This method is especially useful at the vertex boundary nodes when the boundary line direction changes rapidly. The unstructured finite difference scheme is described in detail in section (6.7). However, the result of that procedure is:

$$\left(\frac{\partial \phi}{\partial x}\right)_j = \sum_{i=1}^{m} c_i \cdot \phi_i - \phi_j \sum_{i=1}^{m} c_i \quad , \quad \left(\frac{\partial \phi}{\partial y}\right)_j = \sum_{i=1}^{m} d_i \cdot \phi_i - \phi_j \sum_{i=1}^{m} d_i \tag{6.74}$$

where c_i and d_i are constant coefficients based on the mesh geometry, the subscript j stands for the boundary node, and ϕ_i is the nodal value at the i^{th} neighbouring node. Figure (6.10) illustrates the selected neighbouring nodes for two vertex boundary nodes in a sample grid.

The following relationship links the Neumann boundary condition to the nodal values of the boundary node and its surrounding nodes:

$$\phi_{,n}=n_x\phi_{,x}+n_y\phi_{,y}=-\left(n_x\cdot\sum_{i=1}^{m}c_i+n_y\cdot\sum_{i=1}^{m}d_i\right)\cdot\phi_j+\sum_{i=1}^{m}(n_x\cdot c_i+n_y\cdot d_i)\cdot\phi_i \tag{6.75}$$

where $\phi_{,x}$, $\phi_{,y}$ and $\phi_{,n}$ are the spatial gradients of ϕ in the x, y, and the normal directions respectively. The subscript n stands for the direction normal to the boundary. The coefficients n_x and n_y are the constituents of the outward unit vector normal to the boundary (n_x, n_y). The normal direction at a boundary node is determined using the side between its neighbouring boundary nodes.

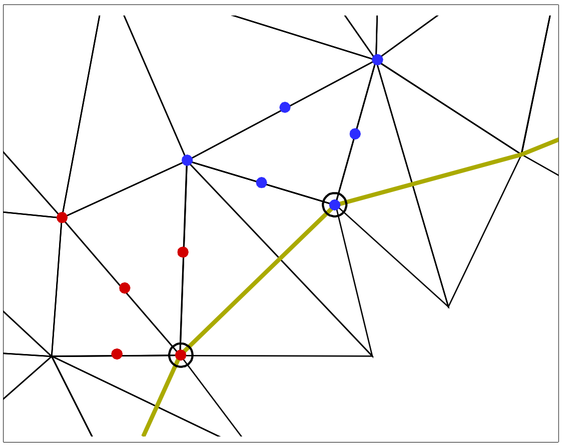


Figure 6.10. Selected surrounding nodes for two vertex boundary nodes

When the second gradient of the solution quantity in the direction normal to the boundary is zero, the following relationship is used:

$$\begin{aligned}\phi_{,nn}&=\frac{\partial}{\partial n}(\phi_{,n})=n_x\frac{\partial}{\partial x}(\phi_{,n})+n_y\frac{\partial}{\partial y}(\phi_{,n})\\&=n_x\frac{\partial}{\partial x}(n_x\phi_{,x}+n_y\phi_{,y})+n_y\frac{\partial}{\partial y}(n_x\phi_{,x}+n_y\phi_{,y})\\&=n_x^2\phi_{,xx}+2n_xn_y\phi_{,xy}+n_y^2\phi_{,yy}=0\end{aligned} \tag{6.76}$$

where the second derivatives, $\phi_{,xx}$, $\phi_{,xy}$ and $\phi_{,yy}$, are derived using the second order unstructured finite difference scheme as follows (see section 6.7):

$$\phi_{,xx}=\sum_{i=1}^{6}\mathring{c}_i\cdot\phi_i \quad , \quad \phi_{,yy}=\sum_{i=1}^{6}\mathring{d}_i\cdot\phi_i \quad , \quad \phi_{,xy}=\sum_{i=1}^{6}\mathring{e}_i\cdot\phi_i \tag{6.77}$$

where $\mathring{c}_i$, $\mathring{d}_i$ and $\mathring{e}_i$ are constant coefficients based on the mesh geometry.

Therefore, the following relationship is used for the boundary nodes:

$$\phi_{,nn}=n_x^2\sum_{i=1}^{6}\mathring{c}_i\cdot\phi_i+2n_xn_y\sum_{i=1}^{6}\mathring{e}_i\cdot\phi_i+n_y^2\sum_{i=1}^{6}\mathring{d}_i\cdot\phi_i=0 \tag{6.78}$$

The finite difference approach mentioned above is not dependent on the virtual nodes and the Neumann elements.

6.10 Numerical Estimation of the Spatial Gradients

In the present model, the integral of the spatial gradients on a FV cell are computed throughout the convergence theorem, which is usual in the FV models. This procedure links the call-integral of a spatial gradient to the nodal values inside and around a cell. However, there is also another option available when the spatial gradients are needed at a computational node. This option is based on the finite difference method. The issue is simple and straightforward when a Cartesian or curvilinear structured computational grid is used. However it is more complicated for the unstructured grid. Some procedures in this regard are presented by (Ollivier-Gooch, 1997), (Lu and Wai, 1998) and (Magesh and Ruhle, 2003). The method of (Ollivier-Gooch, 1997) has been used herein which is more general and has the capability to produce any arbitrary level of accuracy. A second order version of the algorithm has been used in order to give the same accuracy as the interpolation shape functions. A first order version is also available.

The method of (Ollivier-Gooch, 1997) is based on the two dimensional Taylor series in order to link a nodal value and its spatial derivatives to the neighbouring nodal values: and Taylor series may be written for each surrounding node, i, around a specific node, j:

$$\Delta \phi_i = \Delta x_i \cdot \phi_x + \Delta y_i \cdot \phi_y + \frac{1}{2!}\left((\Delta x_i)^2 \cdot \phi_{xx} + 2\Delta x_i \cdot \Delta y_i \cdot \phi_{xy} + (\Delta y_i)^2 \cdot \phi_{yy}\right) \qquad (6.79)$$

where j is the node in which the derivatives are needed, i is a neighbouring node, $\Delta \phi_i = \phi_i - \phi_j$, $\Delta x_i = x_i - x_j$, and $\Delta y_i = y_i - y_j$. The derivatives ϕ_x, ϕ_y, ϕ_{xx} and ϕ_{yy} are determined at node j.

There are five unknown derivatives in equation (6.79). This equation must be written for a selected set of the neighbouring nodes. At least five nodes are needed in this regard. If the number of nodes is more than five, a weighted least squares optimization algorithm may be used in order to reduce the number equations into five. The relevant equation is:

$$[A]^T \cdot [A] \cdot [X] = [A]^T \cdot [B] \qquad (6.80)$$

in which:

$$[A]_{m\times5}=\begin{bmatrix} \omega_1\cdot\Delta x_1 & \omega_1\cdot\Delta y_1 & \frac{(\omega_1\cdot\Delta x_1)^2}{2} & (\omega_1^2\cdot\Delta x_1\cdot\Delta y_1) & \frac{(\omega_1\cdot\Delta y_1)^2}{2} \\ \omega_2\cdot\Delta x_2 & \omega_2\cdot\Delta y_2 & \frac{(\omega_2\cdot\Delta x_2)^2}{2} & (\omega_2^2\cdot\Delta x_2\cdot\Delta y_2) & \frac{(\omega_2\cdot\Delta y_2)^2}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_m\cdot\Delta x_m & \omega_m\cdot\Delta y_m & \frac{(\omega_m\cdot\Delta x_m)^2}{2} & (\omega_m^2\cdot\Delta x_m\cdot\Delta y_n) & \frac{(\omega_m\cdot\Delta y_m)^2}{2} \end{bmatrix}_{m\times5} \tag{6.81}$$

$$[X]_{5\times1}=\begin{bmatrix} \phi_x \\ \phi_y \\ \phi_{xx} \\ \phi_{xy} \\ \phi_{yy} \end{bmatrix}_{5\times1} \tag{6.82}$$

$$[B]_{m\times1}=\begin{bmatrix} \phi_1-\phi_j \\ \phi_2-\phi_j \\ \vdots \\ \phi_m-\phi_j \end{bmatrix}_{m\times1} \tag{6.83}$$

and m is the number of the selected neighbouring nodes and ω_i is the weighting factor which weights data from more distant nodes less heavily. One suitable choice is $\omega_i=1$.

Equation (6.80) may be solved as follows:

$$[X]=\left([A]^T\cdot[A]\right)^{-1}[A]^T\cdot[B] \tag{6.84}$$

The matrix $\left([A]^T\cdot[A]\right)^{-1}[A]^T$ depends solely on the mesh geometry. It is calculated just once before the solution procedure. In this way, a definite relationship is produced between the x- and y-derivatives of the quantity ϕ at a computational node with the neighbouring nodal values.

The general relationships for the first derivatives are:

$$\begin{aligned} \left(\frac{\partial\phi}{\partial x}\right)_j &= \sum_{i=1}^{m} c_i\cdot(\phi_i-\phi_j)=\sum_{i=1}^{m} c_i\cdot\phi_i-\phi_j\sum_{i=1}^{m} c_i \\ \left(\frac{\partial\phi}{\partial y}\right)_j &= \sum_{i=1}^{m} d_i\cdot(\phi_i-\phi_j)=\sum_{i=1}^{m} d_i\cdot\phi_i-\phi_j\sum_{i=1}^{m} d_i \end{aligned} \tag{6.85}$$

where c_i and d_i are the components of the first and the second rows of the matrix $\left([A]^T\cdot[A]\right)^{-1}[A]^T$ respectively.

The general relationships for the second derivatives are:

$$\left(\frac{\partial^2 \phi}{\partial x^2}\right)_j = \sum_{i=1}^{m} \mathring{c}_i \cdot (\phi_i - \phi_j) = \sum_{i=1}^{m} \mathring{c}_i \cdot \phi_i - \phi_j \sum_{i=1}^{m} \mathring{c}_i$$
$$\left(\frac{\partial^2 \phi}{\partial y^2}\right)_j = \sum_{i=1}^{m} \mathring{d}_i \cdot (\phi_i - \phi_j) = \sum_{i=1}^{m} \mathring{d}_i \cdot \phi_i - \phi_j \sum_{i=1}^{m} \mathring{d}_i \qquad (6.86)$$
$$\left(\frac{\partial^2 \phi}{\partial x \partial y}\right)_j = \sum_{i=1}^{m} \mathring{e}_i \cdot (\phi_i - \phi_j) = \sum_{i=1}^{m} \mathring{e}_i \cdot \phi_i - \phi_j \sum_{i=1}^{m} \mathring{e}_i$$

where $\mathring{c}_i$, $\mathring{d}_i$ and $\mathring{e}_i$ are the components of the third, forth and fifth rows of the matrix $\left([A]^T \cdot [A]\right)^{-1} [A]^T$ respectively.

When the number of the selected neighbouring nodes (m) is equal to five, the matrix equation $[A] \cdot [X] = [B]$ can be solved directly. Therefore the coefficients c_i, d_i, $\mathring{c}_i$, $\mathring{d}_i$ and $\mathring{e}_i$ will be the components of the first to fifth rows of the matrix $[A]^{-1}$.

It is possible to produce a linear version of the above procedure, which is needed for situations with the risk of numerical oscillation. The linear order of the Taylor series may be written for each surrounding node, i, around a specific node, j as follows:

$$\Delta \phi_i = \Delta x_i \cdot \phi_x + \Delta y_i \cdot \phi_y \qquad (6.87)$$

where $\Delta \phi_i = \phi_i - \phi_j$, $\Delta x_i = x_i - x_j$, $\Delta y_i = y_i - y_j$, and all derivatives are evaluated at j.

The two derivatives in equation (6.87) are unknown. They are computed by writing equation (6.87) for a selected set of surrounding nodes. At least two nodes are needed in this regard. If the number of the selected nodes is more than the unknowns, a weighted least squares optimization algorithm should be used as follows:

$$[A]^T \cdot [A] \cdot [X] = [A]^T \cdot [B] \qquad (6.88)$$

where:

$$[A]_{m\times 2} = \begin{bmatrix} \omega_1 \cdot \Delta x_1 & \omega_1 \cdot \Delta y_1 \\ \omega_2 \cdot \Delta x_2 & \omega_2 \cdot \Delta y_2 \\ \vdots & \vdots \\ \omega_m \cdot \Delta x_m & \omega_m \cdot \Delta y_m \end{bmatrix}_{m\times 2}, \quad [X]_{2\times 1} = \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix}_{2\times 1} \text{ and } [B]_{m\times 1} = \begin{bmatrix} \phi_1 - \phi_j \\ \phi_2 - \phi_j \\ \vdots \\ \phi_m - \phi_j \end{bmatrix}_{m\times 1} \qquad (6.89)$$

m is the number of selected surrounding nodes and ω_i is the weighting factor which weights data from more distant nodes less heavily. One suitable choice is $\omega_i = 1$.

The solution for equation (6.88) is:

$$[X] = \left([A]^T \cdot [A]\right)^{-1} [A]^T \cdot [B] \qquad (6.90)$$

The matrix $\left([A]^T\cdot[A]\right)^{-1}[A]^T$ depends only on the computational grid geometry. It is calculated once at the start of a simulation. The participant surrounding nodes for the linear version are different from the surrounding nodes for the second-order procedure. The quadratic algorithm uses the mid-side and the vertex neighbouring nodes from the linked interpolation elements. However, the linear procedure uses just the mid-side element nodes. In this way, the linear FD scheme is identical to the linear interpolation sub-elements described in section (6.4).

The linear relationships for the first derivatives are:

$$\begin{aligned}\left(\frac{\partial\phi}{\partial x}\right)_j&=\sum_{i=1}^{m}c_i\cdot(\phi_i-\phi_j)=\sum_{i=1}^{m}c_i\cdot\phi_i-\phi_j\sum_{i=1}^{m}c_i\\ \left(\frac{\partial\phi}{\partial y}\right)_j&=\sum_{i=1}^{m}d_i\cdot(\phi_i-\phi_j)=\sum_{i=1}^{m}d_i\cdot\phi_i-\phi_j\sum_{i=1}^{m}d_i\end{aligned}\tag{6.91}$$

where c_i and d_i are components of the first and second rows of the matrix, $\left([A]^T\cdot[A]\right)^{-1}[A]^T$, respectively. They are functions of the mesh geometry and are fixed during the solution procedure.

If the number of the selected neighbouring nodes, m, is equal to 2, the matrix equation $[A]\cdot[X]=[B]$ can be solved directly $[X]=[A]^{-1}[B]$.

6.11 The Vertical Discretisation Algorithm

The vertical diffusion step in the model is based on the one dimensional finite volume discretisation method. The finite volume method is based on the interpolation elements and the Lagrangian shape functions described in section (6.5).

Figure (6.11) illustrates three finite volume cells along a grid column (the cells are distinguished by colours). Each finite volume cell involves one computational node. The FV cell is distinguished into two sub-cell areas and each sub-area belongs to one interpolation element. The boundary FV cells have only one sub-cell area.

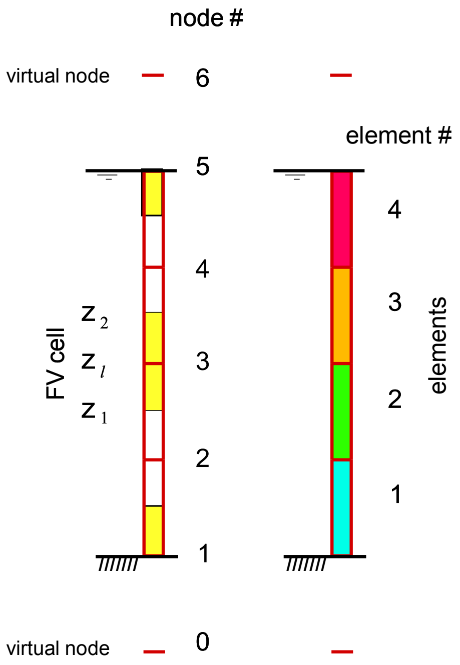


Figure 6.11. One dimensional interpolation elements and FV cells for the vertical direction

Equation (6.17) is used in order to integrate the quantity ϕ on the FV cell:

$$\begin{aligned}\int_{z1}^{z2} \phi(z)\,dz &= \int_{z1}^{z_l} \Big(\sum_{k=l-2}^{l+1} \psi_k^v(z)\cdot\phi_k\Big)dz + \int_{z_l}^{z2} \Big(\sum_{k=l-1}^{l+2} \psi_k^v(z)\cdot\phi_k\Big)dz \\ &= \sum_{k=l-2}^{l+1} \phi_k \int_{z1}^{z_l} \psi_k^v(z)\,dz + \sum_{k=l-1}^{l+2} \phi_k \int_{z_l}^{z2} \psi_k^v(z)\,dz\end{aligned} \tag{6.92}$$

in which l is the node inside the FV cell, $z1$ and $z2$ are the cell boundaries, $\phi(z)$ is the function to be integrated, ϕ_k is the nodal value of $\phi(z)$ at the k^{th} node of the interpolation element which contains the sub-cell area, $\psi_k^v(z)$ is the Lagrangian shape function relevant to the k^{th} node.

The boundary of the FV cell is determined using the following three formulae:

$$z1 = \frac{z_l + z_{l-1}}{2} \quad , \quad z2 = \frac{z_l + z_{l+1}}{2} \qquad for \quad l \neq 1, N \tag{6.93}$$

$$z1 = z_l \quad , \quad z2 = \frac{z_l + z_{l+1}}{2} \qquad for \quad l = 1 \quad \text{(the bed layer's node)} \tag{6.94}$$

$$z1 = \frac{z_l + z_{l-1}}{2} \quad , \quad z2 = z_l \qquad for \quad l = N \quad \text{(the surface layer's node)} \tag{6.95}$$

The integrations in the right hand side of equation (6.92) may be derived analytically as follows:

$$\int_{z1}^{z_l} \psi^v(z)dz = \int_{z1}^{zl} \left[b_1 + b_2(z-z_l) + b_3(z-z_l)^2 + b_4(z-z_l)^3\right] dz$$
$$= -\left(b_1(\Delta z_1) + \frac{1}{2}b_2(\Delta z_1)^2 + \frac{1}{3}b_3(\Delta z_1)^3 + \frac{1}{4}b_4(\Delta z_1)^4\right) \tag{6.96}$$

$$\int_{z_l}^{z2} \psi^v(z)dz = b_1(\Delta z_2) + \frac{1}{2}b_2(\Delta z_2)^2 + \frac{1}{3}b_3(\Delta z_2)^3 + \frac{1}{4}b_4(\Delta z_2)^4 \tag{6.97}$$

where $\Delta z_1 = z1 - z_l$, $\Delta z_2 = z2 - z_l$. The coefficients $b1$ to $b4$ are defined through equations (6.19) to (6.25).

When the Cartesian coordinate system is used, the coefficients in equations (6.96) and (6.97) are not constant in time and not necessarily equivalent for different cells. The reason is that the lengths of the interpolation elements are not necessarily the same and the water depth may be variable. A similar FV procedure in the sigma coordinate system involves constant and unique shape functions.

6.12 Summary

This chapter described the computational grid and the discretisation schemes that have been used as tools in the solution procedures of the present study. Section (6.2) demonstrated the 3D computational grid with the unstructured triangular elements. Section (6.3) presented the general solution algorithm in order to show the role of each discretisation scheme in the overall solution procedure. Section (6.4) described the horizontal interpolation elements used for both the semi-Lagrangian advection interpolation and the analytical integrations needed in the horizontal finite volume scheme. Section (6.5) illustrated the vertical interpolation elements used for both the vertical part of the semi-Lagrangian advection interpolation and the analytical integrations needed in the vertical finite volume scheme. Section (6.6) addressed the complete three-dimensional interpolation procedure used for the advection algorithm. Section (6.7) described the horizontal finite volume discretisation scheme. This scheme is based on analytical integration of the interpolation functions on the FV cells and along their boundaries. Section (6.9) illustrated the lateral boundary conditions and the boundary elements. Section (6.10) described the unstructured finite difference scheme used for numerical estimation of the spatial gradients at the computational nods. Section (6.11) addressed the vertical finite volume discretisation algorithm used in the vertical diffusion algorithm.

7 Model Development

7.1 Introduction

The present chapter includes details about the numerical models and solution algorithms developed and integrated in this study. The present study includes three dimensional and depth averaged two dimensional models for simulating the hydrodynamic and advection diffusion processes. The distributions of salinity level, temperature and suspended sediment concentration are handled through the advection diffusion models. All of the models are based on an unstructured computational grid.

Section (7.2) describes the general algorithm of the 3D hydrodynamic model, which is based on the operator splitting algorithm. Sections (7.3) to (7.9) describe different components of the solution procedure. Section (7.3) gives details of the pure advection step. Section (7.4) deals with the horizontal diffusion step. The free surface solution procedure is described in section (7.5). Section (7.6) deals with the vertical diffusion step. The simulation algorithm for the hydrodynamic pressure correction is the subject of section (7.7). Section (7.8) illustrates the numerical algorithm for the velocity projection into a convergence-free field. Section (7.9) concerns the rare situations where the hydrodynamic pressure or the variation in the bed friction influences the water level. The 3D advection diffusion model is described in section (7.10). The 2D hydrodynamic model is described in section (7.11), which is also based on the fractional step method, identical to the 3D model. Section (7.12) is relevant to the 2D advection diffusion model. The integration procedure of the 2D and the 3D models is described in section (7.13). It includes the integration of the advection steps, integration of the horizontal diffusion steps, and integration of the free surface steps with the bed friction and the free surface slope steps.

7.2 The General Algorithm for the 3D Hydrodynamic Model

The main component of the 3D model is the hydrodynamic model. The salinity, temperature, suspended sediment and turbulence sub-models are also available when required. In the 3D hydrodynamic model the hydrodynamic pressure is taken into account. The solution algorithm for the 3D hydrodynamic model is based on the second type of the projection method and the operator splitting algorithm. The general splitting relationship for a generic quantity, ϕ , is:

$$\frac{\partial \phi}{\partial t} = \underbrace{\frac{\phi^{n+1} - \phi^{d}}{\Delta t}}_{projection\ step} + \underbrace{\frac{\phi^{d} - \phi^{a}}{\Delta t}}_{diffusion\ step} + \underbrace{\frac{\phi^{a} - \phi^{n}}{\Delta t}}_{advection\ step} \tag{7.1}$$

where ϕ^n is the known solution of time step n, ϕ^a is the solution of the advection step, ϕ^d is the solution of the diffusion step including the source terms, and ϕ^{n+1} is the final solution of the new time step.

The diffusion step is partitioned into an explicit horizontal diffusion step and an implicit vertical diffusion one. The free surface is computed implicitly before the vertical diffusion step.

The hydrodynamic solution algorithm contains the following steps:

a) advection step (is three dimensional)

b) horizontal diffusion step, including the source terms except the barotropic pressure gradient (is two dimensional, acts on several layers)

c) implicit free surface step (is two dimensional)

d) vertical diffusion step including the barotropic pressure gradient (is one dimensional)

e) hydrodynamic pressure correction step (is three dimensional, elliptic)

f) velocity projection step

g) water level refining option

Below is a brief description of each solution step.

a) Advection step

In the advection step a 3D hyperbolic equation governing the velocity vector is solved. Three scalar advection equations are solved for the velocity components u, v, and w, as follows:

$$\frac{u^a - u^n}{\Delta t} + u\frac{\partial u}{\partial \tilde{x}} + v\frac{\partial u}{\partial \tilde{y}} + \tilde{w}\frac{\partial u}{\partial \tilde{z}} = 0 \tag{7.2}$$

$$\frac{v^a - v^n}{\Delta t} + u\frac{\partial v}{\partial \tilde{x}} + v\frac{\partial v}{\partial \tilde{y}} + \tilde{w}\frac{\partial v}{\partial \tilde{z}} = 0 \tag{7.3}$$

$$\frac{w^a - w^n}{\Delta t} + u\frac{\partial w}{\partial \tilde{x}} + v\frac{\partial w}{\partial \tilde{y}} + \tilde{w}\frac{\partial w}{\partial \tilde{z}} = 0 \tag{7.4}$$

It is clear from equations (7.2) to (7.4) that the advecting velocity field is u, v, and $\tilde{w}$, while the advected components are u, v, and w. The advection step is described in section (7.3).

b) Horizontal diffusion step

The horizontal diffusion step concerns the horizontal diffusion terms and the source terms of the horizontal momentum equations without a barotropic pressure gradient. It is a two dimensional equation which is solved for the velocity component on the sigma planes. The relevant equations are:

$$\begin{aligned}\frac{u^{d_1}-u^a}{\Delta t}&=\frac{\partial}{\partial\tilde{x}}\left(\nu_h\frac{\partial u^a}{\partial x}\right)+\frac{\partial\tilde{z}}{\partial x}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial u^a}{\partial x}\right)+\frac{\partial}{\partial\tilde{y}}\left(\nu_h\frac{\partial u^a}{\partial y}\right)+\frac{\partial\tilde{z}}{\partial y}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial u^a}{\partial y}\right)\\&-\frac{1}{\rho}\frac{\partial p_a}{\partial\tilde{x}}-g\frac{1}{\rho}\int_z^{\eta}\frac{\partial\rho^n}{\partial x}dz-\frac{1}{\rho}\left(\frac{\partial q^n}{\partial\tilde{x}}+\frac{\partial\tilde{z}}{\partial x}\frac{\partial q^n}{\partial\tilde{z}}\right)\\&+u_s dis_s\end{aligned}\tag{7.5}$$

$$\begin{aligned}\frac{v^{d_1}-v^a}{\Delta t}&=\frac{\partial}{\partial\tilde{x}}\left(\nu_h\frac{\partial v^a}{\partial x}\right)+\frac{\partial\tilde{z}}{\partial x}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial v^a}{\partial x}\right)+\frac{\partial}{\partial\tilde{y}}\left(\nu_h\frac{\partial v^a}{\partial y}\right)+\frac{\partial\tilde{z}}{\partial y}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial v^a}{\partial y}\right)\\&-\frac{1}{\rho}\frac{\partial p_a}{\partial\tilde{y}}-g\frac{1}{\rho}\int_z^{\eta}\frac{\partial\rho^n}{\partial y}dz-\frac{1}{\rho}\left(\frac{\partial q^n}{\partial\tilde{y}}+\frac{\partial\tilde{z}}{\partial y}\frac{\partial q^n}{\partial\tilde{z}}\right)\\&+v_s dis_s\end{aligned}\tag{7.6}$$

$$\begin{aligned}\frac{w^{d_1}-w^a}{\Delta t}&=\frac{\partial}{\partial\tilde{x}}\left(\nu_h\frac{\partial w}{\partial x}\right)+\frac{\partial\tilde{z}}{\partial x}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial w}{\partial x}\right)+\frac{\partial}{\partial\tilde{y}}\left(\nu_h\frac{\partial w}{\partial y}\right)+\frac{\partial\tilde{z}}{\partial y}\frac{\partial}{\partial\tilde{z}}\left(\nu_h\frac{\partial w}{\partial y}\right)\\&-\frac{1}{H}\frac{1}{\rho}\frac{\partial q^n}{\partial\tilde{z}}+w_s dis_s\end{aligned}\tag{7.7}$$

The horizontal diffusion step is described in section (7.4).

c) Free surface step

The free surface elevation is predicted implicitly using three joint equations. They are the depth integrated continuity equation and the result of depth integration of the remaining parts of the horizontal momentum equations. This step includes the vertical diffusion, the barotropic pressure gradient and the Coriolis terms.

The free surface step is described in section (7.5).

d) Vertical diffusion step

The vertical diffusion step concerns the effect of the vertical diffusion of momentum and the barotropic pressure gradient (the free surface slope) on the velocity field. These terms are solved implicitly in order to increase the solution stability. Three one dimensional parabolic equations are solved along the water columns for the components u, v, and w.

The corresponding relationships are:

$$\frac{u^{d_2}-u^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u^{d_2}}{\partial z}\right)-g\frac{\rho_\eta}{\rho}\left(\theta\frac{\partial\eta^{n+1}}{\partial x}+(1-\theta)\frac{\partial\eta^n}{\partial x}\right)+2\cdot v^{d_1}\cdot\omega_z-2\cdot w^{d_1}\cdot\omega_y\tag{7.8}$$

$$\frac{v^{d_2}-v^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial v^{d_2}}{\partial z}\right)-g\frac{\rho_\eta}{\rho}\left(\theta\frac{\partial\eta^{n+1}}{\partial y}+(1-\theta)\frac{\partial\eta^n}{\partial y}\right)-2\cdot u^{d_1}\cdot\omega_z+2\cdot w^{d_1}\cdot\omega_x\tag{7.9}$$

$$\frac{w^{d_2}-w^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial w^{d_2}}{\partial z}\right)+2\cdot u^a\cdot\omega_y-2\cdot v^a\cdot\omega_x \tag{7.10}$$

e) Hydrodynamic pressure correction step

During the pressure correction step the hydrodynamic pressure correction field is determined. It is based on the divergence rate of the intermediate velocity field, see Equation (5.95). It is a three dimensional scalar elliptic equation. The intermediate velocity field at the end of the vertical diffusion step is used herein.

f) Velocity projection step

The final step in calculating the velocity within a time step is to correct the intermediate non-convergence-free velocity field. The hydrodynamic pressure correction field is used in this regard. The following relationships are solved at the computational nodes:

$$u^{n+1}=u^{d_2}-\Delta t\frac{1}{\rho}\frac{\partial\hat{q}}{\partial x} \tag{7.11}$$

$$v^{n+1}=v^{d_2}-\Delta t\frac{1}{\rho}\frac{\partial\hat{q}}{\partial y} \tag{7.12}$$

$$w^{n+1}=w^{d_2}-\Delta t\frac{1}{\rho}\frac{\partial\hat{q}}{\partial z} \tag{7.13}$$

g) Water level refining option:

The water level which has been determined from the implicit free surface step is acceptable for almost most of the hydrodynamic applications. However, it has been computed based on the intermediate velocity field which is not necessarily convergence-free. It is possible to increase the model accuracy to a higher level by re-computing the free surface level on the basis of the convergence free velocity field at the end of a time step. This refining step is optional. It is not necessary because the hydrodynamic part of the pressure field usually has not noticeable influence on the free surface position.

The following sections describe the solution procedure in more details. It should be noted that a hydrostatic-pressure version of the model is also available in which steps (e), (f) and (g) have been omitted and the other steps do not involve the vertical velocity component. The depth integrated continuity equation from the bed to any required z-level is solved in the hydrostatic model in order to calculate the vertical velocity component.

7.3 Advection Step

The pure advection process is addressed within the advection step by solving the advection terms of the momentum equations. The semi-Lagrangian characteristic method has been used in this regard. A brief overview about the semi-Lagrangian algorithms is presented in section (4.4).

Equations (7.2) to (7.4) illustrate the pure advection process. It means that the velocity components u , v and w , as any other scalar quantity, remain constant along the streamlines. The corresponding relationship is:

$$u_i^a = u_p^n \ , \quad v_i^a = v_p^n \ , \quad w_i^a = w_p^n \tag{7.14}$$

where the subscript i stands for the computational node and the subscript p is the departure point (x_p , y_p , $\tilde{z}_p$) of the streamline from the point p at time $n \cdot \Delta t$ to the point i at the time $(n+1) \cdot \Delta t$.

For each node the advection step involves two stages. They are:

a) determining the streamline and the departure point,

b) interpolating data from the surrounding nodes to the departure point at time $n \cdot \Delta t$.

The first stage (a) is handled through backward tracking the streamline explicitly from the computational node i from time $(n+1) \cdot \Delta t$ to time $n \cdot \Delta t$. The backward tracking is based on dividing the time step Δt into $K_{\Delta t}$ sub-steps $\delta t = \Delta t / K_{\Delta t}$.

Therefore, the following iteration is performed:

The first stage:

$$x_0 = x_i , \ u_o = u_i^n , \ y_0 = y_i , \ v_o = v_i^n , \ \tilde{z}_0 = \tilde{z}_i , \ \tilde{w}_o = \tilde{w}_i^n$$

The next stages:

$$\begin{aligned} x_k &= x_{k-1} - \delta t \cdot u_{k-1}^n \ , u_{k-1} = u\left((x, y, \tilde{z})_{k-1}\right) \\ y_k &= y_{k-1} - \delta t \cdot v_{k-1}^n \ , v_{k-1} = v\left((x, y, \tilde{z})_{k-1}\right) \qquad k = 1, \ldots, K_{\Delta t} \\ \tilde{z}_k &= \tilde{z}_{k-1} - \delta t \cdot \tilde{w}_{k-1}^n \ , \tilde{w}_{k-1} = \tilde{w}\left((x, y, \tilde{z})_{k-1}\right) \end{aligned} \tag{7.15}$$

The last stage:

$$x_p = x_K , \ y_p = y_K , \ \tilde{z}_p = \tilde{z}_K$$

An implicit alternative to equation (7.15) is to iterate at each time step, although it is more time consuming. Figure (7.1) illustrates the backward tracking procedure for a computational node.

The second stage (b) is handled via a non-conservative but high-order accurate interpolation scheme in the three dimensional model. This algorithm is an optimal balance of the computation speed and mass conservation for the short and moderate simulation periods. The interpolation equation (6.35) with the shape functions (6.3) to (6.8) and (6.29) to (6.32) has been used in this regard.

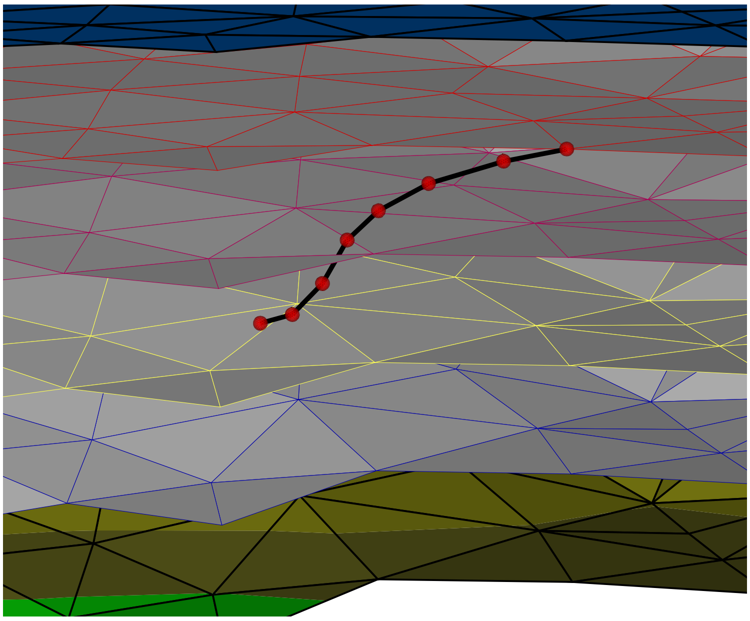


Figure 7.1. The back-tracking procedure for a single node

The clipping algorithm of (Wang et al., 1988) has been used in order to avoid oscillation risk and artificial local extremes in the front of sharp gradients. The clipping algorithm is based on comparing the interpolated data with the maximum and minimum nodal values of an element. The interpolated value must be inside that range. A local weighting procedure is also available which determines the interpolated data by a weighted average of low- and high-order interpolation schemes (Zalesak, S. T., 1979). For a brief review on some oscillation-free semi-Lagrangian algorithms, see (Liu and Dane, 1996).

During the backward tracking procedure, the streamline may cross the domain boundary. It may be a closed boundary or an open input boundary. If the boundary is closed and the boundary condition is not no-slip, the streamline is followed back further along the boundary in the direction of the velocity component up to the previous time step. This is the case for the closed lateral boundary, the bottom boundary and the free surface. For the open input boundary with known velocity, the value at the departure point is determined from the boundary values. Interpolation in time and/or space may be required for the boundary data.

The boundary condition at the input open boundary of a hydrodynamic model may be something other than known velocity. Sometimes it is the water level that is known at the input boundary. The departure points corresponding to the nodes near the input boundary may fall outside the solution domain. It is also the case for the nodes on the boundary. It is possible to use a local model especially for the near-boundary region. It is also possible to use a level-discharge relationship if the boundary is a river section. It may also be possible to use a simple Chezy relationship. Nevertheless, the most practical idea is to locate the input boundary at a location where the velocity is nearly perpendicular to the boundary and the depth integrated velocity (the flux) does not change noticeably along the perpendicular direction. The dominant driving force at the input boundary is the free surface slope. A small error in the advection step does not influences the flow field considerably. The input boundary condition for the advection diffusion problems such as the salinity, temperature or suspended sediment concentration must be of Dirichlet type. It is not possible for the back-ward streamline to cross the outflow open boundary.

7.4 Horizontal Diffusion Step

The horizontal diffusion and other remaining terms, except for the free surface slope and the Coriolis force, are solved in the second step. The free surface slope is solved later in conjunction with the vertical diffusion. The horizontal diffusion step is based on the numerical solution of equations (7.5) to (7.7). These equations are solved layer by layer on the sigma planes.

Following the two dimensional finite volume approach, the above equations are integrated on the FV cells and discretised through the procedure described in section (6.7). The resulting relationships are as follows:

$$
\begin{aligned}
&\frac{1}{\Delta t}\sum_{1}^{m}\sum_{j=1}^{6} u_j^{d_1}\iint\limits_{sp}\psi_j^h\,ds=\frac{1}{\Delta t}\sum_{1}^{m}\sum_{j=1}^{6} u_j^{a}\iint\limits_{sp}\psi_j^h\,ds\\
&+\sum_{1}^{m}\sum_{j=1}^{6}\left(\nu_h\frac{\partial u^a}{\partial x}\right)_j\iint\limits_{sp}\frac{\partial\psi_j^h}{\partial\tilde{x}}\,ds+\sum_{1}^{m}\frac{\partial\tilde{z}}{\partial x}\sum_{j=1}^{6}\left(\frac{\partial(\nu_h(\partial u^a/\partial x))}{\partial\tilde{z}}\right)_j\iint\limits_{sp}\psi_j^h\,ds\\
&+\sum_{1}^{m}\sum_{j=1}^{6}\left(\nu_h\frac{\partial u^a}{\partial y}\right)_j\iint\limits_{sp}\frac{\partial\psi_j^h}{\partial\tilde{y}}\,ds+\sum_{1}^{m}\frac{\partial\tilde{z}}{\partial y}\sum_{j=1}^{6}\left(\frac{\partial(\nu_h(\partial u^a/\partial y))}{\partial\tilde{z}}\right)_j\iint\limits_{sp}\psi_j^h\,ds\\
&-\sum_{1}^{m}\frac{1}{\rho_{sp}}\sum_{j=1}^{6}(p_a)_j\iint\limits_{sp}\frac{\partial\psi_j^h}{\partial\tilde{x}}\,ds-g\sum_{1}^{m}\frac{1}{\rho_{sp}}\sum_{j=1}^{6}\left(\int_{z}^{\eta}\frac{\partial\rho^n}{\partial x}dz\right)_j\iint\limits_{sp}\psi_j^h\,ds\\
&-\sum_{1}^{m}\frac{1}{\rho_{sp}}\sum_{j=1}^{6}q_j^n\iint\limits_{sp}\frac{\partial\psi_j^h}{\partial\tilde{x}}\,ds-\sum_{1}^{m}\frac{1}{\rho_{sp}}\frac{\partial\tilde{z}}{\partial x}\sum_{j=1}^{6}\left(\frac{\partial q^n}{\partial\tilde{z}}\right)_j\iint\limits_{sp}\psi_j^h\,ds+u_s dis_s
\end{aligned}
\tag{7.16}
$$

$$\begin{aligned}
&\frac{1}{\Delta t}\sum_{1}^{m}\sum_{j=1}^{6} v_j^{d_1}\iint_{sp}\psi_j^h\,ds=\frac{1}{\Delta t}\sum_{1}^{m}\sum_{j=1}^{6} v_j^{a}\iint_{sp}\psi_j^h\,ds\\
&+\sum_{1}^{m}\sum_{j=1}^{6}\left(\nu_h\frac{\partial v^a}{\partial x}\right)_j\iint_{sp}\frac{\partial\psi_j^h}{\partial\tilde{x}}\,ds+\sum_{1}^{m}\frac{\partial\tilde{z}}{\partial x}\sum_{j=1}^{6}\left(\frac{\partial(\nu_h(\partial v^a/\partial x))}{\partial\tilde{z}}\right)_j\iint_{sp}\psi_j^h\,ds\\
&+\sum_{1}^{m}\sum_{j=1}^{6}\left(\nu_h\frac{\partial v^a}{\partial y}\right)_j\iint_{sp}\frac{\partial\psi_j^h}{\partial\tilde{y}}\,ds+\sum_{1}^{m}\frac{\partial\tilde{z}}{\partial y}\sum_{j=1}^{6}\left(\frac{\partial(\nu_h(\partial v^a/\partial y))}{\partial\tilde{z}}\right)_j\iint_{sp}\psi_j^h\,ds\\
&-\sum_{1}^{m}\frac{1}{\rho_{sp}}\sum_{j=1}^{6}(p_a)_j\iint_{sp}\frac{\partial\psi_j^h}{\partial\tilde{y}}\,ds-g\sum_{1}^{m}\frac{1}{\rho_{sp}}\sum_{j=1}^{6}\left(\int_{z}^{\eta}\frac{\partial\rho^n}{\partial y}dz\right)_j\iint_{sp}\psi_j^h\,ds\\
&-\sum_{1}^{m}\frac{1}{\rho_{sp}}\sum_{j=1}^{6}q_j^n\iint_{sp}\frac{\partial\psi_j^h}{\partial\tilde{y}}\,ds-\sum_{1}^{m}\frac{1}{\rho_{sp}}\frac{\partial\tilde{z}}{\partial y}\sum_{j=1}^{6}\left(\frac{\partial q^n}{\partial\tilde{z}}\right)_j\iint_{sp}\psi_j^h\,ds+u_s dis_s
\end{aligned}\qquad(7.17)$$

$$\begin{aligned}
&\frac{1}{\Delta t}\sum_{1}^{m}\sum_{j=1}^{6} w_j^{d_1}\iint_{sp}\psi_j^h\,ds=\frac{1}{\Delta t}\sum_{1}^{m}\sum_{j=1}^{6} w_j^{a}\iint_{sp}\psi_j^h\,ds\\
&+\sum_{1}^{m}\sum_{j=1}^{6}\left(\nu_h\frac{\partial w^a}{\partial x}\right)_j\iint_{sp}\frac{\partial\psi_j^h}{\partial\tilde{x}}\,ds+\sum_{1}^{m}\frac{\partial\tilde{z}}{\partial x}\sum_{j=1}^{6}\left(\frac{\partial(\nu_h(\partial w^a/\partial x))}{\partial\tilde{z}}\right)_j\iint_{sp}\psi_j^h\,ds\\
&+\sum_{1}^{m}\sum_{j=1}^{6}\left(\nu_h\frac{\partial w^a}{\partial y}\right)_j\iint_{sp}\frac{\partial\psi_j^h}{\partial\tilde{y}}\,ds+\sum_{1}^{m}\frac{\partial\tilde{z}}{\partial y}\sum_{j=1}^{6}\left(\frac{\partial(\nu_h(\partial w^a/\partial y))}{\partial\tilde{z}}\right)_j\iint_{sp}\psi_j^h\,ds\\
&-\sum_{1}^{m}\frac{1}{\rho_{sp}}\sum_{j=1}^{6}\left(\frac{1}{H}\frac{\partial q^n}{\partial\tilde{z}}\right)_j\iint_{sp}\psi_j^h\,ds+w_s dis_s
\end{aligned}\qquad(7.18)$$

where sp stands for the sub-cell triangles in a FV cell, m is the number of sub-cells, and ρ_{sp} is the averaged density in the sub cell. The quantities $\partial\tilde{z}/\partial x$ and $\partial\tilde{z}/\partial y$ are constant in the sub cell triangles.

The vertical gradient terms in the above equations are determined by the finite difference scheme along the vertical direction. The nodal values of the quantities $\nu^h\partial u/\partial x$ and $\nu^h\partial u/\partial y$ in equation (7.16) are the stress components due to the horizontal diffusion. They are computed in the Cartesian coordinate system, not the sigma system, in order to prevent inaccuracy in stratified flow conditions with considerable bed slope. The same procedure has been used for computing the horizontal gradient of the water density $\partial\rho/\partial x$.

The terms $\nu^h\partial u/\partial x$, $\nu^h\partial u/\partial y$, $\partial\rho/\partial x$ and $\partial\rho/\partial y$ are determined via two stages, including:

a) vertical interpolation along the surrounding grid columns in order to define the corresponding values at the same z level,

b) computing the spatial derivatives in the horizontal plane.

The first stage (a) is handled through the vertical interpolation relationships described in section (6.5). If the z value along a neighbouring column falls below the bed level or above the surface level, the corresponding point is neglected. The second stage (b) may be handled through the FV divergence theorem for the spatial gradients or through the unstructured finite difference algorithm discussed in section (6.10). The terms of explicit hydrodynamic pressure gradients are optional (they are active if projection method-2 is used instead of projection method-1).

The cell integrals in equations (7.16) to (7.18) have been derived analytically in section (6.7). Applying each one of the above equations to the FV cells of a layer in addition to the boundary conditions results in a set of linear equations of the form ($[\![A]\!][x]=[b]$). The coefficient matrix, $[\![A]\!]$, is constant during the simulation procedure. The inverse matrix, $[\![A]\!]^{-1}$, is computed once and is used for every time step in order to reduce computational time .

In spite of the fact that equations (7.16) to (7.18) have been discretised explicitly, the solution procedure here is based on a matrix solver. This is the characteristics of the hybrid finite volume finite element method and will result in increased model accuracy. With most of the usual finite volume algorithms it is assume that the shape of the unknown quantity over the cells (for example the gradient) at the new time step is equivalent to its shape at the previous time step. The explicit version of such a scheme not only defines the diffusion terms explicitly, but also defines the over-cell variation of the unknown variable at the new time step explicitly. The present method, on the other hand, defines the over-cell variation of the quantity at the new time step implicitly. It is possible to eliminate the need for a matrix solver by adding a similar simplification assumption. However, there is no need for it, because the present scheme uses an efficient matrix solver nearly as fast as the explicit procedure.

The horizontal diffusion equation is solved for the velocity components, u , v and w layer by layer. Each velocity component has boundary conditions at the lateral boundaries. The kinematic boundary condition links the velocity components u and v at the closed boundaries. The horizontal velocity components u and v in the present study are solved together via solving a larger system of equations. The larger system is created by combining the u and v - system of equations and adding the kinematic closed boundary conditions. In this way, the dependency of u and v at the boundaries is addressed implicitly. The linear system is solved by inverting its coefficient matrix, which is constant in time. The shape of the solved matrix equation is as follows:

$$\begin{bmatrix} u_1 \\ . \\ u_N \\ v_1 \\ . \\ v_N \end{bmatrix}_{2\mathrm{N}\times 1} = [A^{-1}]_{2\mathrm{N}\times 2\mathrm{N}} \times \begin{bmatrix} rhs_{u_1} \\ . \\ rhs_{u_N} \\ rhs_{v_1} \\ . \\ rhs_{v_N} \end{bmatrix}_{2\mathrm{N}\times 1} \tag{7.19}$$

where the coefficient matrix may be fractioned into four sub-matrices as follows:

$$[A^{-1}]_{2\mathrm{N}\times 2\mathrm{N}} = \begin{bmatrix} [I] & [II] \\ [III] & [IV] \end{bmatrix} \tag{7.20}$$

and the large matrix solution may be partitioned as follows:

$$\begin{gathered} \begin{bmatrix} u_1 \\ . \\ u_N \end{bmatrix}_{N\times 1} = [I]_{N\times N} \times \begin{bmatrix} rhs_{u_1} \\ . \\ rhs_{u_N} \end{bmatrix}_{N\times 1} + [II]_{N\times N} \times \begin{bmatrix} rhs_{v_1} \\ . \\ rhs_{v_N} \end{bmatrix}_{N\times 1} \\ \text{and} \\ \begin{bmatrix} v_1 \\ . \\ v_N \end{bmatrix}_{N\times 1} = [III]_{N\times N} \times \begin{bmatrix} rhs_{u_1} \\ . \\ rhs_{u_N} \end{bmatrix}_{N\times 1} + [IV]_{N\times N} \times \begin{bmatrix} rhs_{v_1} \\ . \\ rhs_{v_N} \end{bmatrix}_{N\times 1} \end{gathered} \tag{7.21}$$

Other quantities such as the vertical velocity component, the salinity and the temperature are solved separately.

It is common in coastal engineering studies to use simplified assumptions for the velocity at lateral boundaries, because they are generally located in shallow water regions. However, there are applications in which the water depth is not shallow near the closed lateral boundaries. This type of boundary is common in dam reservoir modelling. Considering that the nodal columns are directed vertically, the lateral boundaries are vertical and the boundary condition for the vertical velocity component is independent from the horizontal velocity components.

The dynamic BC for the closed lateral boundary may be no-slip, free-slip or wall-function. The free-slip condition for the vertical velocity component is:

$$\frac{\partial w}{\partial n} = 0 \;\Rightarrow\; \frac{\partial w}{\partial \tilde{n}} = -\frac{\partial \tilde{z}}{\partial n}\frac{\partial w^n}{\partial \tilde{z}} \tag{7.22}$$

The following relationship may be used as the wall-function boundary condition for the vertical velocity component:

$$\frac{\tau_{wall,zn}}{\rho}=\nu_t\frac{\partial w}{\partial n} \tag{7.23}$$

with the following relationships for $\tau_{wall,zn}/\rho$ and ν_t :

$$\nu_t=\kappa\, d_o\frac{|\vec{V}|}{\frac{1}{\kappa}\ln\left(\frac{d_o}{(k_s/30)}\right)} \tag{7.24}$$

$$\frac{\tau_{wall,zn}}{\rho}=-\frac{|\vec{V}|\, w}{\left(\frac{1}{\kappa}\ln\left(\frac{d_o}{(k_s/30)}\right)\right)^2} \tag{7.25}$$

where $\kappa=0.41$ is the von Karman's constant, k_s is the roughness height, and d_0 is the normal distance between the boundary point and the wall. The boundary node is assumed to be within the wall-function region. The wall is assumed to be rough.

Equations (7.24) and (7.25) are inserted into equation (7.23) to get:

$$\frac{\partial w}{\partial n}=\frac{w_{do}}{d_o\cdot\ln\left(\frac{d_o}{(k_s/30)}\right)} \tag{7.26}$$

The vector $\vec{n}$ is horizontal while the sigma layer is not necessarily horizontal at the boundary. This phenomenon might be important in the mountainous reservoir simulation. The right hand side of equation (7.26) is transformed into the sigma coordinate system:

$$\frac{\partial w}{\partial\tilde{n}}+\frac{\partial\tilde{z}}{\partial n}\frac{\partial w}{\partial\tilde{z}}=\frac{w_{do}}{d_o\cdot\ln\left(\frac{d_o}{(k_s/30)}\right)} \quad\Rightarrow\quad \frac{\partial w}{\partial\tilde{n}}-\frac{w}{d_o\cdot\ln\left(\frac{d_o}{(k_s/30)}\right)}=-\frac{\partial\tilde{z}}{\partial n}\frac{\partial w^n}{\partial\tilde{z}} \tag{7.27}$$

where the direction $\tilde{n}$ is the intersection of the zn -plane and the sigma plane.

The spatial gradient, $\partial/\partial\tilde{n}$, in equation (7.27) may be discretised and linked to the neighbouring computational nodes in the sigma plane, while it is not possible for $\partial/\partial n$. The algorithm for handling equation (7.27) as the lateral boundary condition has been discussed in section (6.9).

The kinematic boundary condition in the horizontal plane may be written as follows:

$$V_n=0 \quad\Rightarrow\quad n_x u+n_y v=0 \tag{7.28}$$

where the vector (n_x,n_y) is the unit outward vector normal to the boundary and V_n is the velocity component normal to the boundary.

Generally, neither of the horizontal velocity components is parallel to the boundary. The parallel component of the horizontal velocity for the free-slip condition is derived as follows:

$$\frac{\partial V_h^{\tan}}{\partial n}=0 \;\Rightarrow\; \frac{\partial V_h^{\tan}}{\partial \tilde{n}}=-\frac{\partial \tilde{z}}{\partial n}\frac{\partial (V_h^{\tan})^n}{\partial \tilde{z}} \qquad (7.29)$$

where $V_h^{\tan}=-n_y u+n_x v$ is the horizontal component of the 3D velocity parallel to the boundary.

The wall-function boundary condition may be written as follows:

$$\frac{\partial V_h^{\tan}}{\partial n}=\frac{(V_h^{\tan})_{do}}{d_o \cdot \ln\left(\frac{d_o}{(k_s/30)}\right)} \;\Rightarrow\; \frac{\partial V_h^{\tan}}{\partial \tilde{n}}-\frac{V_h^{\tan}}{d_o \cdot \ln\left(\frac{d_o}{(k_s/30)}\right)}=-\frac{\partial \tilde{z}}{\partial n}\frac{\partial (V_h^{\tan})^n}{\partial \tilde{z}} \qquad (7.30)$$

For the “no-slip” condition, the computational grid near the boundary must be dense enough in order to simulate the boundary layer accurately. A wall function is needed in this case in order to estimate the near-wall eddy viscosity. If the no-slip condition is used without dense mesh near the boundary, unrealistic large shear stress may be transmitted from the boundary to the fluid. This inaccuracy results in under-prediction of the flow field (Li and Falconer, 1995).

In coastal modelling applications, the lateral closed boundaries are shallow and the horizontal grid-size is much larger than the vertical one, therefore the wall function is no longer adequate. The free slip condition is usually acceptable for such a condition. It is the bed roughness, not the shore-line velocity condition that controls the velocity in shallow regions.

For the open boundary, if there is no other data, the velocity gradient normal to the boundary may be assumed to be zero or not to change in the normal direction. This condition may be used for each velocity component. It is handled easily using the algorithm described in section (6.9). If one of the velocity components is known at the open boundary, that data may be used as a Dirichlet condition. Usually, the vertical velocity component may be assumed to be zero at the open boundary. The horizontal velocity component parallel to the boundary is also negligible for many cases.

The idea of soft start is used in the current study in order to avoid unrealistic shock waves in the computational domain. This technique is usually sufficient for avoiding the need for a non-reflective condition at the open boundary.

7.5 Free Surface Step

The present model handles the barotropic pressure gradient terms during the vertical diffusion step. Therefore, the free surface is determined just before the vertical diffusion step. It is essential that the barotropic pressure gradient terms are computed implicitly in order to increase model stability and reduce the time step limitation. In the present study the depth integrated continuity equation is used in this regard, as described in section (3.3.2.5).

The depth-integrated continuity equation and the horizontal momentum equations are solved jointly. The velocity gradient terms in the continuity equation and the free surface slope terms in the momentum equations are discretised implicitly while the remaining terms are handled explicitly. The advection and the horizontal diffusion steps have been computed beforehand, therefore it is sufficient to solve just the remaining parts of the momentum equations in conjunction with the continuity equation.

The following equations must be solved together:

$$\begin{aligned}-\frac{\partial\eta}{\partial t}&=-\frac{\eta^{n+1}-\eta^{n}}{\Delta t}\\&=\theta\left(\frac{\partial}{\partial x}\left(\int_{z_b}^{\eta}u^{n+1}\,dz\right)+\frac{\partial}{\partial y}\left(\int_{z_b}^{\eta}v^{n+1}\,dz\right)\right)+(1-\theta)\left(\frac{\partial}{\partial x}\left(\int_{z_b}^{\eta^n}u^{n}\,dz\right)+\frac{\partial}{\partial y}\left(\int_{z_b}^{\eta^n}v^{n}\,dz\right)\right)\\&\approx\theta\left(\frac{\partial}{\partial x}\left(\int_{z_b}^{\eta}u^{d_2}\,dz\right)+\frac{\partial}{\partial y}\left(\int_{z_b}^{\eta}v^{d_2}\,dz\right)\right)+(1-\theta)\left(\frac{\partial}{\partial x}\left(\int_{z_b}^{\eta^n}u^{n}\,dz\right)+\frac{\partial}{\partial y}\left(\int_{z_b}^{\eta^n}v^{n}\,dz\right)\right)\end{aligned}\qquad(7.31)$$

$$\frac{u^{d_2}-u^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u}{\partial z}\right)-g\frac{\rho_\eta}{\rho}\left(\theta\frac{\partial\eta^{n+1}}{\partial x}+(1-\theta)\frac{\partial\eta^{n}}{\partial x}\right)+2\cdot\omega_z\cdot v^{d_1}-2\cdot\omega_y\cdot w^{d_1}\qquad(7.32)$$

$$\frac{v^{d_2}-v^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial v}{\partial z}\right)-g\frac{\rho_\eta}{\rho}\left(\theta\frac{\partial\eta^{n+1}}{\partial y}+(1-\theta)\frac{\partial\eta^{n}}{\partial y}\right)-2\cdot\omega_z\cdot u^{d_1}+2\cdot\omega_x\cdot w^{d_1}\qquad(7.33)$$

where θ is the implicitness factor, η is the free surface level at the new time step, and η^n is the free surface at the previous time step.

Replacing u^{n+1} and v^{n+1} with u^{d_2} and v^{d_2} in equation (7.31) can only obtain an approximation, because the intermediate velocity field is not necessarily convergence-free. However, it is an acceptable solution because at least the explicit value of the hydrodynamic pressure gradient has been taken into account at the horizontal diffusion step. On the other hand the hydrodynamic pressure does not influences the water level considerably. The widely-used algorithm of Stansby and Zhao (1998) involves the same approximation.

Equations (7.32) and (7.33) are re-arranged and integrated vertically along the water depth in order to give the same velocity terms as participated in equation (7.31). The procedure for solving equation (7.32) is given below:

$$\begin{aligned} u^{d_2} = {} & u^{d_1} + \Delta t \frac{\partial}{\partial z}\left(\nu_v \frac{\partial u}{\partial z}\right) - \Delta t\, g \frac{\rho_\eta}{\rho} \theta \frac{\partial \eta^{n+1}}{\partial x} - \Delta t\, g \frac{\rho_\eta}{\rho}(1-\theta)\frac{\partial \eta^n}{\partial x} \\ & + 2\,\omega_z \cdot v^{d_1} \Delta t - 2\,\omega_y \cdot w^{d_1} \Delta t \end{aligned} \tag{7.34}$$

Therefore:

$$\begin{aligned} \int_{z_b}^{\eta} u^{d_2} dz = {} & \int_{z_b}^{\eta} u^{d_1} dz + \int_{z_b}^{\eta} \Delta t \frac{\partial}{\partial z}\left(\nu_v \frac{\partial u}{\partial z}\right) dz \\ & - \int_{z_b}^{\eta} \Delta t\, g \frac{\rho_\eta}{\rho} \theta \frac{\partial \eta^{n+1}}{\partial x} dz - \int_{z_b}^{\eta} \Delta t\, g \frac{\rho_\eta}{\rho}(1-\theta)\frac{\partial \eta^n}{\partial x} dz \\ & + 2 \cdot \omega_z \Delta t \int_{z_b}^{\eta} v^{d_1} dz - 2 \cdot \omega_y \Delta t \int_{z_b}^{\eta} w^{d_1} dz \end{aligned} \tag{7.35}$$

or:

$$\begin{aligned} \int_{z_b}^{\eta} u^{d_2} dz = {} & \int_{z_b}^{\eta} u^{d_1} dz + \Delta t \int_{z_b}^{\eta} \frac{\partial}{\partial z}\left(\nu_v \frac{\partial u}{\partial z}\right) dz \\ & - \Delta t\, g\, \theta \frac{\partial \eta^{n+1}}{\partial x} \int_{z_b}^{\eta} \frac{\rho_\eta}{\rho} dz - \Delta t\, g (1-\theta) \frac{\partial \eta^n}{\partial x} \int_{z_b}^{\eta} \frac{\rho_\eta}{\rho} dz \\ & + 2 \cdot \omega_z \Delta t \int_{z_b}^{\eta} v^{d_1} dz - 2 \cdot \omega_y \Delta t \int_{z_b}^{\eta} w^{d_1} dz \end{aligned} \tag{7.36}$$

The depth integrated vertical diffusion term in equation (7.36) may be defined as follows:

$$\int_{z_b}^{\eta} \frac{\partial}{\partial z}\left(\nu_v \frac{\partial u}{\partial z}\right) dz = \left(\nu_v \frac{\partial u}{\partial z}\right)_\eta - \left(\nu_v \frac{\partial u}{\partial z}\right)_{z_b} = \frac{(\tau_x)_\eta}{\rho_\eta} - \frac{(\tau_x)_b}{\rho_b} \tag{7.37}$$

where $(\tau_x)_\eta$ is positive if $(U_x)_{wind} > (u_x)_\eta$ and $(\tau_x)_b$ is positive if u_b is positive.

Inserting equation (7.37) in equation (7.36) results in the following relationship:

$$\begin{aligned} \int_{z_b}^{\eta} u^{d_2} dz = {} & \int_{z_b}^{\eta} u^{d_1} dz + \Delta t \left(\frac{(\tau_x)_\eta}{\rho_\eta} - \frac{(\tau_x)_b}{\rho_b} \right) \\ & - \Delta t\, g\, \theta \frac{\partial \eta^{n+1}}{\partial x} \Pi - \Delta t\, g (1-\theta) \frac{\partial \eta^n}{\partial x} \Pi \\ & + 2 \cdot \omega_z \Delta t \int_{z_b}^{\eta} v^{d_1} dz - 2 \cdot \omega_y \Delta t \int_{z_b}^{\eta} w^{d_1} dz \end{aligned} \tag{7.38}$$

where:

$$\Pi = \int_{z_b}^{\eta^{n+1}} \frac{\rho_\eta}{\rho_z} dz \tag{7.39}$$

The parameter Π is equal to the water depth if the density is homogeneous.

Therefore the resulting equations to be solved for determine the free surface are:

$$
\begin{aligned}
&-\frac{\eta^{n+1}-\eta^{n}}{\Delta t}= \\
&\theta\left(\frac{\partial}{\partial x}\left(\int_{z_b}^{\eta} u^{d_2}\,dz\right)+\frac{\partial}{\partial y}\left(\int_{z_b}^{\eta} v^{d_2}\,dz\right)\right)+(1-\theta)\left(\frac{\partial}{\partial x}\left(\int_{z_b}^{\eta^n} u^{n}\,dz\right)+\frac{\partial}{\partial y}\left(\int_{z_b}^{\eta^n} v^{n}\,dz\right)\right)
\end{aligned}
\tag{7.40}
$$

$$
\begin{aligned}
&\int_{z_b}^{\eta} u^{d_2}\,dz=\int_{z_b}^{\eta} u^{d_1}\,dz+\Delta t\left(\frac{(\tau_x)_\eta}{\rho_\eta}-\frac{(\tau_x)_b}{\rho_b}\right) \\
&-\Delta t\, g\,\theta\frac{\partial\eta^{n+1}}{\partial x}\Pi-\Delta t\, g\,(1-\theta)\frac{\partial\eta^{n}}{\partial x}\Pi \\
&+2\cdot\omega_z\Delta t\int_{z_b}^{\eta} v^{d_1}\,dz-2\cdot\omega_y\Delta t\int_{z_b}^{\eta} w^{d_1}\,dz
\end{aligned}
\tag{7.41}
$$

$$
\begin{aligned}
&\int_{z_b}^{\eta} v^{d_2}\,dz=\int_{z_b}^{\eta} v^{d_1}\,dz+\Delta t\left(\frac{(\tau_y)_\eta}{\rho_\eta}-\frac{(\tau_y)_b}{\rho_b}\right) \\
&-\Delta t\, g\,\theta\frac{\partial\eta^{n+1}}{\partial y}\Pi-\Delta t\, g\,(1-\theta)\frac{\partial\eta^{n}}{\partial y}\Pi \\
&-2\cdot\omega_z\Delta t\int_{z_b}^{\eta} u^{d_1}\,dz+2\cdot\omega_x\Delta t\int_{z_b}^{\eta} w^{d_1}\,dz
\end{aligned}
\tag{7.42}
$$

Two approaches are common in the literature for solving equations (7.40) to (7.42). The first one is to solve the three equations jointly by writing their discretised form for each computational node and solving a linear system of equations that has three equations and three unknowns for every node. The second approach is to derive the x- and y-gradients of the depth integrated velocity components by deriving equations (7.41) and (7.42), then insert the results in equation (7.40). This approach leads to a single unknown and single equation at each computational node which includes not only the water level but also the curvature of the water surface. The present model is based on the first approach. Handling the boundary conditions is straightforward in this algorithm and it is free from the water surface curvature, which is more sensitive to the truncation errors than the water surface gradient.

Equations (7.40) to (7.42) may be written as following:

$$
\eta^{n+1}+\Delta t\,\theta\left(\frac{\partial}{\partial x}(DU^{d_2})+\frac{\partial}{\partial y}(DV^{d_2})\right)=\eta^{n}-\Delta t(1-\theta)\left(\frac{\partial}{\partial x}(D^{n}U^{n})+\frac{\partial}{\partial y}(D^{n}V^{n})\right) \tag{7.43}
$$

$$
\begin{aligned}
&(DU^{d_2})+\Delta t\, g\,\theta\frac{\partial\eta^{n+1}}{\partial x}\Pi=(DU^{d_1})-\Delta t\, g\,(1-\theta)\frac{\partial\eta^{n}}{\partial x}\Pi \\
&+\Delta t\left(\frac{(\tau_x)_\eta}{\rho_\eta}-\frac{(\tau_x)_b}{\rho_b}\right)+2\cdot\omega_z\Delta t(DV^{d_1})-2\cdot\omega_y\Delta t(DW^{d_1})
\end{aligned}
\tag{7.44}
$$

$$\begin{aligned}&(DV^{d_2})+\Delta t\, g\,\theta\frac{\partial \eta^{n+1}}{\partial y}\Pi = (DV^{d_1})-\Delta t\, g(1-\theta)\frac{\partial \eta^{n}}{\partial y}\Pi\\&+\Delta t\left(\frac{(\tau_y)_\eta}{\rho_\eta}-\frac{(\tau_y)_b}{\rho_b}\right)-2\cdot\omega_z\Delta t(DU^{d_1})+2\cdot\omega_x\Delta t(DW^{d_1})\end{aligned}\tag{7.45}$$

where DU, DV and DW are the depth integrated velocity components in the x and y directions, respectively. All of the unknown parameters, η^{n+1}, DU^{d_2} and DV^{d_2}, are in the left hand side of the above equations and everything in the right hand sides is known. The known values of D and Π are used in order to linearise the above equations.

Equation (7.43) may be written in the following form:

$$\eta^{n+1}+\Delta t\,\theta\,\nabla\cdot\left(DU^{d_2}, DV^{d_2}\right)=\eta^{n}-\Delta t(1-\theta)\nabla\cdot\left(D^n U^n, D^n V^n\right)\tag{7.46}$$

Equations (7.46), (7.44) and (7.45) may be integrated on the 2D FV cells in order to be solved using the finite volume algorithm. The integral form of equation (7.46) is:

$$\begin{aligned}&\iint_S \eta^{n+1}\,dA+\Delta t\,\theta\iint_S \nabla\cdot\left(DU^{d_2}, DV^{d_2}\right)dA=\\&\iint_S \eta^{n}\,dA-\Delta t(1-\theta)\iint_S \nabla\cdot\left(D^n U^n, D^n V^n\right)dA\end{aligned}\tag{7.47}$$

Applying the divergence theorem on equation (7.47) results in the following relationship:

$$\begin{aligned}&\iint_S \eta^{n+1}dA+\Delta t\,\theta\int_{\partial S}\left(n_x(DU^{d_2})+n_y(DV^{d_2})\right)ds=\\&\iint_S \eta^{n}\,dA-\Delta t(1-\theta)\int_{\partial S}\left(n_x(D^n U^n)+n_y(D^n V^n)\right)ds\end{aligned}\tag{7.48}$$

The integral form of equations (7.44) and (7.45) are:

$$\begin{aligned}&\iint_S (DU^{d_2})dA+\Delta t\, g\,\theta\,\Pi\iint_S\frac{\partial \eta^{n+1}}{\partial x}ds=\\&\iint_S (DU^{d_1})ds-\Delta t\, g(1-\theta)\Pi\iint_S\frac{\partial \eta^{n}}{\partial x}ds+\Delta t\iint_S (\Delta\tau)_x\,ds\\&+2\Delta t\iint_S\left(\omega_z(DV^{d_1})-\omega_y(DW^{d_1})\right)\end{aligned}\tag{7.49}$$

$$\begin{aligned}&\iint_S (DV^{d_2})ds+\Delta t\, g\theta\,\Pi\iint_S\frac{\partial \eta^{n+1}}{\partial y}ds=\\&\iint_S (DV^{d_1})ds+\Delta t\iint_S (\Delta\tau)_y\,ds-\Delta t\, g(1-\theta)\Pi\iint_S\frac{\partial \eta^{n}}{\partial y}ds\\&-2\Delta t\iint_S\left(\omega_z(DU^{d_1})-\omega_x(DW^{d_1})\right)\end{aligned}\tag{7.50}$$

where:

$$(\Delta\tau_x)=\frac{(\tau_x)_\eta}{\rho_\eta}-\frac{(\tau_x)_b}{\rho_b}\,,\qquad(\Delta\tau_y)=\frac{(\tau_y)_\eta}{\rho_\eta}-\frac{(\tau_y)_b}{\rho_b}\tag{7.51}$$

The free surface slope terms in equations (7.49) and (7.50) may be addressed by the divergence theorem as following:

$$\iint_S \frac{\partial \eta}{\partial x} ds = \iint_S \vec{\nabla} \cdot \overrightarrow{\left(\eta \vec{e}_x , 0 \vec{e}_y\right)} \, ds = \int_{\partial S} \left(\vec{n} \cdot \vec{e}_x \eta\right) dl = \int_{\partial S} \left(n_x \eta\right) dl \tag{7.52}$$

$$\iint_S \frac{\partial \eta}{\partial y} ds = \iint_S \vec{\nabla} \cdot \overrightarrow{\left(0 \vec{e}_x , \eta \vec{e}_y\right)} \, ds = \int_{\partial S} \left(\vec{n} \cdot \vec{e}_y \eta\right) dl = \int_{\partial S} \left(n_y \eta\right) dl \tag{7.53}$$

Following the FV algorithm described in sections (6.7) and (6.8), the discretised form of equations (7.48) to (7.50) may be written as follows:

$$\begin{aligned}
&\sum_1^m \sum_{j=1}^6 \eta_j^{n+1} \iint_{sp} \psi_j \, ds \\
&+\Delta t \theta \sum_{i=1}^m \sum_{j=1}^6 \left((n_x)_i \, (DU^{d_2})_j + (n_y)_i \, (DV^{d_2})_j\right) \int_{\partial s_i} \psi_j \, dl = \\
&\sum_1^m \sum_{j=1}^6 \eta_j^n \iint_{sp} \psi_j \, ds \\
&-\Delta t (1-\theta) \sum_{i=1}^m \sum_{j=1}^6 \left((n_x)_i \, (D^n U^n)_j + (n_y)_i \, (D^n V^n)_j\right) \int_{\partial s_i} \psi_j \, ds
\end{aligned} \tag{7.54}$$

$$\begin{aligned}
&\sum_1^m \sum_{j=1}^6 (DU^{d_2})_j \iint_{sp} \psi_j \, ds + \Delta t \, g \theta \Pi \sum_{i=1}^m \sum_{j=1}^6 \eta_j^{n+1} (n_x)_i \int_{\partial s_i} \psi_j \, dl = \\
&\sum_1^m \sum_{j=1}^6 (DU^{d_1})_j \iint_{sp} \psi_j \, ds - \Delta t \, g (1-\theta) \Pi \sum_{i=1}^m \sum_{j=1}^6 \eta_j^n (n_x)_i \int_{\partial s_i} \psi_j \, dl \\
&+\Delta t \sum_1^m \sum_{j=1}^6 (\Delta \tau_x)_j \iint_{sp} \psi_j \, ds + 2 \cdot \Delta t \sum_1^m \sum_{j=1}^6 \left(\omega_z \cdot DV_j^{d_1} - \omega_y \cdot DW_j^{d_1}\right) \iint_S \psi_j \, ds
\end{aligned} \tag{7.55}$$

$$\begin{aligned}
&\sum_1^m \sum_{j=1}^6 (DV^{d_2})_j \iint_{sp} \psi_j \, ds + \Delta t \, g \theta \Pi \sum_{i=1}^m \sum_{j=1}^6 \eta_j^{n+1} (n_y)_i \int_{\partial s_i} \psi_j \, dl = \\
&\sum_1^m \sum_{j=1}^6 (DV^{d_1})_j \iint_{sp} \psi_j \, ds - \Delta t \, g (1-\theta) \Pi \sum_{i=1}^m \sum_{j=1}^6 \eta_j^n (n_y)_i \int_{\partial s_i} \psi_j \, dl \\
&+\Delta t \sum_1^m \sum_{j=1}^6 (\Delta \tau_y)_j \iint_{sp} \psi_j \, ds - 2 \cdot \Delta t \sum_1^m \sum_{j=1}^6 \left(\omega_z \cdot DU_j^{d_1} - \omega_x \cdot DW_j^{d_1}\right) \iint_S \psi_j \, ds
\end{aligned} \tag{7.56}$$

If the velocity and/or free surface gradient is too steep at a location, the second order interpolation element may cause numerical oscillation. In this case the first order interpolation sub-elements are used locally for the FV scheme (see equations 6.49 to 6.55). The risk of oscillation is checked by comparing the sign of the the gradients of the solution quantity at the nodes of the quadratic interpolation.

Writing equations (7.54) to (7.56) for the FV cells and handling the boundary conditions results in a linear system of equations of the form $[A] \cdot [X] = [B]$. The coefficient matrix $[A]$ is not constant in time, because it involves the time-variable quantities D and Π. Therefore an efficient iterative matrix solver is needed in order to solve the matrix equation at each time step. The coefficient matrix $[A]$ is not necessarily symmetric, because the water depth terms (Π and D in equations 7.54 to 7.56) are not equivalent at the neighbouring nodes. Thus, the iterative solver must not be restricted to symmetric matrices. The preconditioned bi-conjugate gradient stabilised method (BI-CGSTAB) has been used in this regard (see van der Vorst, 1992).

The open boundary nodes with known water level are handled by using Dirichlet condition for the water level and applying equations (7.55) and (7.56) for the depth integrated velocity components. The open boundary nodes with known velocity are handled by using Dirichlet condition for the velocity components and applying equation (7.54) for the water level. The closed boundary nodes are handled by using the three FV relationships (7.54) to (7.56) while the no-flow condition of the boundary is applied when the line integrals on the cell boundary are determined in equation (7.54). All of the boundaries are handled by using the normal interpolation elements (not the Neumann elements and the virtual nodes).

The procedure for calculating the water level, as described in this section, determines the depth integrated velocity components in addition to the water level. These extra parameters are not trivial. They will be used in the next step for calculating the effect of vertical diffusion and free surface gradient on the 3D velocity field.

7.6 Vertical Diffusion Step

The vertical diffusion and the effect of the barotropic pressure gradient (i.e. the free surface slope) on the 3D velocity field are determined in the vertical diffusion step. The solution procedure is implicit. Three one-dimensional equations for u, v, and w are solved along the vertical water columns. Each water column is solved independently. The relevant equations are (7.8) to (7.10). They are represented herein for easy referencing.

$$\frac{u^{d_2}-u^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u^{d_2}}{\partial z}\right)-g\frac{\rho_\eta}{\rho}\left(\theta\frac{\partial\eta^{n+1}}{\partial x}+(1-\theta)\frac{\partial\eta^{n}}{\partial x}\right)+2\cdot\omega_z\cdot v^{d_1}-2\cdot\omega_y\cdot w^{d_1} \tag{7.57}$$

$$\frac{v^{d_2}-v^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial v^{d_2}}{\partial z}\right)-g\frac{\rho_\eta}{\rho}\left(\theta\frac{\partial\eta^{n+1}}{\partial y}+(1-\theta)\frac{\partial\eta^{n}}{\partial y}\right)-2\cdot\omega_z\cdot u^{d_1}+2\cdot\omega_x\cdot w^{d_1} \tag{7.58}$$

$$\frac{w^{d_2}-w^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial w^{d_2}}{\partial z}\right)+2\cdot\omega_y\cdot u^{d_1}-2\cdot\omega_x\cdot v^{d_1} \tag{7.59}$$

The free surface slope has a dominant effect on the velocity field in many practical problems. The effect of this parameter on the depth integrated velocity has been determined in the previous step. However, its effect on the 3D velocity field must be calculated now. It is important to determine the free surface gradients exactly as they were calculated in the previous step. Otherwise the 3D velocity field will not be compatible with the water free surface. Recalling that the depth integrated form of equations (7.57) and (7.58) have been used in the previous step for the free surface calculation, the free surface gradients here may be replaced by the results of that step. In this way, the surface gradients are computed only once. Equations (7.41) and (7.42) may be rearranged as follows:

$$\begin{aligned}&-g\left(\theta\frac{\partial\eta^{n+1}}{\partial x}+(1-\theta)\frac{\partial\eta^{n}}{\partial x}\right)=\frac{D}{\Pi}\cdot\frac{U^{d_2}-U^{d_1}}{\Delta t}-\frac{1}{\Pi}\left(\frac{(\tau_x)_\eta}{\rho_\eta}-\frac{(\tau_x)_b}{\rho_b}\right)\\&-\frac{2}{\Pi}(\omega_z\cdot DV^{d_1}-\omega_y\cdot DW^{d_1})\end{aligned}\tag{7.60}$$

$$\begin{aligned}&-g\left(\theta\frac{\partial\eta^{n+1}}{\partial y}+(1-\theta)\frac{\partial\eta^{n}}{\partial y}\right)=\frac{D}{\Pi}\cdot\frac{V^{d_2}-V^{d_1}}{\Delta t}-\frac{1}{\Pi}\left(\frac{(\tau_y)_\eta}{\rho_\eta}-\frac{(\tau_y)_b}{\rho_b}\right)\\&+\frac{2}{\Pi}(\omega_z\cdot DU^{d_1}-\omega_x\cdot DW^{d_1})\end{aligned}\tag{7.61}$$

Inserting equations (7.60) and (7.61) into equations (7.57) and (7.58), we obtain:

$$\begin{aligned}&\frac{u^{d_2}-u^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u^{d_2}}{\partial z}\right)+\frac{\rho_\eta}{\rho}\frac{D}{\Pi}\frac{U^{d_2}-U^{d_1}}{\Delta t}-\frac{\rho_\eta}{\rho}\frac{1}{\Pi}\left(\frac{(\tau_x)_\eta}{\rho_\eta}-\frac{(\tau_x)_b}{\rho_b}\right)\\&+2\omega_z(v^{d_1}-\frac{\rho_\eta}{\rho}\frac{DV^{d_1}}{\Pi})-2\omega_y(w^{d_1}-\frac{\rho_\eta}{\rho}\frac{DW^{d_1}}{\Pi})\end{aligned}\tag{7.62}$$

$$\begin{aligned}&\frac{v^{d_2}-v^{d_1}}{\Delta t}=\frac{\partial}{\partial z}\left(\nu_v\frac{\partial v^{d_2}}{\partial z}\right)+\frac{\rho_\eta}{\rho}\frac{D}{\Pi}\frac{V^{d_2}-V^{d_1}}{\Delta t}-\frac{\rho_\eta}{\rho}\frac{1}{\Pi}\left(\frac{(\tau_y)_\eta}{\rho_\eta}-\frac{(\tau_y)_b}{\rho_b}\right)\\&-2\omega_z(u^{d_1}-\frac{\rho_\eta}{\rho}\frac{DU^{d_1}}{\Pi})+2\omega_x(w^{d_1}-\frac{\rho_\eta}{\rho}\frac{DW^{d_1}}{\Pi})\end{aligned}\tag{7.63}$$

Equations (7.62), (7.63) and (7.59) are solved on each computational grid column in order to determine the 3D velocity field for this step. The solution procedures for the three equations are the same, while the boundary conditions are different. The procedure for u-component is described below.

Equation (7.62) is rearranged as follows:

$$u^{d_2}-\Delta t\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u^{d_2}}{\partial z}\right)=u^{d_1}+\Delta t\frac{\rho_\eta}{\rho}\frac{D}{\Pi}\frac{U^{d_2}-U^{d_1}}{\Delta t}-\Delta t\frac{\rho_\eta}{\rho}\frac{1}{\Pi}\left(\frac{(\tau_x)_\eta}{\rho_\eta}-\frac{(\tau_x)_b}{\rho_b}\right)$$
$$+2\Delta t\,\omega_z(v^{d_1}-\frac{\rho_\eta}{\rho}\frac{DV^{d_1}}{\Pi})-2\Delta t\,\omega_y(w^{d_1}-\frac{\rho_\eta}{\rho}\frac{DW^{d_1}}{\Pi}) \qquad (7.64)$$

where everything in the right hand side is known.

The 1D finite volume algorithm of section (6.11) has been used in order to discretise equation (7.64). Cartesian coordinate system is used herein instead of the sigma coordinate system. In this way, not only the velocity but also its vertical gradient remains continuous at the element boundaries. It is a benefit for the vertical diffusion term.

The integral of equation (7.64) along a 1D FV cell is written as follows:

$$\int_{z1}^{z2}u^{d_2}dz-\Delta t\int_{z1}^{z2}\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u^{d_2}}{\partial z}\right)dz=\int_{z1}^{z2}u^{d_1}dz+\Delta t\,A_{col}\int_{z1}^{z2}\frac{\rho_\eta}{\rho}dz$$
$$+2\Delta t\int_{z1}^{z2}(\omega_z v^{d_1}-\omega_y w^{d_1})dz \qquad (7.65)$$

where l is the node number in the vertical direction,

$$A_{col}=\frac{1}{\Pi}\frac{DU^{d_2}-DU^{d_1}}{\Delta t}-\frac{1}{\Pi}\left(\frac{(\tau_x)_\eta}{\rho}-\frac{(\tau_x)_b}{\rho}\right)-2\left(\omega_z\frac{DV^{d_1}}{\Pi}-\omega_y\frac{DW^{d_1}}{\Pi}\right) \qquad (7.66)$$

where A_{col} is constant along the column and is known from the previous step for the free surface.

$$z1=\frac{z_{l-1}+z_l}{2}\ ,\ z2=\frac{z_l+z_{l+1}}{2}\qquad for\ \ l\neq 1,N$$

$z1=z_l\ ,\ z2=\frac{z_l+z_{l+1}}{2}\qquad for\ \ l=1$ (the bed layer's node)

$z1=\frac{z_l+z_{l-1}}{2}\ ,\ z2=z_l\qquad for\ \ l=N$ (the surface layer's node)

The second term in the left hand side of equation (7.65) is simplified as follows:

$$\int_{z1}^{z2}\frac{\partial}{\partial z}\left(\nu_v\frac{\partial u^{d_2}}{\partial z}\right)dz=\left[\nu_v\frac{\partial u^{d_2}}{\partial z}\right]_{z1}^{z2}=\left(\nu_v\frac{\partial u^{d_2}}{\partial z}\right)_{z2}-\left(\nu_v\frac{\partial u^{d_2}}{\partial z}\right)_{z1} \qquad (7.67)$$

The point $z1$ is located in the interpolation element that is bounded by the nodes $l-1$ and l while the point $z2$ is located in the interpolation element that is bounded by the nodes l and $l+1$. The right hand side terms in equation (7.67) is derived through centred FD between the layers. The result is:

$$\left(\nu_v \frac{\partial u^{d_2}}{\partial z}\right)_{z1} = \left(\nu_v\right)_{z1} \frac{u_l - u_{l-1}}{\Delta z} \tag{7.68}$$

$$\left(\nu_v \frac{\partial u^{d_2}}{\partial z}\right)_{z2} = \left(\nu_v\right)_{z2} \frac{u_{l+1} - u_l}{\Delta z} \tag{7.69}$$

Following the approach described in section (6.11) and combining equations (7.67) to (7.69), equation (7.65) are discretised as follows:

$$\begin{aligned}
&\sum_{k=l-2}^{l+1} u_k^{d_2} \int_{z1}^{z_l} \psi_k^v(z)\,dz + \sum_{k=l-1}^{l+2} u_k^{d_2} \int_{z_l}^{z2} \psi_k^v(z)\,dz \\
&-\Delta t \left(\left(\nu_v\right)_{z2} \frac{u_{l+1}^{d_2} - u_l^{d_2}}{\Delta z} - \left(\nu_v\right)_{z1} \frac{u_l^{d_2} - u_{l-1}^{d_2}}{\Delta z} \right) \\
&= \sum_{k=l-2}^{l+1} u_k^{d_1} \int_{z1}^{z_l} \psi_k^v(z)\,dz + \sum_{k=l-1}^{l+2} u_k^{d_1} \int_{z_l}^{z2} \psi_k^v(z)\,dz \\
&+\Delta t\, A_{col} \left(\sum_{k=l-2}^{l+1} \left(\frac{\rho_\eta}{\rho}\right)_k \int_{z1}^{z_l} \psi_k^v(z)\,dz + \sum_{k=l-1}^{l+2} \left(\frac{\rho_\eta}{\rho}\right)_k \int_{z_l}^{z2} \psi_k^v(z)\,dz \right) \\
&+2\Delta t \left(\sum_{k=l-2}^{l+1} (\omega_z v_k^{d_1} - \omega_y w_k^{d_1}) \int_{z1}^{z_l} \psi_k^v\,dz + \sum_{k=l-1}^{l+2} (\omega_z v_k^{d_1} - \omega_y w_k^{d_1}) \int_{z_l}^{z2} \psi_k^v\,dz \right)
\end{aligned} \tag{7.70}$$

The discretised equation for v is derived in a similar way. The resulting equation is:

$$\begin{aligned}
&\sum_{k=l-2}^{l+1} v_k^{d_2} \int_{z1}^{z_l} \psi_k^v(z)\,dz + \sum_{k=l-1}^{l+2} v_k^{d_2} \int_{z_l}^{z2} \psi_k^v(z)\,dz \\
&-\Delta t \left(\left(\nu_v\right)_{z2} \frac{v_{l+1}^{d_2} - v_l^{d_2}}{\Delta z} - \left(\nu_v\right)_{z1} \frac{v_l^{d_2} - v_{l-1}^{d_2}}{\Delta z} \right) \\
&= \sum_{k=l-2}^{l+1} v_k^{d_1} \int_{z1}^{z_l} \psi_k^v(z)\,dz + \sum_{k=l-1}^{l+2} v_k^{d_1} \int_{z_l}^{z2} \psi_k^v(z)\,dz \\
&+\Delta t\, B_{col} \left(\sum_{k=l-2}^{l+1} \left(\frac{\rho_\eta}{\rho}\right)_k \int_{z1}^{z_l} \psi_k^v(z)\,dz + \sum_{k=l-1}^{l+2} \left(\frac{\rho_\eta}{\rho}\right)_k \int_{z_l}^{z2} \psi_k^v(z)\,dz \right) \\
&-2\Delta t \left(\sum_{k=l-2}^{l+1} (\omega_z u_k^{d_1} - \omega_x w_k^{d_1}) \int_{z1}^{z_l} \psi_k^v\,dz + \sum_{k=l-1}^{l+2} (\omega_z u_k^{d_1} - \omega_x w_k^{d_1}) \int_{z_l}^{z2} \psi_k^v\,dz \right)
\end{aligned} \tag{7.71}$$

where

$$B_{col} = \frac{1}{\Pi} \frac{DV^{d_2} - DV^{d_1}}{\Delta t} - \frac{1}{\Pi} \left(\frac{(\tau_y)_\eta}{\rho} - \frac{(\tau_y)_b}{\rho} \right) + 2\left(\omega_z \frac{DU^{d_1}}{\Pi} - \omega_x \frac{DW^{d_1}}{\Pi} \right) \tag{7.72}$$

where B_{col} is constant along the column and is known from the previous step for the free surface.

The discretised equation for w is derived from equation (7.59) as follows:

$$\begin{aligned}
&\sum_{k=l-2}^{l+1} w_k^{d_2}\int_{z1}^{z_l}\psi_k^v(z)\,dz+\sum_{k=l-1}^{l+2} w_k^{d_2}\int_{z_l}^{z2}\psi_k^v(z)\,dz\\
&-\Delta t\left(\left(\nu_v\right)_{z2}\frac{w_{l+1}^{d_2}-w_l^{d_2}}{\Delta z}-\left(\nu_v\right)_{z1}\frac{w_l^{d_2}-w_{l-1}^{d_2}}{\Delta z}\right)\\
&=\sum_{k=l-2}^{l+1} w_k^{d_1}\int_{z1}^{z_l}\psi_k^v(z)\,dz+\sum_{k=l-1}^{l+2} w_k^{d_1}\int_{z_l}^{z2}\psi_k^v(z)\,dz\\
&+2\Delta t\left(\sum_{k=l-2}^{l+1}\left(\omega_y u_k^{d_1}-\omega_x v_k^{d_1}\right)\int_{z1}^{z_l}\psi_k^v\,dz+\sum_{k=l-1}^{l+2}\left(\omega_y u_k^{d_1}-\omega_x v_k^{d_1}\right)\int_{z_l}^{z2}\psi_k^v\,dz\right)
\end{aligned}\qquad(7.73)$$

The derivations of the shape function integrals in equations (7.70), (7.71) and (7.73) have been presented in section (6.11).

Equations (7.70), (7.71) and (7.73) are solved along the vertical computational grid columns. This procedure, in addition to the boundary conditions, results in an individual linear system of equations for each velocity component at each column. Each system of equations is of the form ($[\![A]\!][x]=[b]$) being five-diagonal coefficient matrix. The number of unknowns is equal to the number of the layers in a column plus two. The two extra unknowns are due to the virtual nodes discussed in section (6.5). They are included in order to handle the boundary conditions at the bed and free surface. A standard five-diagonal matrix solver is used in order to solve each nodal column separately. The numerical solution procedures for solve equations (7.9) and (7.10), to get v and w, are similar to that described for equation (7.8).

The dynamic boundary conditions at the free surface and the bed are used for solving equations (7.8) and (7.9). The kinematic boundary conditions are used for equation (7.10). The Neumann boundary conditions for the u- and v-equations at the water surface are extracted from equations (5.8) and (5.9). The terms $\partial w_\eta/\partial x$ and $\partial w_\eta/\partial y$ are determined explicitly in this regard. The corresponding Neumann boundary conditions at the bed are extracted from equations (5.15) and (5.16) if the bed slope is negligible. These equations link the vertical gradient of the velocity component to its own value at the boundary via a linear relationship with constant coefficients. The numerical procedure used to handle this kind of boundary condition in the vertical direction is described in section (6.5).

The Dirichlet boundary condition for the w -equation at the bed is given as:

$$w_b = u_b \frac{\partial z_b}{\partial x} + v_b \frac{\partial z_b}{\partial y} \tag{7.74}$$

The Dirichlet boundary condition for the w -equation at the free surface is given as:

$$w_\eta = \frac{\partial \eta}{\partial t} + u_\eta \frac{\partial \eta}{\partial x} + v_\eta \frac{\partial \eta}{\partial y} \tag{7.75}$$

The virtual nodes are not used for the w -equation (7.10).

At the lateral free-slip closed boundaries, the vertical diffusion sub-model is solved for the parallel direction.

7.7 Hydrodynamic Pressure Correction Step

The pressure correction equation cannot be transformed into a simple form in the sigma coordinate system due to the nonconformity of this transformation (Jonkowsky, 1999). In order to avoid the difficulties caused by transforming the second order derivative directly, the transformation is conducted by applying the divergence theorem on the Poisson equation in the real mesh. The Poisson equation in the Cartesian coordinate system is:

$$\frac{\partial}{\partial x}\left(\frac{\partial \hat{q}}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\partial \hat{q}}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{\partial \hat{q}}{\partial z}\right) = \frac{\rho}{\Delta t}\left(\frac{\partial u_{d_2}}{\partial x} + \frac{\partial v_{d_2}}{\partial y} + \frac{\partial w_{d_2}}{\partial z}\right) \tag{7.76}$$

where u_{d_2} , v_{d_2} , and w_{d_2} are the intermediate velocity components as computed by the vertical diffusion step. Equation (7.76) is written in the following form:

$$\nabla \cdot \left(\frac{\partial \hat{q}}{\partial x}, \frac{\partial \hat{q}}{\partial y}, \frac{\partial \hat{q}}{\partial z}\right) = \frac{\rho}{\Delta t} \nabla \cdot \left(u_{d_2}, v_{d_2}, w_{d_2}\right) \tag{7.77}$$

Integrating equation (7.77) over the three dimensional finite volume cell results in the following relationship:

$$\iiint_\Omega \nabla \cdot \left(\frac{\partial \hat{q}}{\partial x}, \frac{\partial \hat{q}}{\partial y}, \frac{\partial \hat{q}}{\partial z}\right) d\,\Omega = \frac{\rho}{\Delta t} \iiint_\Omega \nabla \cdot \left(u_{d_2}, v_{d_2}, w_{d_2}\right) d\,\Omega \tag{7.78}$$

where Ω is the volume of the 3D FV cell.

Applying the divergence theorem on equation (7.78) results in the following relationship:

$$\iint_{\partial\Omega} \left(n_x \frac{\partial \hat{q}}{\partial x} + n_y \frac{\partial \hat{q}}{\partial y} + n_z \frac{\partial \hat{q}}{\partial z}\right) d\,s = \frac{\rho}{\Delta t} \iint_{\partial\Omega} \left(n_x u_{d_2} + n_y v_{d_2} + n_z w_{d_2}\right) d\,s \tag{7.79}$$

where $\partial\Omega$ is the boundary surface of the 3D FV cell.

Equation (7.79) is not transformed into the sigma coordinate system. It is solved in the real space. Considering that the computational nodes are located on non-horizontal layers, the transformation rules of the derivatives may be used to transform $\partial / \partial x$ into $\partial / \partial \tilde{x}$ and $\partial / \partial y$ into $\partial / \partial \tilde{y}$. The transformation rules are (Thompson et al., 1985):

$$\frac{\partial \hat{q}}{\partial x} = \frac{\partial \hat{q}}{\partial \tilde{x}} - \frac{\partial z}{\partial \tilde{x}} \cdot \frac{\partial \hat{q}}{\partial z} , \quad \frac{\partial \hat{q}}{\partial y} = \frac{\partial \hat{q}}{\partial \tilde{y}} - \frac{\partial z}{\partial \tilde{y}} \cdot \frac{\partial \hat{q}}{\partial z} \tag{7.80}$$

Therefore equation (7.79) may be rewritten as:

$$\begin{gathered} \iint_{\partial \Omega} \left(n_x \frac{\partial \hat{q}}{\partial \tilde{x}} - n_x \frac{\partial z}{\partial \tilde{x}} \cdot \frac{\partial \hat{q}}{\partial z} + n_y \frac{\partial \hat{q}}{\partial \tilde{y}} - n_y \frac{\partial z}{\partial \tilde{y}} \cdot \frac{\partial \hat{q}}{\partial z} + n_z \frac{\partial \hat{q}}{\partial z} \right) d s = \\ \frac{\rho}{\Delta t} \iint_{\partial \Omega} \left(n_x u_{d_2} + n_y v_{d_2} + n_z w_{d_2} \right) d s \end{gathered} \tag{7.81}$$

This can be further rearranged as follows:

$$\iint_{\partial \Omega} \left(n_x \frac{\partial \hat{q}}{\partial \tilde{x}} + n_y \frac{\partial \hat{q}}{\partial \tilde{y}} + \frac{\partial \hat{q}}{\partial z} \left(n_z - N_z \right) \right) d s = \frac{\rho}{\Delta t} \iint_{\partial \Omega} \left(n_x u_{d_2} + n_y v_{d_2} + n_z w_{d_2} \right) d s \tag{7.82}$$

where:

$$N_z = n_x \frac{\partial z}{\partial \tilde{x}} + n_y \frac{\partial z}{\partial \tilde{y}} \tag{7.83}$$

The boundary surface of a 3D FV cell is partitioned into three surfaces:

$$\partial \Omega = S1 + S2 + S3 \tag{7.84}$$

where $S1$ is the lower face of a 3D FV cell, $S2$ is the upper face, and $S3$ is the lateral boundary surface.

In the same way, the boundary surface integrals in equation (7.82) are partitioned into three parts: integrals on $S1$, $S2$, and $S3$. The boundary faces $S1$ and $S2$ are located between the sigma layers. The integration over $S1$ and $S2$ is done on the 2D FV cells that are located on the computational grid layers. Then the surface integrals on $S1$ and $S2$ are determined by averaging the 2D FV cell integrals of the layers.

The surface integrals on the 2D FV cells are calculated according to equations (6.49) to (6.55) in section (6.7). For example:

$$\iint_S \left(n_z \frac{\partial \hat{q}}{\partial z} \right) d s = n_z \cdot \iint_S \frac{\partial \hat{q}}{\partial z} \, d s = n_z \cdot \sum_1^m \sum_{j=1}^3 \left(\frac{\partial \hat{q}}{\partial z} \right)_j \left(\iint_{sp} \psi_j^h dx \, dy \right) \tag{7.85}$$

The discretised form of equation (7.82) on the faces S1 and S2 may be written as follows:

$$\begin{aligned}&\sum_{1}^{m}\sum_{j=1}^{3}\left(n_x(\iint_{sp}\frac{\partial\psi_j^h}{\partial x}dx\,dy)+n_y(\iint_{sp}\frac{\partial\psi_j^h}{\partial y}dx\,dy)\right)\hat{q}_j\\&+\sum_{1}^{m}\sum_{j=1}^{3}(n_z-N_z)\cdot(\iint_{sp}\psi_j^h dx\,dy)\left(\frac{\partial\hat{q}}{\partial z}\right)_j\\&=\frac{\rho}{\Delta t}\cdot\\&\sum_{1}^{m}\sum_{j=1}^{3}\left(n_x\cdot\left(u_{d_2}\right)_j+n_y\cdot\left(v_{d_2}\right)_j+n_z\cdot\left(w_{d_2}\right)_j\right)(\iint_{sp}\psi_j^h dx\,dy)\end{aligned}\tag{7.86}$$

If the computational grid layers are horizontal, then $n_x=n_y=0$ on the boundary faces $S1$ and $S2$. Therefore:

$$\iint_{S1}\left(n_x\frac{\partial\hat{q}}{\partial\tilde{x}}\right)d\,s=\iint_{S1}\left(n_y\frac{\partial\hat{q}}{\partial\tilde{y}}\right)d\,s=\iint_{S2}\left(n_x\frac{\partial\hat{q}}{\partial\tilde{x}}\right)d\,s=\iint_{S2}\left(n_y\frac{\partial\hat{q}}{\partial\tilde{y}}\right)d\,s=0\tag{7.87}$$

The surface integrals on the lateral face of the 3D FV cell $S3$ are computed through the following steps:

a) to calculate the 1D integrals on the boundary line of the lower and upper faces $S1$ and $S2$.

b) to average the 1D integrals of step (a) and multiply it by the height of the 3D cell.

The corresponding equation is:

$$\iint_{S3}L_{HS}\,d\,s=h_{cell}\cdot\frac{1}{2}\cdot\left(\int_{\partial S1}L_{HS}\,d\,l+\int_{\partial S2}L_{HS}\,d\,l\right)\tag{7.88}$$

where h_{cell} is the height of the 3D FV cell and L_{HS} is the left hand side of equation (7.82).

In fact, one dimensional boundary integration is determined on the boundaries of the 2D FV cells that are located on the computational grid layers. The 1D boundary integrals for $\partial S1$ and $\partial S2$ are determined by averaging the calculated 1D boundary integrals on the layers.

The value n_z is always zero on $S3$. Thus we have:

$$\iint_{S3}\left(n_z\frac{\partial\hat{q}}{\partial z}\right)d\,s=n_z\cdot\iint_{S3}\left(\frac{\partial\hat{q}}{\partial z}\right)d\,s=0\tag{7.89}$$

Therefore it is sufficient to calculate the 1D boundary integrals of the remaining parts of equation (7.82) on the 2D FV cells for layers. It is determined based on equations (6.58) to (6.61) given in section (6.8), i.e.s:

$$\begin{aligned}&\int_{\partial S}\left(n_x\frac{\partial \hat{q}}{\partial \tilde{x}}+n_y\frac{\partial \hat{q}}{\partial \tilde{y}}-N_z\frac{\partial \hat{q}}{\partial z}\right)d\,s=\\&\sum_{i=1}^{m}\sum_{j=1}^{3}\left(n_x\int_{\partial s_i}\frac{\partial \psi_j^h}{\partial x}\,ds+n_y\int_{\partial s_i}\frac{\partial \psi_j^h}{\partial y}\,ds\right)\hat{q}_j-\sum_{i=1}^{m}\sum_{j=1}^{3}N_z(\int_{\partial s_i}\psi_j^h\,ds)\left(\frac{\partial \hat{q}}{\partial z}\right)_j=\\&\frac{\rho}{\Delta t}\cdot\sum_{1}^{m}\sum_{j=1}^{3}\left(n_x\cdot\left(u_{d_2}\right)_j+n_y\cdot\left(v_{d_2}\right)_j\right)(\int_{\partial s_i}\psi_j^h dx\,dy)\end{aligned}\qquad(7.90)$$

where ∂S is the boundary of the 2D FV cells on the computational grid layers. The $\tilde{x}$ and $\tilde{y}$ gradients in equation (7.90) are discretised according to equations (6.45) and (6.46) and (6.52) and (6.53) in section (6.7). The vertical gradient is discretised using the finite difference scheme. The boundary integrals are discretised through the procedure described in section (6.8).

The discretisation procedure for the right hand side of equation (7.82) is similar to that for the left hand side using quadratic elements. If the velocity gradient is too steep at a location, the second order interpolation method may cause numerical oscillations. In this case the first order sub-elements are used locally for the right hand side terms. The risk of oscillation is checked by comparing the sign of the gradients of the solution quantity at the nodes.

The discretised form of equation (7.82), as described above, is written for all of the 3D FV cells in all of the sigma layers except at Dirichlet boundaries. This procedure, in addition to the boundary conditions, results in a linear system of equations for the hydrodynamic pressure correction field. The linear system is solved by an iterative matrix solver. The bi-conjugate gradient stabilised (BI-CGSTAB) method (van der Vorst, 1992) has been used in this regard.

The boundary condition for the free surface layer and the lateral open boundaries is $\hat{q}=0$. The boundary condition for the bed and the closed lateral boundaries is $(\partial \hat{q}/\partial n)=0$ where n is perpendicular to the boundary. This condition is used because the impermeability condition at the closed boundaries has been satisfied in the intermediate velocity field $(u^{d_2}, v^{d_2}, w^{d_2})$.

The procedure for handling the Neumann boundary condition at the lateral boundaries has been discussed in section (6.9). The virtual nodes are used herein. The procedure for the boundary condition at the bed is more complex, because the bed is not necessarily horizontal. The relationship used at the bottom boundary is:

$$\frac{\partial \hat{q}_b}{\partial n}=n_x\frac{\partial \hat{q}}{\partial x}+n_y\frac{\partial \hat{q}}{\partial y}+n_z\frac{\partial \hat{q}}{\partial z}=0\qquad(7.91)$$

The bed boundary condition is handled through the virtual nodes under the bed as described in section (6.5).

7.8 Velocity Projection Step

The intermediate velocity field $(u^{d_2}, v^{d_2}, w^{d_2})$ generally does not satisfy the continuity equation. In other words, it is not convergence-free. The reason is that the hydrodynamic pressure field used in calculating the velocity is from the previous time step. It is not necessarily consistent with the hydrodynamic pressure at the new time step. However, the hydrodynamic pressure correction for the new time step has been determined from the previous step. Thus it is right time for the intermediate velocity field to be projected into a convergence-free form. Equations (7.11) to (7.13) may be used for this purpose. They must be transformed into the sigma coordinate system before discretisation. The transformed forms are:

$$\frac{\partial \hat{q}}{\partial x}=\frac{\partial \hat{q}}{\partial \tilde{x}}-\frac{\partial z}{\partial \tilde{x}}\cdot\frac{\partial \hat{q}}{\partial z}, \quad \frac{\partial \hat{q}}{\partial y}=\frac{\partial \hat{q}}{\partial \tilde{y}}-\frac{\partial z}{\partial \tilde{y}}\cdot\frac{\partial \hat{q}}{\partial z}$$

$$u^{n+1}=u^{d_2}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial x}=u^{d_2}-\Delta t\frac{1}{\rho}\left(\frac{\partial \hat{q}}{\partial \tilde{x}}-\frac{\partial z}{\partial \tilde{x}}\frac{\partial \hat{q}}{\partial z}\right) \tag{7.92}$$

$$v^{n+1}=v^{d_2}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial y}=v^{d_2}-\Delta t\frac{1}{\rho}\left(\frac{\partial \hat{q}}{\partial \tilde{y}}-\frac{\partial z}{\partial \tilde{y}}\frac{\partial \hat{q}}{\partial z}\right) \tag{7.93}$$

$$w^{n+1}=w^{d_2}-\Delta t\frac{1}{\rho}\frac{\partial \hat{q}}{\partial z} \tag{7.94}$$

Equations (7.92) to (7.94) have been solved for the computational nodes except for the Dirichlet boundary ones where the velocity is known. Equations (7.92) and (7.93) are discretised and solved on the sigma planes through the 2D finite volume algorithm described in sections (6.7). Linear elements are used. In this way, the discretisation scheme is consistent with the one used in the hydrodynamic pressure correction step. The vertical gradient is determined through the finite difference scheme.

The discretised form of equations (7.92) to (7.94) are derived as follows:

$$\begin{aligned}\sum_{1}^{m}\sum_{j=1}^{6}u_j^{n+1}\iint_{sp}\psi_j^h ds&=\sum_{1}^{m}\sum_{j=1}^{6}u_j^{d_2}\iint_{sp}\psi_j^h ds\\ &-\Delta t\frac{1}{\rho}\sum_{1}^{m}\sum_{j=1}^{3}\hat{q}_j\iint_{sp}\frac{\partial \psi_j^h}{\partial \tilde{x}}ds+\Delta t\frac{1}{\rho}\sum_{1}^{m}\sum_{j=1}^{3}\frac{\partial z}{\partial \tilde{x}}\left(\frac{\partial \hat{q}}{\partial z}\right)_j\iint_{sp}\psi_j^h ds\end{aligned} \tag{7.95}$$

$$\begin{aligned}\sum_{1}^{m}\sum_{j=1}^{6}v_j^{n+1}\iint_{sp}\psi_j^h ds&=\sum_{1}^{m}\sum_{j=1}^{6}v_j^{d_2}\iint_{sp}\psi_j^h ds\\ &-\Delta t\frac{1}{\rho}\sum_{1}^{m}\sum_{j=1}^{3}\hat{q}_j\iint_{sp}\frac{\partial \psi_j^h}{\partial \tilde{y}}ds+\Delta t\frac{1}{\rho}\sum_{1}^{m}\sum_{j=1}^{3}\frac{\partial z}{\partial \tilde{y}}\left(\frac{\partial \hat{q}}{\partial z}\right)_j\iint_{sp}\psi_j^h ds\end{aligned} \tag{7.96}$$

$$\sum_{1}^{m}\sum_{j=1}^{6} w_j^{n+1}\iint_{sp}\psi_j^h ds=\sum_{1}^{m}\sum_{j=1}^{6} w_j^{d_2}\iint_{sp}\psi_j^h ds-\Delta t\frac{1}{\rho}\sum_{1}^{m}\sum_{j=1}^{3}\left(\frac{\partial\hat{q}}{\partial z}\right)_j\iint_{sp}\psi_j^h ds \tag{7.97}$$

Each one of the above equations is applied to the FV cells on a layer in order to produce a linear system of equations in combination with the boundary conditions. The coefficient matrix is fixed in time and unique for each layer. It is inverted at the start of simulation and is multiplied by the right hand side vector for each layer at each time step.

It would be possible to solve equations (7.92) to (7.94) explicitly without matrix solution. However, the above algorithm guaranties that the discretisation schemes here are identical with other parts of the model. Considering that the inverted coefficient matrix is used, this solution procedure is not slower than the explicit algorithm.

7.9 Water Level Refining Option

The water level resulted from the implicit free surface step has been computed based on the intermediate velocity field, which is not necessarily convergence-free. Nevertheless it is an acceptable implicit estimation to be used for the barotropic pressure gradient terms in order to increase the simulation stability. It is also acceptable for most of the practical applications, because the hydrodynamic pressure usually do not have noticeable effect on the water level. However, it is possible, if needed, to solve the depth integrated continuity equation again by the final velocity field of the new time step, in order to determine the water level more precisely. The resulted water level will be in complete harmony with the velocity field.

The semi discretised form of the depth integrated continuity equation is:

$$\begin{aligned}&-\frac{\partial\eta}{\partial t}=-\frac{\eta^{n+1}-\eta^n}{\Delta t}\\&=\theta\left(\frac{\partial}{\partial x}\left(\int_{z_b}^{\eta^*}u^{n+1}dz\right)+\frac{\partial}{\partial y}\left(\int_{z_b}^{\eta^*}v^{n+1}dz\right)\right)+(1-\theta)\left(\frac{\partial}{\partial x}\left(\int_{z_b}^{\eta^n}u^n dz\right)+\frac{\partial}{\partial y}\left(\int_{z_b}^{\eta^n}v^n dz\right)\right)\end{aligned} \tag{7.98}$$

where η^* is the implicit water level estimated in the “Free Surface Step”. Equation (7.98) may be solved iteratively until η^* converges to η^{n+1}, if necessary.

Equation (7.98) may be re-arranged as the following:

$$\eta^{n+1}=\eta^n-\Delta t\,\theta\left(\frac{\partial DU^{n+1}}{\partial x}+\frac{\partial DV^{n+1}}{\partial y}\right)-\Delta t(1-\theta)\left(\frac{\partial DU^n}{\partial x}+\frac{\partial DV^n}{\partial y}\right) \tag{7.99}$$

where:

$$DU=\int_{z_b}^{\eta} u\,dz \quad \text{and} \quad DV=\int_{z_b}^{\eta} v\,dz \tag{7.100}$$

All of the quantities in the right hand side of equation (7.99) are known. The vertical integrations are discretised through equations (6.17) and (6.18). Details in this regard have been discussed in section (6.5). Equation (7.99) may be solved for each computational node separately. However, the FV approach has been used in order to use an identical discretisation scheme as the other parts of the model. In order to apply the divergence theorem, equation (7.99) is rewritten to give:

$$\eta^{n+1}=\eta^{n}-\Delta t\,\theta\,\nabla\cdot\left(DU^{n+1},DV^{n+1}\right)-\Delta t(1-\theta)\nabla\cdot\left(DU^{n},DV^{n}\right) \tag{7.101}$$

Applying the FV approach to equation (7.101) results in the following relationship:

$$\begin{aligned}&\iint_S \eta^{n+1}\,ds=\iint_S \eta^{n}\,ds\\&-\Delta t\,\theta\iint_S \nabla\cdot\left(DU^{n+1},DV^{n+1}\right)ds-\Delta t(1-\theta)\iint_S \nabla\cdot\left(DU^{n},DV^{n}\right)ds\end{aligned} \tag{7.102}$$

Applying the divergence theorem, equation (7.102) may be written as follows:

$$\begin{aligned}&\iint_S \eta^{n+1}\,ds=\iint_S \eta^{n}\,ds\\&-\Delta t\,\theta\int_{\partial S}\left(n_x DU^{n+1}+n_y DV^{n+1}\right)ds-\Delta t(1-\theta)\int_{\partial S}\left(n_x DU^{n}+n_y DV^{n}\right)ds\end{aligned} \tag{7.103}$$

Equation (7.103) may be discretised using the algorithms described in sections (6.7) and (6.8). The result is:

$$\begin{aligned}&\sum_{1}^{m}\sum_{j=1}^{6}\eta_j^{n+1}\iint_{sp}\psi_j^h ds=\sum_{1}^{m}\sum_{j=1}^{6}\eta_j^{n}\iint_{sp}\psi_j^h ds\\&-\Delta t\,\theta\sum_{i=1}^{m}\sum_{j=1}^{6}\left(n_x DU_j^{n+1}+n_y DV_j^{n+1}\right)\int_{\partial s_i}\psi_j\,ds\\&-\Delta t(1-\theta)\sum_{i=1}^{m}\sum_{j=1}^{6}\left(n_x DU_j^{n}+n_y DV_j^{n}\right)\int_{\partial s_i}\psi_j\,ds\end{aligned} \tag{7.104}$$

The integrals in equation (7.104) are derived analytically through equations (6.37) to (6.43) and (6.56) to (6.61). Equation (7.104) is written for all of the FV cells, except for the Dirichlet boundary nodes. The result is a linear set of equations of the form ($[\![A]\!][x]=[b]$) in which the coefficient matrix is constant in time. The coefficient matrix is inverted once and is multiplied by the RHS vector at every time step.

7.10 3D Advection Diffusion Model

There are other 3D equations that are required to be solved in addition to the flow equations. They are the salinity equation (5.29), the temperature equation (5.33), the suspended sediment concentration equation (5.56), and the k-ε equations (5.69 and 5.70). Equations (5.98) to (6.40) represent the transformed form of these equations in the sigma coordinate system. They are specific versions of the general advection-diffusion equation, solved with a numerical algorithm similar to that used for the momentum equations.

The general algorithm contains the following steps:

a) advection step

b) horizontal diffusion step including the source terms

c) vertical diffusion step

The advection solver is exactly the same as that used for the flow model. Equation (7.15) and the interpolation relationship (6.35) have been used in this regard. The advecting velocity field in this regard is $(u\,,v\,,\tilde{w})$, except for the suspended sediment concentration where the settling velocity is also included.

The settling velocity is not necessarily constant in time, because it is a function of the suspended sediment concentration. The present study introduces a numerical algorithm which handles the settling velocity implicitly. Such a procedure is helpful in situations where the concentration changes rapidly, as is the case in the front of the turbidity current.

The implicit value of the settling velocity may be written as follows:

$$w_s = w_s^n + \theta(\delta w_s) \tag{7.105}$$

where $(\delta w_s) = w_s^{n+1} - w_s^n$ and θ is the implicitness factor.

The settling velocity term in equation (5.100) may be expanded by using equation (7.105). The result is as follows:

$$\begin{aligned} -\frac{\partial w_s c}{\partial z} &= -w_s^n \frac{\partial c}{\partial z} - \theta(\delta w_s)\frac{\partial c}{\partial z} - \frac{\partial w_s^n}{\partial z}c - \theta\frac{\partial(\delta w_s)}{\partial z}c \\ &= \underbrace{-\frac{w_s^n}{H}\frac{\partial c}{\partial \tilde{z}}}_{1} \underbrace{-\theta(\delta w_s)\frac{\partial c}{\partial z}}_{2} \underbrace{-\frac{\partial w_s^n}{\partial z}c}_{3} \underbrace{-\theta\frac{\partial(\delta w_s)}{\partial z}c}_{4} \end{aligned} \tag{7.106}$$

The term (1) in the right hand side of equation (7.106) is added to its identical term in the advection part of equation (5.100). In this way, the advecting velocity field for the semi-Lagrangian procedure will be $(u\,,v\,,\tilde{w} - w_s^n / H\,)$. The terms (2), (3) and (4) are handled in the vertical diffusion step. The solution algorithms for the horizontal and the vertical diffusion steps are similar to their counterparts in the flow model. The vertical diffusion step is solved for each nodal column separately. An iterative procedure may be used until convergence is achieved in (δw_s) at each node.

7.11 The 2D Flow Model

The present study aims at developing a smooth and efficient combination of 2D/3D modelling framework. The three dimensional model has been described in the previous sections. This section describes the numerical algorithm of the two dimensional model. The general algorithm employed in the 2D model in the present study is similar to the 3D model. The operator splitting method has been used with the semi-Lagrangian approach for the advection step and the FV approach for the diffusion step. It is not unusual to simulate a 2DH problem for relatively large time duration, where the conservativeness of the numerical method becomes crucial. A conservative semi-Lagrangian procedure has been used in this regard. The faster non-conservative algorithm of the 3D model is also available.

The 2D flow equations (5.103) to (5.105) are solved numerically in order to simulate the depth averaged velocity field. The general splitting approach is as follows:

$$\frac{\partial f}{\partial t}=\underbrace{\frac{f^{n+1}-f^d}{\Delta t}}_{\textit{barotropic step}}+\underbrace{\frac{f^d-f^a}{\Delta t}}_{\textit{diffusion step}}+\underbrace{\frac{f^a-f^n}{\Delta t}}_{\textit{advection step}} \tag{7.107}$$

The general algorithm is combined from the following steps:

a) the advection step

b) the diffusion step

c) the free surface, barotropic pressure and friction step

The equations to be solved in the advection step are:

$$\frac{U^a-U^n}{\Delta t}+U\frac{\partial U}{\partial x}+V\frac{\partial U}{\partial y}=0 \tag{7.108}$$

$$\frac{V^a-V_n}{\Delta t}+U\frac{\partial V}{\partial x}+V\frac{\partial V}{\partial y}=0 \tag{7.109}$$

The equations to be solved in the diffusion step are:

$$\frac{U^d-U^a}{\Delta t}=\omega_z V-\frac{1}{\rho}\frac{\partial p_a}{\partial x}-\frac{g D}{2\rho}\frac{\partial \rho}{\partial x}+\bar{\nu}_t\left(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}\right)+(u_s-U)\,dis_s \tag{7.110}$$

$$\frac{V^d-V^a}{\Delta t}=-\omega_z U-\frac{1}{\rho}\frac{\partial p_a}{\partial y}-\frac{g D}{2\rho}\frac{\partial \rho}{\partial y}+\bar{\nu}_t\left(\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2}\right)+(v_s-V)\,dis_s \tag{7.111}$$

The equations to be solved in the free surface, barotropic pressure and friction step are (see section 7.11.3):

$$\begin{aligned}&\eta^{n+1}+\Delta t\,\theta\left(\frac{\partial}{\partial x}(DU)^{n+1}+\frac{\partial}{\partial y}(DV)^{n+1}\right)=\\&\eta^{n}-\Delta t(1-\theta)\left(\frac{\partial}{\partial x}(DU)^{n}+\frac{\partial}{\partial y}(DV)^{n}\right)\end{aligned} \tag{7.112}$$

The equations used in the barotropic and friction step are:

$$\frac{DU^{n+1}-DU^d}{\Delta t}=-g\left(\theta D\frac{\partial \eta^{n+1}}{\partial x}+(1-\theta)D\frac{\partial \eta^{n}}{\partial x}\right)+\frac{\tau_{sx}}{\rho}-c_f U^{n+1}\sqrt{(U^n)^2+(V^n)^2} \tag{7.113}$$

$$\frac{DV^{n+1}-DV^d}{\Delta t}=-g\left(\theta D\frac{\partial \eta^{n+1}}{\partial y}+(1-\theta)D\frac{\partial \eta^{n}}{\partial y}\right)+\frac{\tau_{sy}}{\rho}-c_f V^{n+1}\sqrt{(U^n)^2+(V^n)^2} \tag{7.114}$$

7.11.1 Advection Step

The semi-Lagrangian characteristic method has been used in order to solve equations (7.108) and (7.109). Two algorithms have been used in this regard:

a) transforming the computational nodes along the characteristic lines

b) transforming the FV cells along the characteristic lines.

The above algorithms are described below.

a) Transform the computational nodes:

This approach is similar to the algorithm for the 3D model. It is fast but non-conservative. This algorithm is based on the assumption that the two velocity components and any other scalar quantity remain constant along the characteristic curves during the pure advection process, i.e.:

$$U_i^a=U_p^n\ ,\quad V_i^a=V_p^n \tag{7.115}$$

where the subscript i stands for the computational node and the subscript p is for the departure point (x_p , y_p) of the characteristic curve from the time $n\cdot\Delta t$ to the time $(n+1)\cdot\Delta t$.

The characteristic curve is identical to the current streamline when pure advection is considered. The 2D streamline is traced back from the point i during the time period Δt explicitly by dividing the time step into $K_{\Delta t}$ sub-steps $\delta t = \Delta t / K_{\Delta t}$:

The first stage:
$x_0 = x_i,\ \ U_o = U_i^n,\ \ y_0 = y_i,\ \ V_o = V_i^n$

The intermediate stages $(k = 1, \ldots, K_{\Delta t})$:

$$x_k = x_{k-1} - \delta t \cdot U_{k-1}^n \quad , U_{k-1} = U\left((x, y)_{k-1}\right)$$
$$y_k = y_{k-1} - \delta t \cdot V_{k-1}^n \quad , V_{k-1} = V\left((x, y)_{k-1}\right) \qquad (7.116)$$

The last stage:
$x_p = x_K,\ \ y_p = y_K$

The quadratic interpolation equation (6.1) with the shape functions (6.3) to (6.8) have been used in order to interpolate data from the computational nodes to the departure points at the previous time step. In this way, the nodal values are defined at the new time step. This algorithm is not conservative, however the interpolation procedure is sufficiently accurate for many practical problems. The clipping algorithm of (Wang et al., 1988) has been used in order to avoid oscillation risk and artificial local extremes in front of sharp gradients. The clipping algorithm is based on comparing the interpolated data with the maximum and minimum nodal values of the element. The interpolated value must be inside that range.

The boundaries of the solution domain are handled in a similar way as in the 3D advection step. If the backward streamline crosses a closed boundary at an intermediate time, it is followed further along the boundary in the backward direction of the velocity parallel to the boundary. If the boundary is open (input) with a known velocity, the concentration value at the departure point is computed through interpolation in time and space on the boundary. If there is any input boundary with a known water level instead of velocity, a procedure similar to that developed for the 3D model is then used.

b) Transform the FV cells:

This algorithm is a conservative semi-Lagrangian discretisation scheme. It is based on the fact that the cell-integral of the solution quantity over a 2D FV cell remains constant while the cell transforms and reshapes with pure advection. The FV cell, as described in section (6.7), has been used in this regard. This algorithm produces an individual relationship for the solute at each computational node. It is the cell boundary points, not the computational nodes that are traced backward along the streamlines. In this way, the position and geometry of the FV cell at the previous time step are determined. Then the cell-integral of the solution quantity in the transformed cell is computed through the divergence theorem by 1D integration along the cell boundaries. The interpolation shape functions and the nodal values at the previous time step are used for the analytical derivation of the boundary integrals. The boundary of the transformed FV cell is partitioned into several sub-cell edges, with each one in a single interpolation element. The integral may be defined analytically over each edge according to equations (6.56) to (6.61). The sub-cells here are not necessarily identical to the sub-cells in section (6.7). Because each sub-cell of the second type may sit in more than one interpolation element (see figure 7.2).

Recalling that the cell-integral of the solution quantity does not change during the pure advection transformation, the integrals over the original FV cells at the new time step are equivalent to the integral over the back-warded cell at the previous time step. On the other hand, the cell integrals at the new time step may also be calculated through the nodal values at the computational nodes (see section 6.7). The cell-integrals from the two methods are equivalent. This relationship links the nodal values at the new time step to the nodal values from the previous time step. The relationship is written for all of the FV cells. This procedure, in addition to the boundary conditions, results in a linear set of equations with a coefficient matrix that is fixed in time.

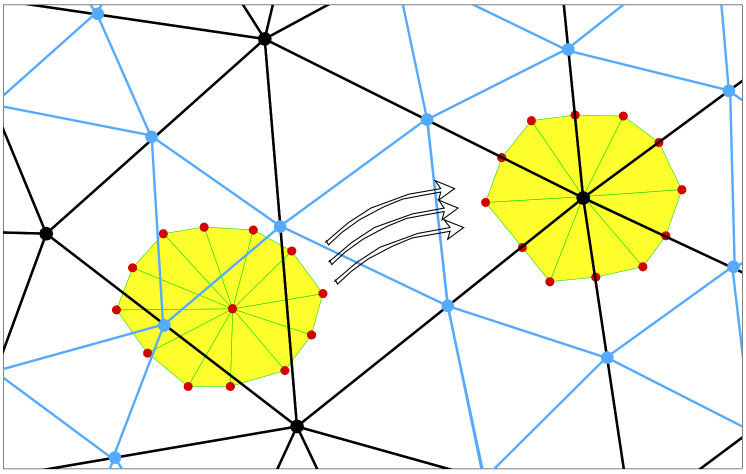


Figure 7.2. Semi-Lagrangian transformation of a FV-cell

7.11.2 Diffusion Step

The explicit FV method with FE shape functions has been used in order to solve equations (7.110) and (7.111). The procedure is similar to that used in the horizontal diffusion step for the

3D model with the same discretisation algorithm and the same boundary condition, and with the same matrix solver. For details see section (7.4).

7.11.3 Free Surface, Barotropic and Friction Step

Three equations are used in order to determine the free surface and final velocity implicitly. They are the depth integrated continuity equation, the remaining part of the momentum equation in the x direction, and the remaining part of the momentum equation in the y direction. Two approaches may be used for predicting the water level. One is to solve the three equations jointly in a linear system of equations with three unknowns and three equations for each computational node. The second approach is to derive a unique relationship with only the water level as the unknown by inserting the quantities DU and DV from the momentum equations (the remaining part) into the continuity equation. The first approach has been used in the present study. It is identical to the procedure used in the 3D model.

In the present step, the barotropic pressure (the free surface slope) and the new velocity distributions are determined implicitly. This method results in more stable solution and with less time step restriction. The bed shear stress has no significant effect on the flow field if the water depth is large. However it may be effective and dominant in the shallow waters regions. Using an implicit discretisation of the bed shear stress will increase the stability and robustness of the model in very shallow or wetting/drying regions. Thus the bed friction is solved implicitly in the current study.

The semi discretised equations to be solved are as follows:

$$\frac{\eta^{n+1}-\eta^{n}}{\Delta t}+\theta\left(\frac{\partial(DU)^{n+1}}{\partial x}+\frac{\partial(DV)^{n+1}}{\partial y}\right)+(1-\theta)\left(\frac{\partial(DU)^{n}}{\partial x}+\frac{\partial(DV)^{n}}{\partial y}\right)=DQ_{dis} \qquad (7.117)$$

$$\begin{aligned}\frac{DU^{n+1}-DU^{d}}{\Delta t}&=-g\left(\theta D\frac{\partial\eta^{n+1}}{\partial x}+(1-\theta)D\frac{\partial\eta^{n}}{\partial x}\right)\\&+\frac{\tau_{sx}}{\rho}-c_f U^{n+1}\sqrt{(U^n)^2+(V^n)^2}+(u_{dis}-U)DQ_{dis}\end{aligned} \qquad (7.118)$$

$$\begin{aligned}\frac{DV^{n+1}-DV^{d}}{\Delta t}&=-g\left(\theta D\frac{\partial\eta^{n+1}}{\partial y}+(1-\theta)D\frac{\partial\eta^{n}}{\partial y}\right)\\&+\frac{\tau_{sy}}{\rho}-c_f V^{n+1}\sqrt{(U^n)^2+(V^n)^2}+(v_{dis}-V)DQ_{dis}\end{aligned} \qquad (7.119)$$

Equations (7.117) to (7.119) may be rearranged in the following manner:

$$\eta^{n+1}+\Delta t\theta\left(\frac{\partial}{\partial x}(DU)^{n+1}+\frac{\partial}{\partial y}(DV)^{n+1}\right)= \\ \eta^{n}-\Delta t(1-\theta)\left(\frac{\partial}{\partial x}(DU)^{n}+\frac{\partial}{\partial y}(DV)^{n}\right)+D\,dis_s \tag{7.120}$$

$$DU^{n+1}+DU^{n+1}\Delta t\frac{c_f}{D}\sqrt{(U^n)^2+(V^n)^2}+\Delta t\,g\,\theta\,D\frac{\partial\eta^{n+1}}{\partial x}= \\ DU^d-\Delta t\,g\,D(1-\theta)\frac{\partial\eta^n}{\partial x}+\Delta t\frac{\tau_{sx}}{\rho}+\Delta t(u_{dis}-U)DQ_{dis} \tag{7.121}$$

$$DV^{n+1}+DV^{n+1}\Delta t\frac{c_f}{D}\sqrt{(U^n)^2+(V^n)^2}+\Delta t\,g\,\theta\,D\frac{\partial\eta^{n+1}}{\partial y}= \\ DV^d-\Delta t\,g\,D(1-\theta)\frac{\partial\eta^n}{\partial y}+\Delta t\frac{\tau_{sy}}{\rho}+\Delta t(v_{dis}-V)DQ_{dis} \tag{7.122}$$

Equation (7.120) may be rewritten in the following form:

$$\eta^{n+1}+\Delta t\theta\nabla\cdot\left(DU^{n+1},DV^{n+1}\right)=\eta^n-\Delta t(1-\theta)\nabla\cdot\left(DU^n,DV^n\right)+D\,dis_s \tag{7.123}$$

Following the finite volume procedure, equations (7.123), (7.121) and (7.122) may be integrated over the 2D FV cells in order to produce the following relationships:

$$\iint_S \eta^{n+1}ds+\Delta t\theta\iint_S \nabla\cdot\left(DU^{n+1},DV^{n+1}\right)ds= \\ \iint_S \eta^n ds-\Delta t(1-\theta)\iint_S \nabla\cdot\left(DU^n,DV^n\right)ds+\iint_S D\,dis_s\,ds \tag{7.124}$$

$$\iint_S DU^{n+1}ds+\Delta t\,c_f\iint_S DU^{n+1}\frac{\sqrt{(U^n)^2+(V^n)^2}}{D^n}ds \\ +\Delta t\,g\,\theta\,D^n\iint_S\frac{\partial\eta^{n+1}}{\partial x}ds=\iint_S DU^d ds-\Delta t\,g\,D^n(1-\theta)\iint_S\frac{\partial\eta^n}{\partial x}ds \\ +\Delta t\iint_S\frac{\tau_{sx}}{\rho}ds+\Delta t\iint_S(u_{dis}-U)DQ_{dis}ds \tag{7.125}$$

$$\iint_S DV^{n+1}ds+\Delta t\,c_f\iint_S DV^{n+1}\frac{\sqrt{(U^n)^2+(V^n)^2}}{D^n}ds \\ +\Delta t\,g\,\theta\,D^n\iint_S\frac{\partial\eta^{n+1}}{\partial y}ds=\iint_S DV^d ds-\Delta t\,g\,D^n(1-\theta)\iint_S\frac{\partial\eta^n}{\partial y}ds \\ +\Delta t\iint_S\frac{\tau_{sy}}{\rho}ds+\Delta t\iint_S(v_{dis}-V)DQ_{dis}ds \tag{7.126}$$

Applying the divergence theorem to equation (7.124) results in the following relationship:

$$\iint_S \eta^{n+1}ds+\Delta t\theta\int_{\partial S}\left(n_x DU^{n+1}+n_y DV^{n+1}\right)dl= \\ \iint_S \eta^n ds-\Delta t(1-\theta)\int_{\partial S}\left(n_x DU^n+n_y DV^n\right)dl+\iint_S D\,dis_s\,ds \tag{7.127}$$

Equations (7.127), (7.125) and (7.126) are discretised through the finite volume scheme described in sections (6.7) and (6.8). The result is:

$$\begin{aligned}
&\sum_{1}^{m}\sum_{j=1}^{6}\eta_j^{n+1}\iint_{sp}\psi_j\,ds+\Delta t\,\theta\sum_{i=1}^{m}\sum_{j=1}^{6}\left(n_x DU_j^{n+1}+n_y DV_j^{n+1}\right)\int_{\partial s_i}\psi_j\,dl=\\
&\sum_{1}^{m}\sum_{j=1}^{6}\eta_j^{n}\iint_{sp}\psi_j\,ds-\Delta t(1-\theta)\sum_{i=1}^{m}\sum_{j=1}^{6}\left(n_x DU_j^{n}+n_y DV_j^{n}\right)\int_{\partial s_i}\psi_j\,dl \qquad (7.128)\\
&+\sum_{1}^{m}\sum_{j=1}^{6}D\,dis_s\iint_{sp}\psi_j\,ds
\end{aligned}$$

$$\begin{aligned}
&\sum_{1}^{m}\sum_{j=1}^{6}DU_j^{n+1}\iint_{sp}\psi_j\,ds+\Delta t\,c_f\sum_{1}^{m}\sum_{j=1}^{6}DU_j^{n+1}\frac{\sqrt{(U_j^n)^2+(V_j^n)^2}}{D_j^n}\iint_{sp}\psi_j\,ds\\
&+\Delta t\,g\,\theta\,D^n\sum_{1}^{m}\sum_{j=1}^{6}\eta_j^{n+1}\iint_{sp}\frac{\partial\psi_j}{\partial x}\,ds=\\
&\sum_{1}^{m}\sum_{j=1}^{6}DU_j^{d}\iint_{sp}\psi_j\,ds-\Delta t\,g\,D^n(1-\theta)\sum_{1}^{m}\sum_{j=1}^{6}\eta_j^{n}\iint_{sp}\frac{\partial\psi_j}{\partial x}\,ds \qquad (7.129)\\
&+\Delta t\sum_{1}^{m}\sum_{j=1}^{6}\left(\frac{\tau_{sx}}{\rho}\right)_j\iint_{sp}\psi_j\,ds+\Delta t(u_{dis}-U)\,DQ_{dis}\sum_{1}^{m}\sum_{j=1}^{6}\iint_{sp}\psi_j\,ds
\end{aligned}$$

$$\begin{aligned}
&\sum_{1}^{m}\sum_{j=1}^{6}DV_j^{n+1}\iint_{sp}\psi_j\,ds+\Delta t\,c_f\sum_{1}^{m}\sum_{j=1}^{6}DV_j^{n+1}\frac{\sqrt{(U_j^n)^2+(V_j^n)^2}}{D_j^n}\iint_{sp}\psi_j\,ds\\
&+\Delta t\,g\,\theta\,D^n\sum_{1}^{m}\sum_{j=1}^{6}\eta_j^{n+1}\iint_{sp}\frac{\partial\psi_j}{\partial y}\,ds=\\
&\sum_{1}^{m}\sum_{j=1}^{6}DV_j^{d}\iint_{sp}\psi_j\,ds-\Delta t\,g\,D^n(1-\theta)\sum_{1}^{m}\sum_{j=1}^{6}\eta_j^{n}\iint_{sp}\frac{\partial\psi_j}{\partial y}\,ds \qquad (7.130)\\
&+\Delta t\sum_{1}^{m}\sum_{j=1}^{6}\left(\frac{\tau_{sy}}{\rho}\right)_j\iint_{sp}\psi_j\,ds+\Delta t(v_{dis}-U)\,DQ_{dis}\sum_{1}^{m}\sum_{j=1}^{6}\iint_{sp}\psi_j\,ds
\end{aligned}$$

If the velocity and/or free surface gradient are too steep at a location, the second order interpolation element may cause numerical oscillation. In this case the first order interpolation sub-elements are used locally (see equations 6.49 to 6.55). The risk of oscillation is checked by comparing the sign of the gradients of the solution quantity at the nodes of the quadratic interpolation.

Writing equations (7.128) to (7.130) for the FV cells and handling the boundary conditions results in a linear system of equations of the form $[A]\cdot[X]=[B]$. The coefficient matrix $[A]$ is not constant and may not necessarily be symmetric. An efficient iterative matrix solver, not restricted to symmetric matrices, is needed in order to solve the matrix equation at each time step. The preconditioned bi-conjugate gradient stabilised method (BI-CGSTAB) has been used in this regard, see van der Vorst (1992).

The open boundary nodes with known water level are handled by using Dirichlet condition for the water level and applying equations (7.129) and (7.130) for the velocity components. The open boundary nodes with known velocity are handled by using Dirichlet condition for the velocity components and applying equation (7.128) for the water level. The closed boundary nodes are handled by using the three FV relationships (7.128) to (7.130) while the impermeability condition of the boundary is applied when the line integrals on the cell boundary are determined in equation (7.128). All of the boundaries are handled by the normal interpolation elements (not the Neumann elements and the virtual nodes).

7.12 2D Advection Diffusion Model

The 2D advection diffusion equation for the generic depth averaged quantity Φ is given as:

$$\frac{\partial \Phi}{\partial t}+U\frac{\partial \Phi}{\partial x}+V\frac{\partial \Phi}{\partial y}=\bar{\nu}_t\left(\frac{\partial^2 \Phi}{\partial x^2}+\frac{\partial^2 \Phi}{\partial y^2}\right)+(\phi_s-\Phi)\,dis_s+S_D \tag{7.131}$$

where dis_s and Φ_s are the discharge and the ϕ value (for example, the temperature or salinity) of the point source respectively. S_D is the distributed source term such as the interaction with the bed.

Equation (7.131) is solved numerically using the operator splitting approach. The general formula is:

$$\frac{\partial \Phi}{\partial t}=\underbrace{\frac{\Phi^{n+1}-\Phi^{a}}{\Delta t}}_{diffusion\ step}+\underbrace{\frac{\Phi^{a}-\Phi^{n}}{\Delta t}}_{advection\ step} \tag{7.132}$$

The overall algorithm contains the following steps:

a) advection step

b) diffusion step

The solver for the advection step is identical to the semi-Lagrangian solver used in the 2D flow model. The solution algorithm for the horizontal diffusion step is identical to its counterpart in the flow model. For more details see sections (7.11.1) and (7.11.2). The Neumann condition at the input boundary is not allowed.

7.13 2D/3D Integration Algorithm

There is a generous literature due to research on single depth integrated two dimensional free surface simulation. There is also a valuable literature regarding to research on single hydrostatic or non-hydrostatic three dimensional free surface simulation. However, there are just a few research works on coupled the free surface model systems dynamically. Still there are fewer research works on integrating 2DH and fully 3D free surface flow and quality models together.

Several 1D-2DH integrated modelling systems have been developed for integrated flood modelling. The 2DH model in these systems is a simplified version of the shallow water equations, in order to be efficient for the over-land region. There are also research works on integrated watershed-hydrodynamic model systems and on groundwater-free surface model systems, which are beyond the scope of the present work.

The nested grid technique has been used widely in single 2DH models and also in some 3D models that are based on structured computational grid. There is also a research work (Namin, 2003) which has used the nested grid technique for coupling a structured 3D model to a structured 2D one. All of the nested grid models are based on interpolation and averaging for data exchange at the interface of the nested grids. This procedure is not numerically conservative.

Two approaches have been employed in the literature for combining multi-dimensional free surface models. One approach is to use a simpler model in the entire solution domain and a more complex model in confined sub-domains. The simpler model provides boundary conditions for the more complex model, but they are not coupled. This approach is called the hybrid modelling. The other approach is is to fully couple the models (Wu W., 2007; Sebastian and Shu, 2003; McAnally et al., 1986; Wu and Li, 1992; Zhang, 1999; Verwey, 2001).

The current fully coupled 2D-3D model systems are based on the interpolation/averaging procedure in order to exchange data between the models. Therefore they are not fully conservative at the interface. They use mesh overlapping or nested grid techniques at the interface.

McAnally et al. (1986) used a hybrid (not coupled) type of mixed 2DH-3D model to study the salinity intrusion in New York harbour. Wu and Li (1992) applied a coupled 1D-2DH quasi-steady model system in the study of sedimentation problems in the China's Three Gorges project (TGP). The entire study domain was divided into four 1D and four 2DH reaches. Yu, L. et al (1998) presented a hybrid type (no coupled) mixed 2DH-3D flow model to analyse flow in an estuary. Their hybrid system consisted of a three dimensional finite volume approach sub-model being nested with a two dimensional finite element model. The 2D FE model was explicit on unstructured grid, while the 3D FV model used structured grid. Zhang (1999) applied a coupled

1D-2DH model system in order to simulate the flow and sediment transport at the Yellow River mouth, based on the multi-block approach. However, it was not a fully implicit coupling algorithm. Kashefipour et al (2002) presented a coupled 1D-2DH model system in order to simulate the hydro-environmental parameters of water at Ribble Estuary, UK. He used a multi-block approach with domain overlapping at the interface. Each model received its boundary condition at the interface, from inside the other model's domain. Multi-block approach with domain overlapping is common in model coupling. However, it is not a fully implicit coupling algorithm.

Namin (2003) introduced a fully 3D model nested with a 2D shallow water one. Both of the models use structured computational grid. The 3D model is based on solving the vertical planes with a 2DV module filling the domain in each direction as an ADI method. The boundary conditions for the 3D domain are provided implicitly from the the 2DH model. The linking procedure is based on the behaviour of the ADI algorithm, which solves the 2D domain, line by line via the tri-diagonal system, and solves the 3D domain, vertical plane by vertical plane via a block tri-diagonal system. The tri-diagonal systems are solved by the double sweep algorithm. The two tri-diagonal systems are mixed in a single double sweep algorithm over the entire model wherever needed. The integrated system of Namin et al works very well in situations where the structured grid is sufficient. However, it depends on interpolation and integration of data at the link-boundary, as any kind of nested model, when the resolution of the 3D computational grid is different from the 2D one. The technique of Namin et al. can not be extended to the unstructured grid without difficulties and loss of efficiency. It depends completely on the ADI algorithm and the structured grid characteristics.

Katopodes N. D. and Kao K. (2003) presented nested grid model for nearshore hydrodynamics in which, two different 3D models coupled together. The coarse model is based on a fixed Cartesian grid. The nested model is based on a variable curvilinear grid. The authors focused on developing a subspace projection method for time integration and non-reflective boundary conditions at the interface. They used the multi-block approach with domain-overlapping at the interface and managed the nested grid to expand gradually in the overlap region until it reaches the desired scale in consistency with the coarse grid. The coupling was based on the interpolation/averaging algorithm. Lin B. et al. (2006) presented a sophisticated 1D-2DH integrated system using the 1D ISIS model and the 2DH DIVAST model. This model system is for predicting flood inundation levels in complex river basins. The ISIS and DIVAST model is linked by a weir equation, in which the volume of flow from the 1D domain to the 2D domain is determined by the water level difference.

A fully implicit 1D/2D integrated model has been developed in WL Delft Hydraulics (Sobek-1D2D) for modelling river and floodplain hydraulics together. It solves the complete set of equations (from the two domains) together using a matrix solver. The two models are coupled via an interface in the horizontal plane that divides the river at the floodplain level. The 2D model uses structured computational grid.

No one of the present free surface 2D-3D coupling techniques are available to present a fully dynamic and conservative integration procedure with implicit behaviour without enforcing solution iterations. The integration strategy in the present work is a novel algorithm which is free from interpolation/averaging so it is fully conservative and accurate at the interface, it is based on unstructured grid, and does not restricts the implicit behaviour of the two models in simulation time step selection.

It is possible to apply a two-way data transformation standard between individual models, by back and forth data transformation at the shared model boundaries. However it results in a fully explicit linking method which restricts the time step limitation. Iteration may also be needed at each time step. This approach decreases the overall efficiency when all of the models are implicit.

As mentioned earlier, one approach for integrating two implicit models is to share a limited part of the solution domain close to the interface of the models. The interface is the contact line/surface between the two models. Figure (7.3) illustrates a schematic of this algorithm. This integration algorithm is not fully implicit, because the boundary conditions of the two models at their internal boundary are not from the new time step. Therefore they are not accurate. This inaccuracy will spreads into the solution domain gradually. If the solution time step of the models is much smaller than the time scale of variation in the boundary conditions and/or the solution domain, the inaccuracy will not be noticeable.

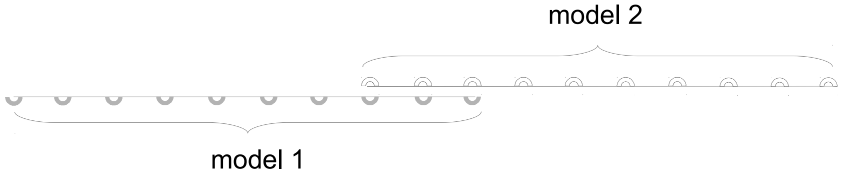


Figure 7.3. Model integration by sharing part of the solution domain

A more accurate coupling approach is to add an extra model which covers the interface and a limited part of the solution domains around it. The extra model simulates the limited area implicitly or explicitly in order to define the solution values on the interface at the new time step. This procedure is applicable if the interface is located at a region where the effect of hydrodynamic pressure is negligible. Figure (7.4) illustrates a schematic view of the three solution domains.

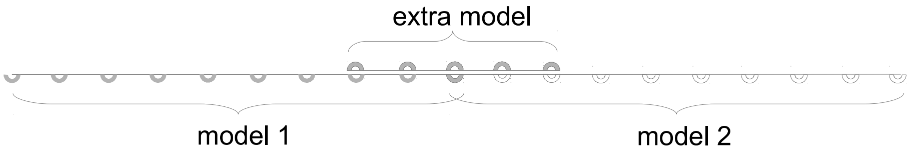


Figure 7.4. Model integration by adding an extra model

The extra model in figure (7.4) receives its boundary conditions from the two main models. The boundary conditions of the extra model are explicit and therefore they are not accurate. However the solution domain size is selected in such a way that the effect of the boundaries does not reach the interface location in the extra model during one time step. In this way, the proposed algorithm is more accurate than the previous one. However, it needs an initial knowledge from the flow characteristics to determine the minimum grid-overlapping extension needed.

The mentioned algorithms above do not deal with the structure of the models that are integrated. It is possible to develop more efficient integration algorithms when the structures of the models are accessible and identical. In this research program a new coupling algorithm has been introduced. The main idea is to use fractional step algorithms for the models and then integrate them fraction by fraction. Hybrid finite volume cells have been introduced for the link nodes which are partly in the 2D domain and partly in the 3D one. The present 2D and 3D models are based on similar operator splitting schemes. They have been coupled fraction by fraction. Common fractions are handled together with similar procedures and data are exchanged directly. The need for the interpolation/averaging algorithm at the interface has been eliminated by using unstructured computational grid and introducing hybrid finite volume cells. The hybrid finite volume cells are partly in the 2D domain and partly in the 3D domain area. A hybrid finite volume cell produces a unique discretised equation for a computational node, while the discretised equation is partly supported by nodes within the 2DH domain and partly by the 3D domain.

The advection steps of the two models are integrated and solved jointly. Then the horizontal diffusion steps of the two models are integrated and solved. Afterwards the free surface steps of the two models are integrated and solved together. Finally the remaining computations of both models are carried out. The following subsections present details of each step of the integration procedure .

7.13.1 Integrating the Advection Sub-models

When a backward streamline of the 2D, or 3D model crosses over the interface boundary, it continues movement into the 3D, or 2D domain using the velocity data from that domain until reaching the position of the previous time step (see figure 7.5). When the backward streamline of the 2D model falls in the 3D domain, it uses the depth averaged velocity values of the 3D model. The vertical velocity components are neglected in this regard. When the backward streamline of the 3D model falls in the 2D domain, the velocity data are determined based on the depth averaged values. The vertical distribution of the horizontal velocity is assumed to be logarithmic in the 2D domain. The temperature and salinity values are assumed to be constant in depth.

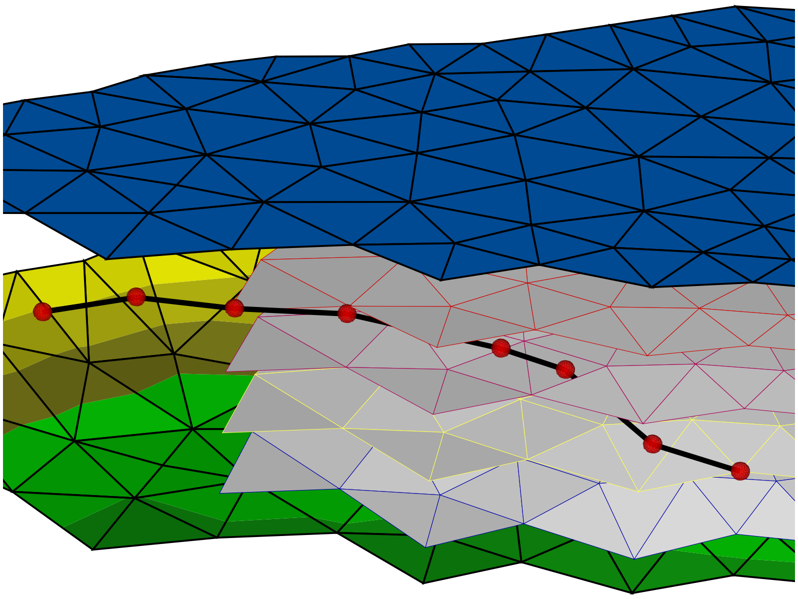


Figure 7.5. Continuing the backward tracking in the 2D depth-averaged domain

7.13.2 Integrating the Horizontal Diffusion Sub-models

Recalling that the horizontal diffusion solver is explicit, the global time step is assumed to satisfy the stability condition. The nodal values at the new time step on the interface boundary may be calculated one by one explicitly using their neighbouring nodes from the two domains (2D and 3D). The introduced hybrid 2D-3D finite volume cells are used for deriving the discretised form of the governing equations at the link nodes. Part of these cells located in the 2D domain and the other part in the 3D domain. In this way, the interface nodes are linked explicitly to nodal data from both sides (2D and 3D domains). The vertical velocity component at the interface nodes assumed to be zero. After defining the interface node values at the new time step, the horizontal diffusion steps of the 2D and 3D models are carried out normally using Dirichlet boundary conditions at the interface nodes.

The interface nodes of the 2D model have some neighbouring nodes (vertical columns) in the 3D field. The depth averaged values of the vertical columns may be used in this regard. On the other hand, the interface nodes of the 3D model (on each sigma plane), have some neighbouring nodes in the 2D domain. The required data in the same z level is determined from the depth averaged values. The vertical distributions of the solution quantities are assumed to be known in the 2D area, as described for integrating the advection steps.

7.13.3 Integrating the Free Surface Sub-models

The free surface equations for the two models are combined in order to produce a unified system of linear equations. The solution procedures for the free surface in the 3D and the 2D models are identical. A unique system of equations is produced for the integrated free surface solution which involves both the 3D and 2D domains. Equations (7.54) to (7.56) are applied to the FV cells in the 2D domain and equations (7.128) to (7.130) are applied to the FV cells in the 3D domain. The computational nodes and the FV cells on the interface of the two models are handled as normal internal cells. The only difference is that part of the shared FV cells, which is in the 3D domain, is handled by equations (7.54) to (7.56) and the other part, which is in the 2D domain, is handled by equations (7.128) to (7.130). The global system of equations is solved with the preconditioned bi-conjugate gradient stabilised method (BI-CGSTAB). Each one of the individual models, the 3D and the 2D versions, are also based on the same matrix solver when working independently.

There is no need to global computational grid or global numbering system of the nodes in the present algorithm. Discretisation of the depth integrated forms of the continuity and momentum equations in the 2D and 3D domains, results in two distinguished matrix equations. However, each matrix (and right hand side vector) involves just half of the data regarding to the hybrid cells on the interface. Then the matrices are linked together at the link nodes in order to be solved as a global matrix equation.

7.14 Summary

The numerical techniques and discretisation procedures which are used in the 2D and 3D models of the present study have been presented in this chapter. Four new numerical models have been created and linked together. They are the 3D hydrodynamic model, 3D advection-diffusion model, 2D depth averaged hydrodynamic model, and 2D depth averaged advection diffusion model. The solution procedures for all of the models are based on the fractional steps algorithm. Each fraction includes one or more physical processes.

Sections (7.2) described the general algorithm of the 3D hydrodynamic model and listed the corresponding solution steps. They include pure advection, horizontal diffusion, free surface simulation, vertical diffusion, hydrodynamic pressure correction and velocity projection steps. Sections (7.3) to (7.8) characterised the above solution steps. Section (7.3) illustrated the semi Lagrangian backward tracking characteristic method that has been used for the pure advection solver. Section (7.4) described the horizontal diffusion step. This process is simulated layer by layer. The new hybrid finite element - finite volume algorithm has been used as described in chapter (6). The horizontal diffusion processes are solved jointly. It is a new algorithm which handles the velocity condition at the closed boundary implicitly. The discretisation procedure for the horizontal diffusion terms involves new schemes. The shear stresses are computed in the horizontal plane, while their gradients are computed in the sigma layers. Section (7.5) addressed the solution procedure for the free surface through the hybrid FV scheme and the corresponding divergence theorem. The depth averaged versions of the continuity equation and the remaining parts of the x- and y-momentum equations have been solved implicitly. The solution procedure for the effect of the vertical diffusion, the free surface slope and the bed friction on the velocity components has been described in section (7.6). The depth integrated velocity from the previous section, has been used here instead of recalculating the free surface slope. This is a new scheme which guaranties the adjustment of the calculated velocity with the free surface slope. Section

(7.7) illustrated the solution algorithm for the hydrodynamic pressure. The 3D FV cells in the real space have been used in this regard. A new algorithm has been presented for calculating the surface integrals on the lateral boundaries of the 3D FV cells. Section (7.8) illustrated the velocity projection step into the convergence-free field. Section (7.9) described the optional recalculation procedure for the free surface by the final velocity field. It is applicable to the rare situations where the hydrodynamic pressure influences the water level.

Section (7.10) described the numerical algorithm of the 3D advection diffusion model. The model is based on the fractional steps algorithm with the pure advection, the horizontal diffusion and the vertical diffusion steps. The solution algorithms for each fraction are identical to its counterpart in the 3D hydrodynamic model. Section (7.11) illustrated the 2D flow model. The 2D flow model is also based on the fractional steps algorithm with the advection step, the diffusion step, and the free surface/barotropic pressure and bed friction step. The solution algorithm for each fraction is identical to its counterpart in the 3D hydrodynamic model. A conservative version of the semi-Lagrangian algorithm has also been introduced to the 2D model, with new scheme for determining the cell boundary integrals. Section (7.12) described the 2D advection diffusion model with its solution fractions; pure advection and diffusion. The solution algorithm for each fraction is again identical to its counterpart in the 3D hydrodynamic model.

Section (7.13) illustrated the algorithm for integrating the 3D and the 2D models. A new implicit and efficient coupling algorithm has been introduced, which couples same operators (solution steps) of the two models. The backward tracking procedure of the semi-Lagrangian advection step continues the tracking when it moves from the 3D domain into the 2D one or vice versa. The free surface steps of the two models are solved jointly in a unified global matrix equation. There is no need for shared solution domain between the 2D and 3D models. It is possible to switch off a physical process without affecting the integration procedure. The matrix equations for several parts of the model contain a constant coefficient-matrix, which is inverted only once and used at each time step. Thus the algorithm of the solution procedure is fast and efficient.

8 Model Evaluation

8.1 Vertical Diffusion – Wind Induced Current

Horizontal and vertical diffusion processes jointly construct the fully 3D diffusion phenomenon together. However, the vertical diffusion is usually the critical diffusion process in a real-world three dimensional simulation problem. A good case for testing the vertical diffusion model is the simulation of the Coriolis effect on the drift current (Ekman spirals).

Both of the drift current and slope current distributions are affected by the Coriolis effect. The drift current is the shear velocity produced by the wind shear stress at the free surface and is restricted by the bed shear stress. The slope current is the velocity profile which is produced by the pressure gradient at the water column (due to the free surface slope) and is restricted by the bed shear stress. The spiral distribution appear due to the disturbance of the geostrophic equilibrium just below the surface by the wind shear (drift current) and just above the ocean floor by the bottom shear (slope current). The geostrophic equilibrium condition means that the pressure gradient (or shear stress) and Coriolis force cancel each other out in the momentum equation.

The analytical solutions for the Ekman profiles were obtained for the steady flow and constant eddy viscosities by V. W. Ekman in 1905. The solution for the drift current is determined through a Neumann wind shear boundary condition at the surface.

Before simulating the Coriolis effect on the wind-driven current, the pure wind-driven current is addressed in order to compare it with the analytical solution and with the changes in velocity distribution after introducing the Coriolis force.

The analytical solution of the steady state pure wind-driven current with wind velocity in the y-direction may be obtained by solving the following equation:

$$\textit{Equation}:\quad \frac{\partial}{\partial z}\left(\nu_v \frac{\partial v}{\partial z}\right)=0$$

$$\textit{Free Surface B.C.}:\quad \left(\nu_v \frac{\partial v}{\partial z}\right)_\eta = \frac{\tau_\eta}{\rho} = \frac{\rho_{air}}{\rho_{water}}.C_w.W.W_y \qquad \textit{Bottom B.C.}:\quad \frac{\partial v_b}{\partial z} = \frac{v_b}{d_o \cdot \ln\left(\frac{d_o}{(k_s/30)}\right)} \qquad (8.1)$$

$$\textit{Simplified Bottom B.C.}:\quad v_b = 0.0$$

where d_0 is the normal distance between the real bed and the first mesh layer above it (reference level).

The no-slip condition for the bottom B.C. results in a vertical velocity distribution which is not exactly the same as the result of wall function. However, the analytical solution of the flow with the Coriolis force (at the next stage) has been obtained using the no-slip condition. Therefore the same boundary condition will be used here.

Equation (8.1) illustrates that the steady state velocity distribution is exactly linear and the vertical gradient of the velocity is controlled by the shear stress:

$$\left(\frac{\partial v}{\partial z}\right)_\eta = \frac{1}{\nu_v}\frac{\tau_\eta}{\rho} = \frac{1}{\nu_v}\frac{\rho_{air}}{\rho_{water}}.C_w.W.W_y \qquad (8.2)$$

The above problem has been simulated numerically on a 200km by 200km solution area with the water-level open boundary condition being used at all of the side boundaries. Figure (8.1) illustrates the plane view of the numerical mesh. Other model parameters are: water depth = 68.8m (this is half of the depth of frictional influence in the next test for Ekman profile), vertical eddy viscosity = 0.1 m^2/s, wind velocity = 32 m/s, and the velocity drag coefficient = 0.002513. This problem has been solved numerically with the time step = 400s. Figures (8.2) and (8.3) illustrate the result of simulation. The predicted velocity is in the wind direction and the steady state free surface velocity is 2.03 m/s and the vertical gradient of the velocity is 0.030. This is in good agreement with the analytical solution in which the free surface velocity is 2.01 m/s and the vertical gradient of the velocity is 0.030. Similar simulations have been done for wind speed = 5 m/s and vertical eddy viscosity = 0.01, 0.05 and 0.1 m^2/s Figure (8.4) illustrates a comparison of the simulation results with the analytical solutions. It shows that the numerical results are in close agreement with the analytical ones. The maximum errors for the velocity at the free surface are 1.2%, 1.2%, and 0.9% for the viscosities $\nu = 0.1$ m^2/s, $\nu = 0.05$ m^2/s, and $\nu = 0.01$ m^2/s respectively.

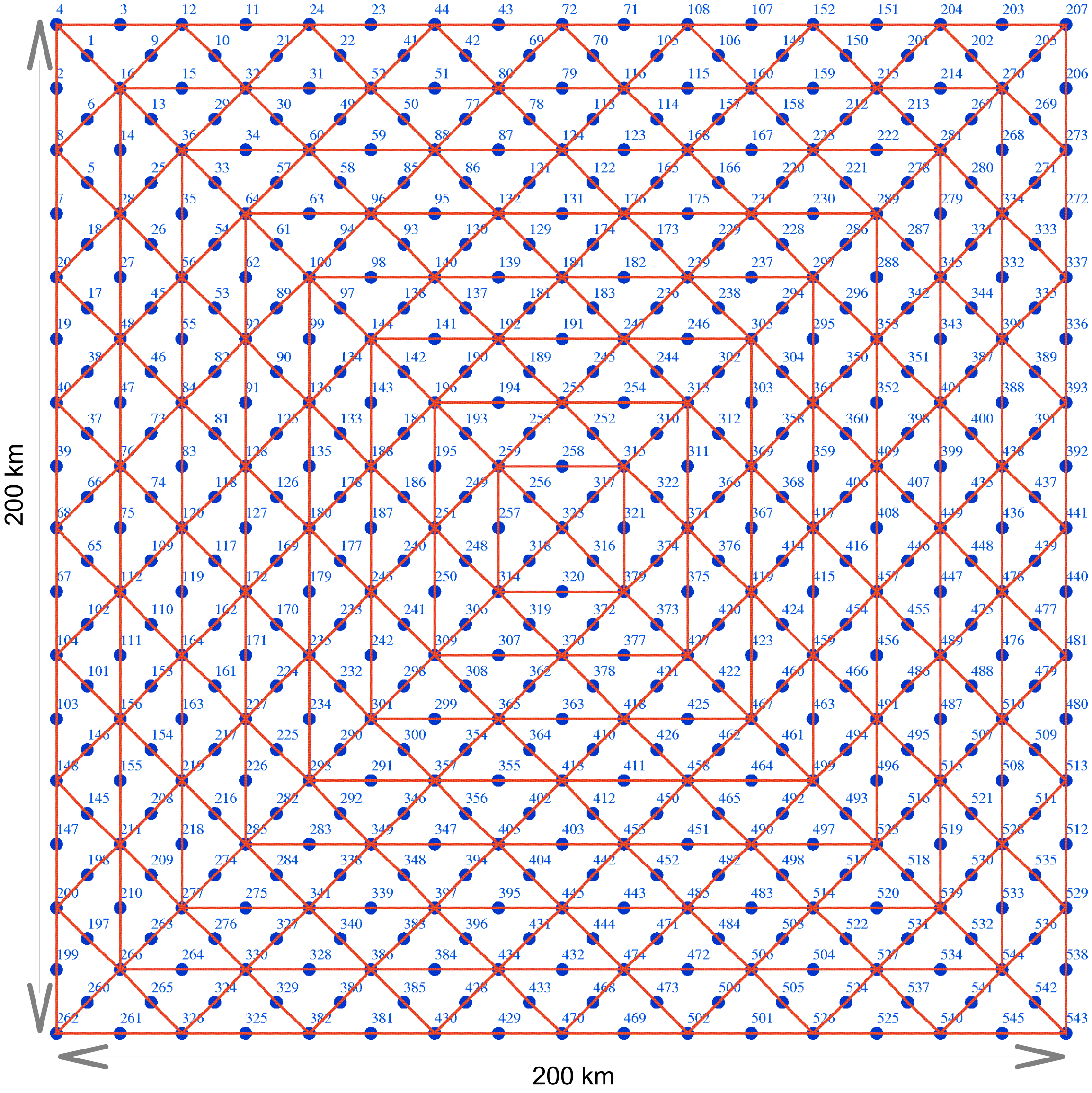


Figure 8.1. Model grid deployed in diffusion test

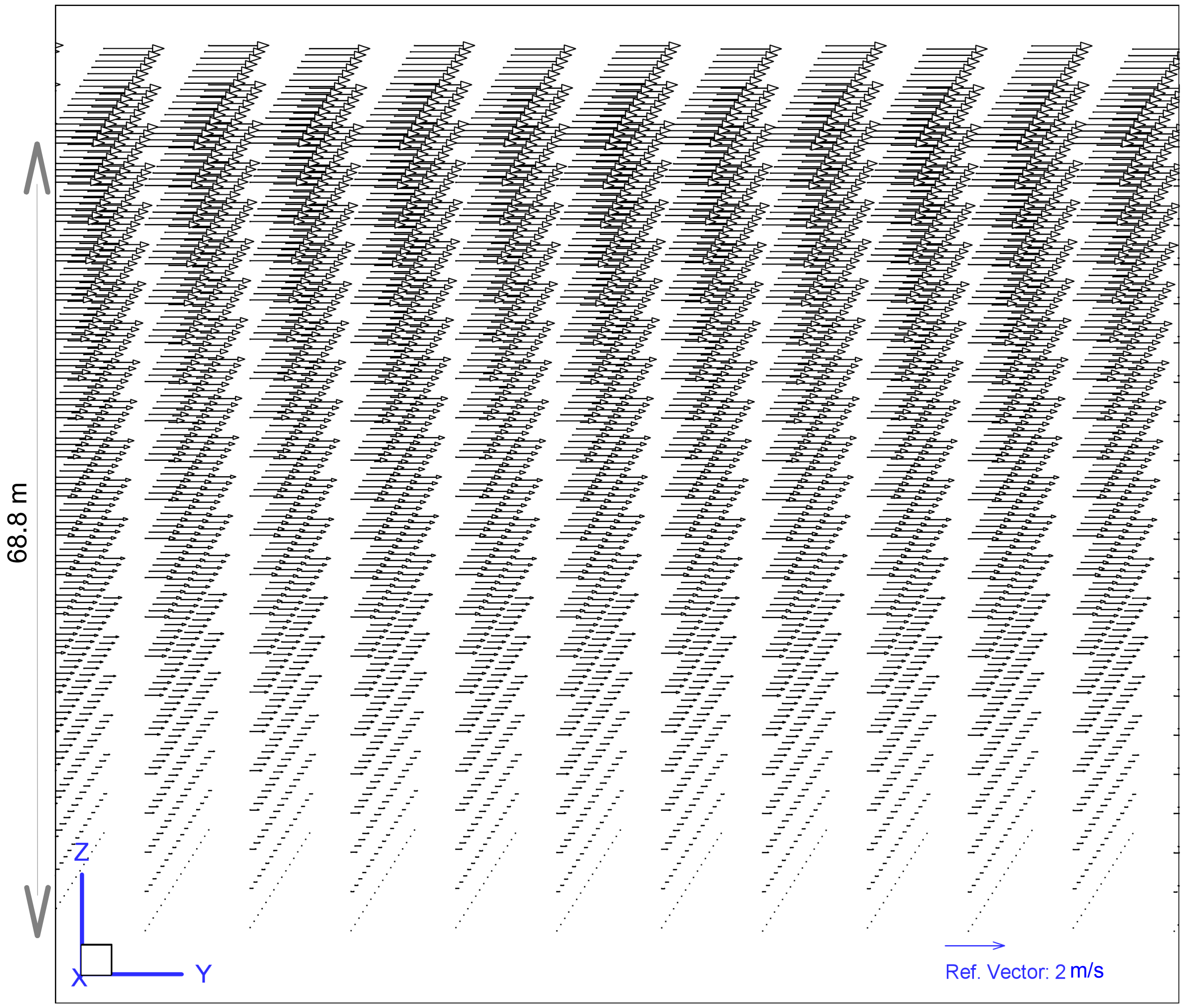


Figure 8.2. Simulated velocity distribution for pure drift current

wind speed = 32 m/s, eddy viscosity = 0.1 m^2/s

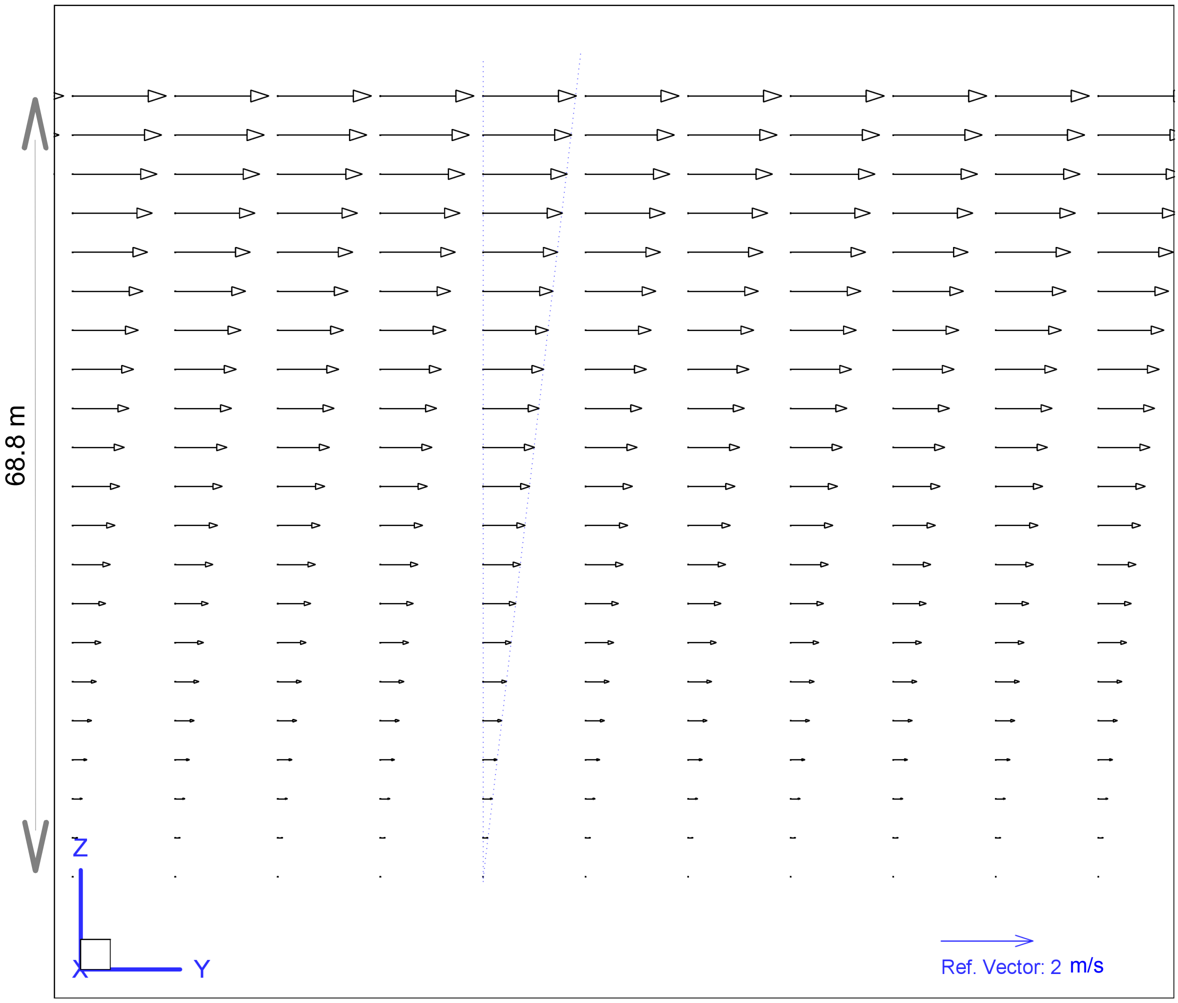


Figure 8.3. Simulated velocity distribution for pure drift current

wind speed = 32 m/s, eddy viscosity = 0.1 m^2/s

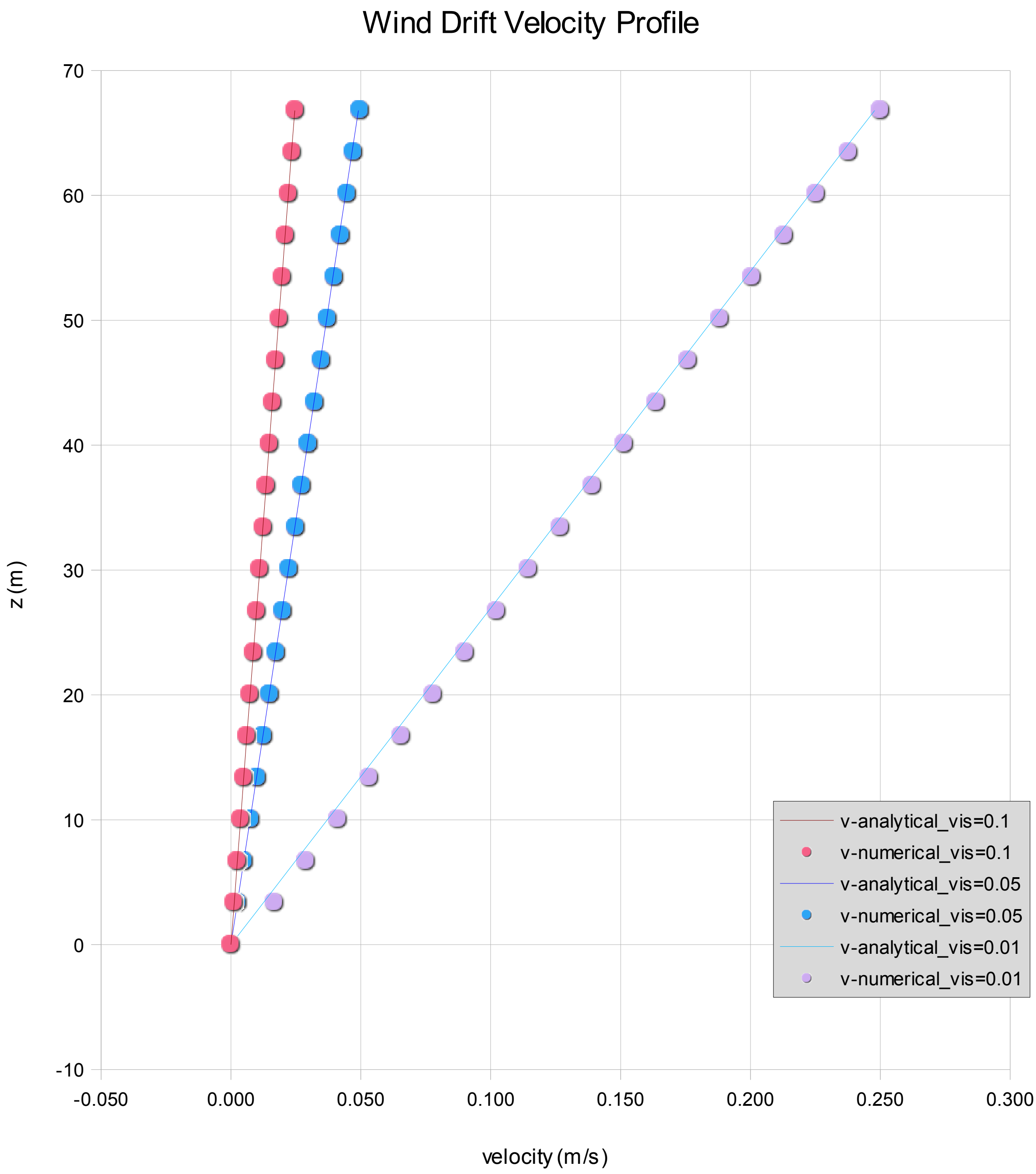


Figure 8.4. Comparisons of the numerical and analytical model predicted wind current profile for different vertical eddy viscosities (wind speed = 5 m/s)

The analytical solution of the steady flow with the Coriolis force on the wind-driven current is obtained by solving the following complex number equation:

$$\begin{aligned}
&\nu_v \frac{d^2 V}{d z^2} - i f V = 0 \\
&B.C.: \\
&T = \rho \nu_v \frac{d V}{d z_\eta} \qquad at \quad z = 0 \\
&V = 0.0 \qquad\qquad at \quad z = -H \qquad \textit{for finite depth}
\end{aligned} \tag{8.3}$$

where i is the imaginary part of the complex unit vector, f is the Coriolis coefficient and z is zero at the free surface and is positive upward. T is the complex wind shear stress:

$$T = \tau_x + i \cdot \tau_y \qquad \tau_x = \rho \nu_z (\partial u / \partial z) \qquad \tau_y = \rho \nu_z (\partial v / \partial z) \tag{8.4}$$

The analytical solution of equation (8.3) is (Jankowski, 1999):

$$V = V_o \frac{\sinh[(1+i)\gamma(z+H)]}{\cosh[(1+i)\gamma H]} e^{-\frac{\pi}{4}i} \tag{8.5}$$

where $\gamma = \sqrt{f / 2\nu_z}$ and the parameter V_o is:

$$V_o = u_o + i v_o = \frac{T}{\rho(\nu_z |f|)^{1/2}} = \frac{\tau_{wx}}{\rho(\nu_z |f|)^{1/2}} + i \frac{\tau_{wy}}{\rho(\nu_z |f|)^{1/2}} \tag{8.6}$$

General solution of equation (8.5) results in lengthy analytical formulae for the real velocity components u and v. Assuming that the wind blows in the y-direction, the solution is (Jankowski, 1999):

$$\begin{aligned}
u = {} & v_o \frac{\sqrt{2}}{2} \left(\frac{[\sinh(\gamma\zeta)\cos(\gamma\zeta) - \cosh(\gamma\zeta)\sin(\gamma\zeta)]\cosh(\gamma H)\cos(\gamma H)}{\cosh^2(\gamma H)\cos^2(\gamma H) + \sinh^2(\gamma H)\sin^2(\gamma H)} \right) \\
& + v_o \frac{\sqrt{2}}{2} \left(\frac{[\sinh(\gamma\zeta)\cos(\gamma\zeta) + \cosh(\gamma\zeta)\sin(\gamma\zeta)]\sinh(\gamma H)\sin(\gamma H)}{\cosh^2(\gamma H)\cos^2(\gamma H) + \sinh^2(\gamma H)\sin^2(\gamma H)} \right)
\end{aligned} \tag{8.7}$$

$$\begin{aligned}
v = {} & v_o \frac{\sqrt{2}}{2} \left(\frac{[\sinh(\gamma\zeta)\cos(\gamma\zeta) + \cosh(\gamma\zeta)\sin(\gamma\zeta)]\cosh(\gamma H)\cos(\gamma H)}{\cosh^2(\gamma H)\cos^2(\gamma H) + \sinh^2(\gamma H)\sin^2(\gamma H)} \right) \\
& - v_o \frac{\sqrt{2}}{2} \left(\frac{[\sinh(\gamma\zeta)\cos(\gamma\zeta) - \cosh(\gamma\zeta)\sin(\gamma\zeta)]\sinh(\gamma H)\sin(\gamma H)}{\cosh^2(\gamma H)\cos^2(\gamma H) + \sinh^2(\gamma H)\sin^2(\gamma H)} \right)
\end{aligned} \tag{8.8}$$

The wind driven current with the Coriolis force has been simulated numerically under the same conditions and model parameters as the pure wind test case. The time step used is 400s. Figures (8.5) and (8.6) illustrate the simulated velocity distribution. Figure (8.7) illustrates a comparison of the analytical and numerical solutions (by 11 and 21 layers). It shows that the numerical result is in good agreement with the analytical one, with the maximum velocity error

1.0% (0.01 m/s) and maximum direction error 0.6 degrees. Figure (8.7) shows that the accuracy has increased by increasing the number of computational points in the vertical direction.

Similar simulations have also been undertaken for water depths = 33.4 and 160 m. Figures (8.9) and (8.10) illustrate the comparison of the simulation results with the analytical ones. They show that the results are in good agreement. The maximum velocity error for depth=33.4m is 0.1% (0.0009m/s) and direction 0.3 degrees. The maximum velocity error for depth=160m is 2.3% (0.02m/s) and direction 1.3 degrees. These error values are satisfactory when the limited vertical resolution ($\Delta Z = 9\text{m}$) is considered into account. The analytical results are very sensitive to the water depth. Figure (8.11) illustrates that if the water depth is assumed to be 159m (instead of 160m) in the analytical solution, the results are similar to the numerical results of 160m depth.

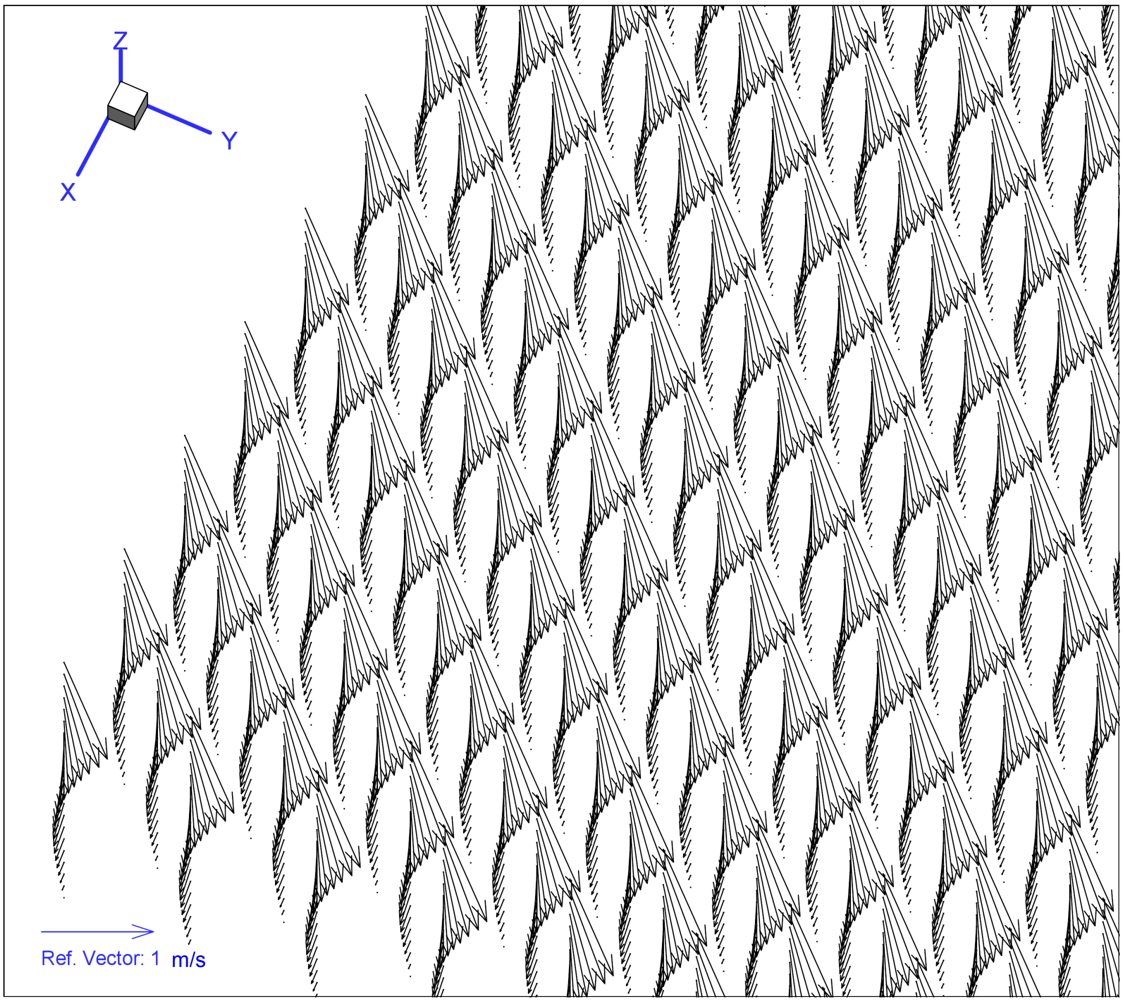


Figure 8.5. Simulated velocity distribution for the Ekman problem (combination of wind shear stress and the Coriolis force).

depth = 66.84m, wind speed = 32 m/s, eddy viscosity = 0.1 m^2/s, latitude = 50^o

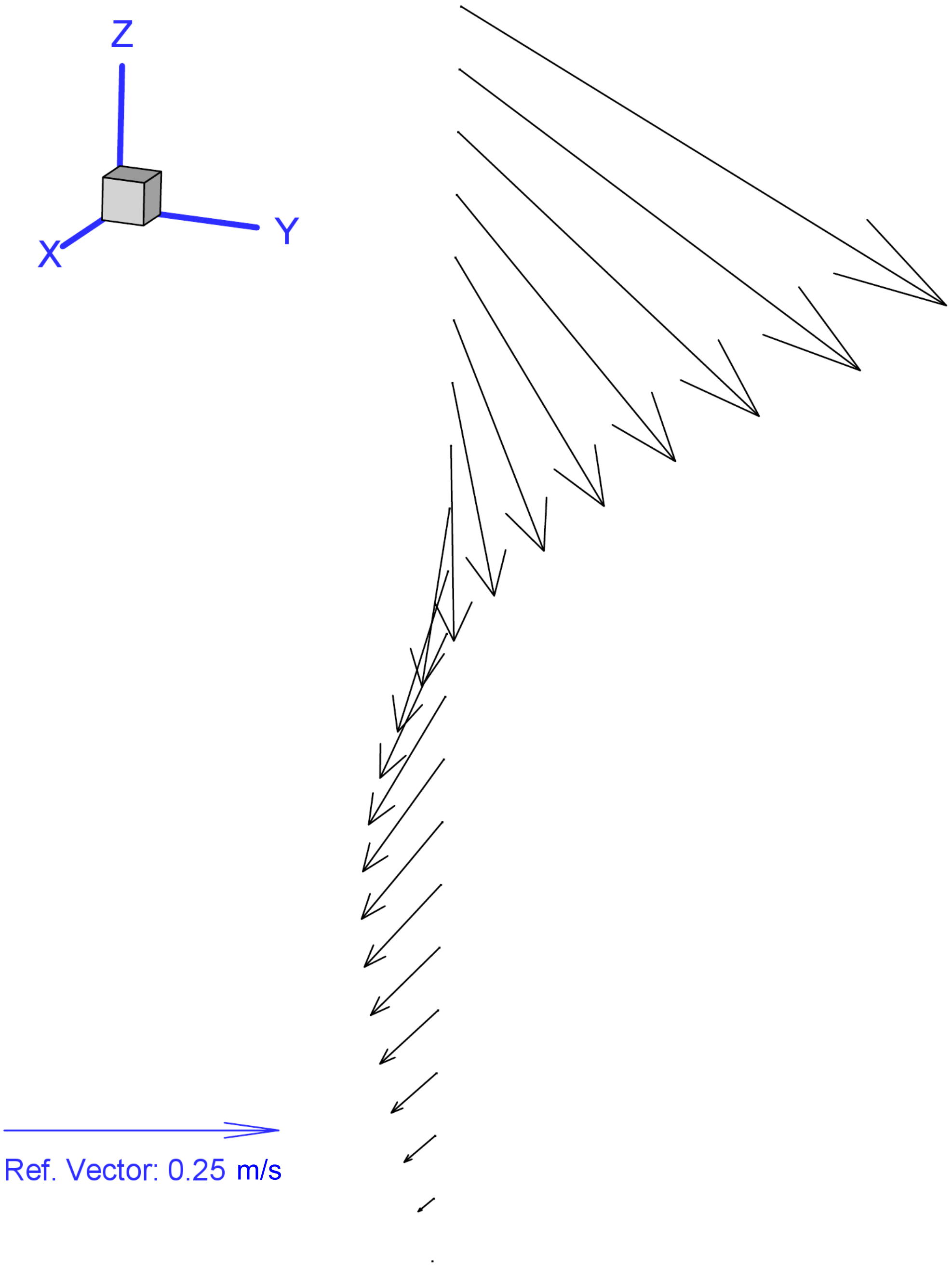


Figure 8.6. Simulated velocity distribution for the Ekman problem (combination of wind shear stress and the Coriolis force).

depth = 66.84m, wind speed = 32 m/s, eddy viscosity = 0.1 m^2/s, latitude = 50^o

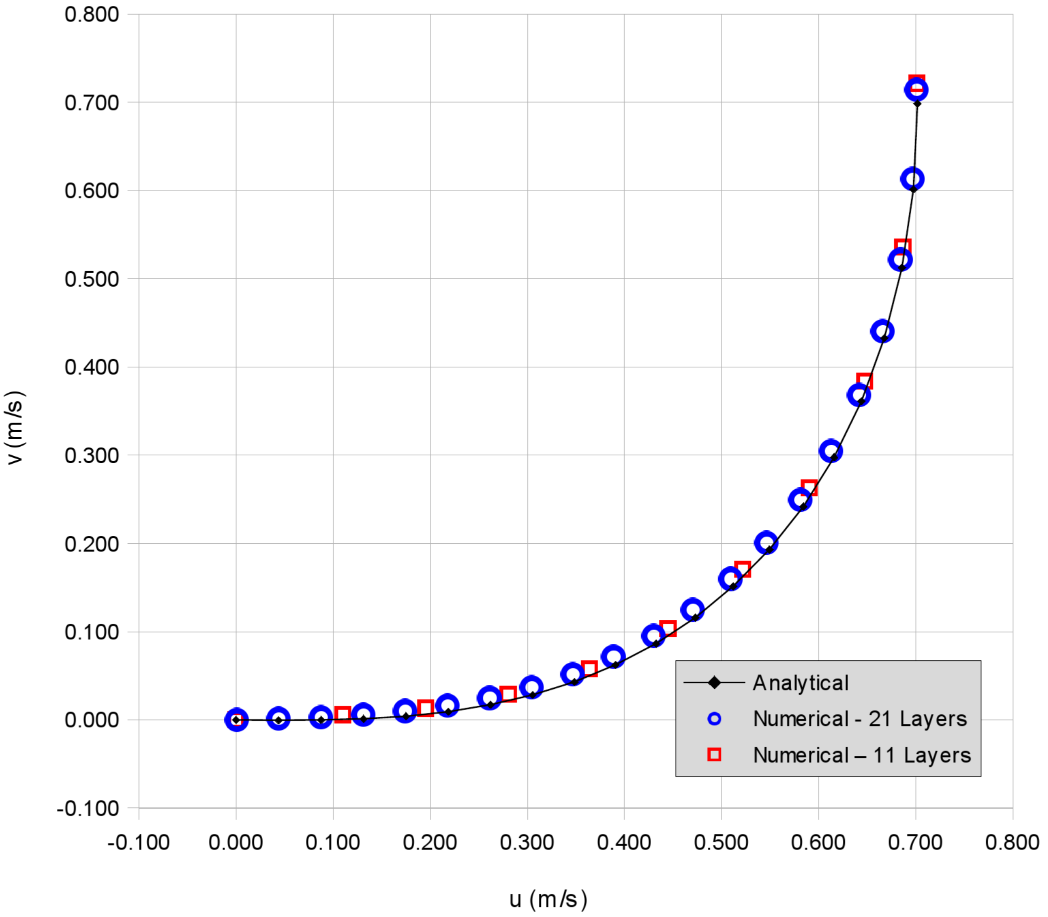


Figure 8.7. Comparison of the numerical and analytical Ekman profile
depth = 66.84m, wind speed = 32 m/s, eddy viscosity = 0.1 m^2/s, latitude = 50°

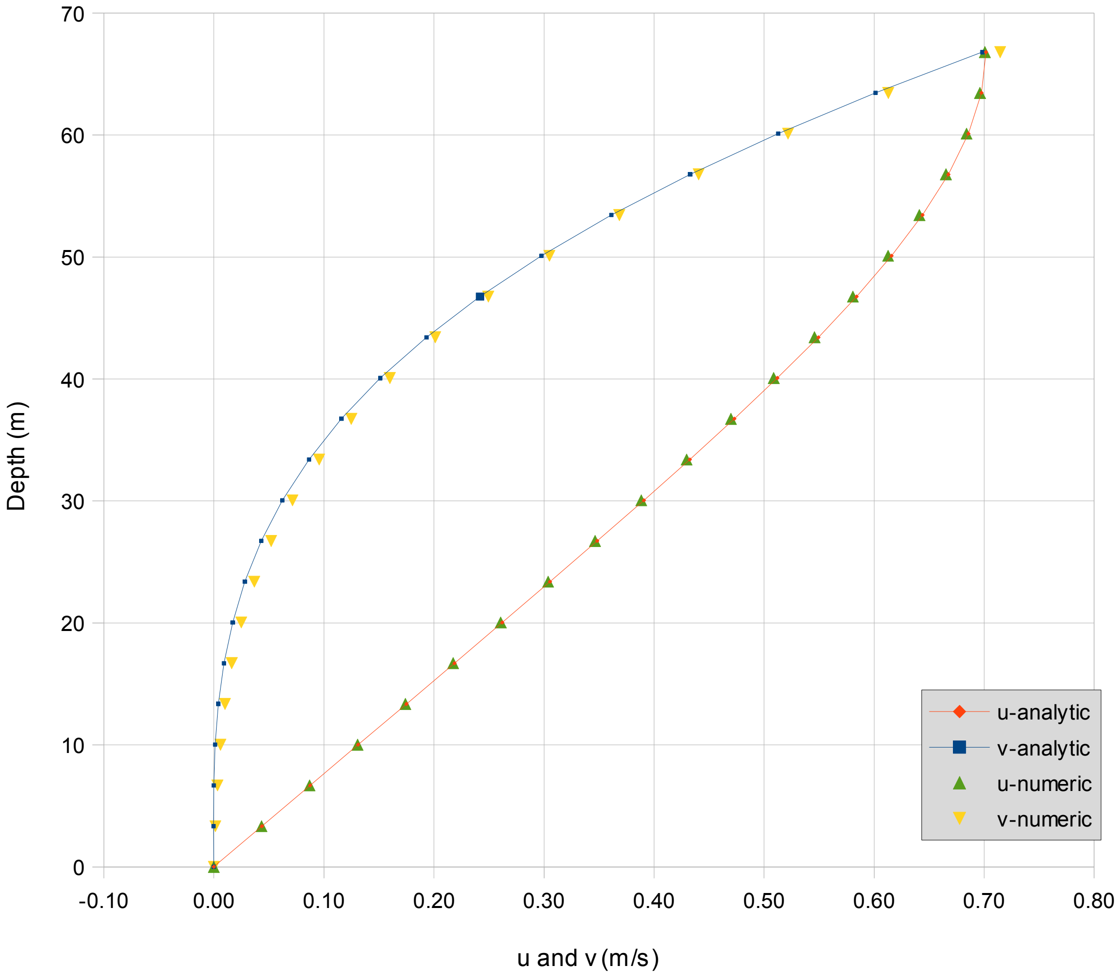
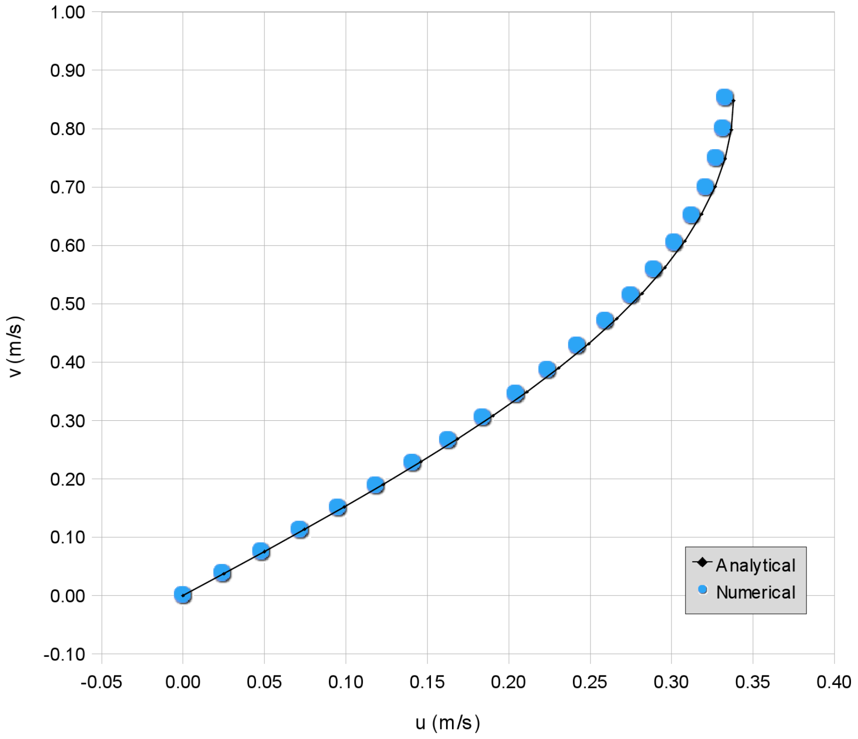


Figure 8.8. Comparison of the numerical and analytical Ekman profile

depth = 66.84m, wind speed = 32 m/s, eddy viscosity = 0.1 m^2/s, latitude = 50^o



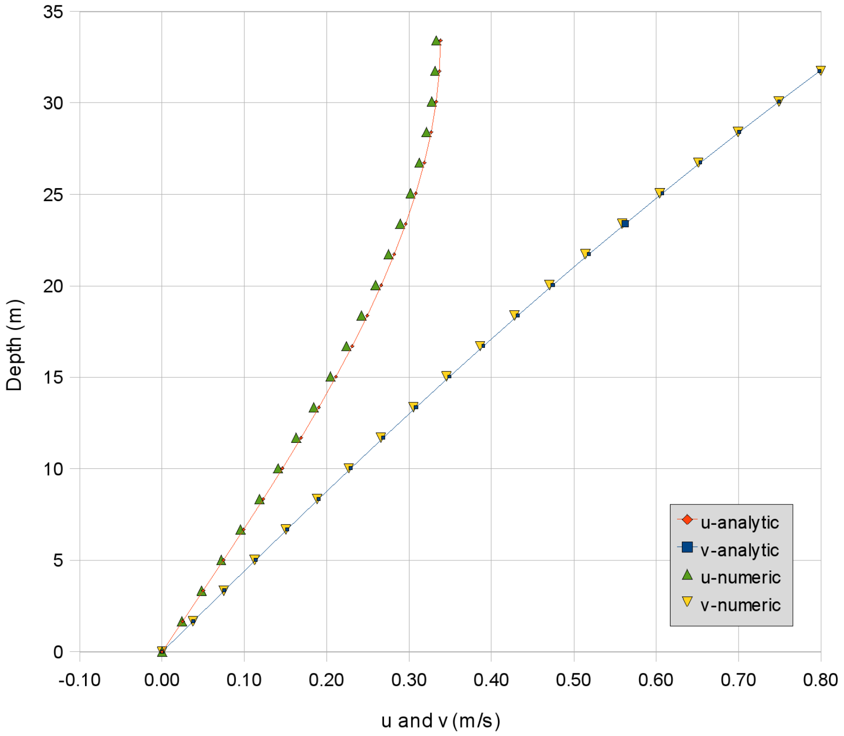
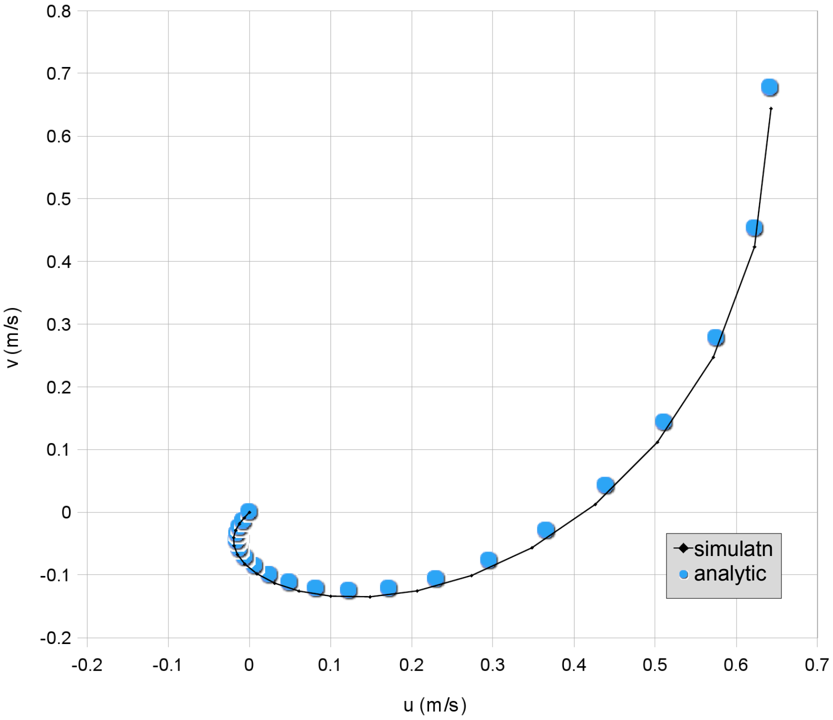


Figure 8.9. Comparison of the numerical and analytical Ekman profile

depth = 33.4m, wind speed = 32 m/s, eddy viscosity = 0.1 m^2/s, latitude = 50°



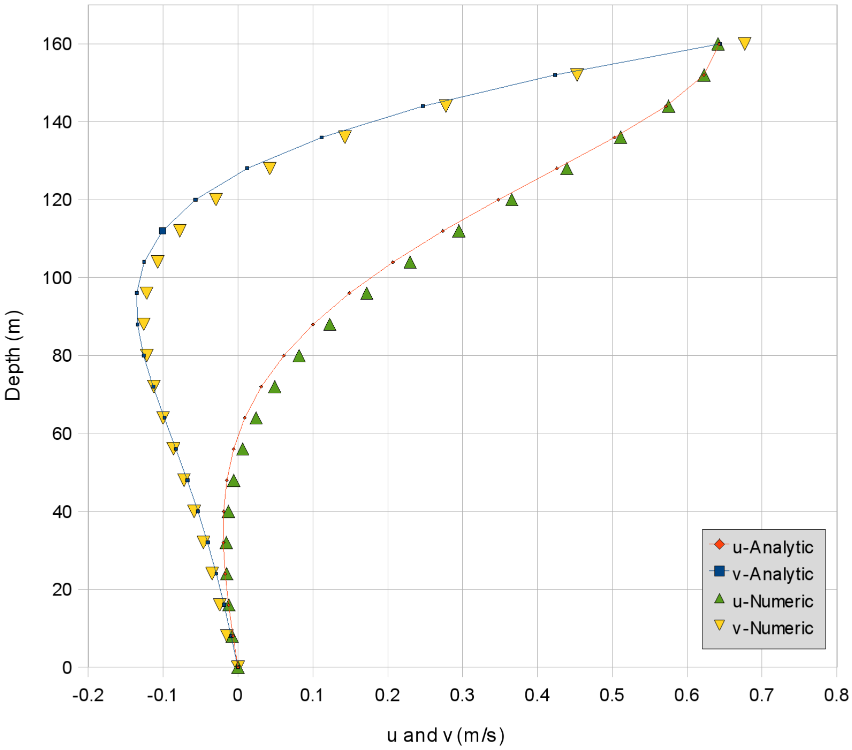


Figure 8.10. Comparison of the numerical and analytical Ekman profile

depth = 160m, wind speed = 32 m/s, eddy viscosity = 0.1 m^2/s, latitude = 50°

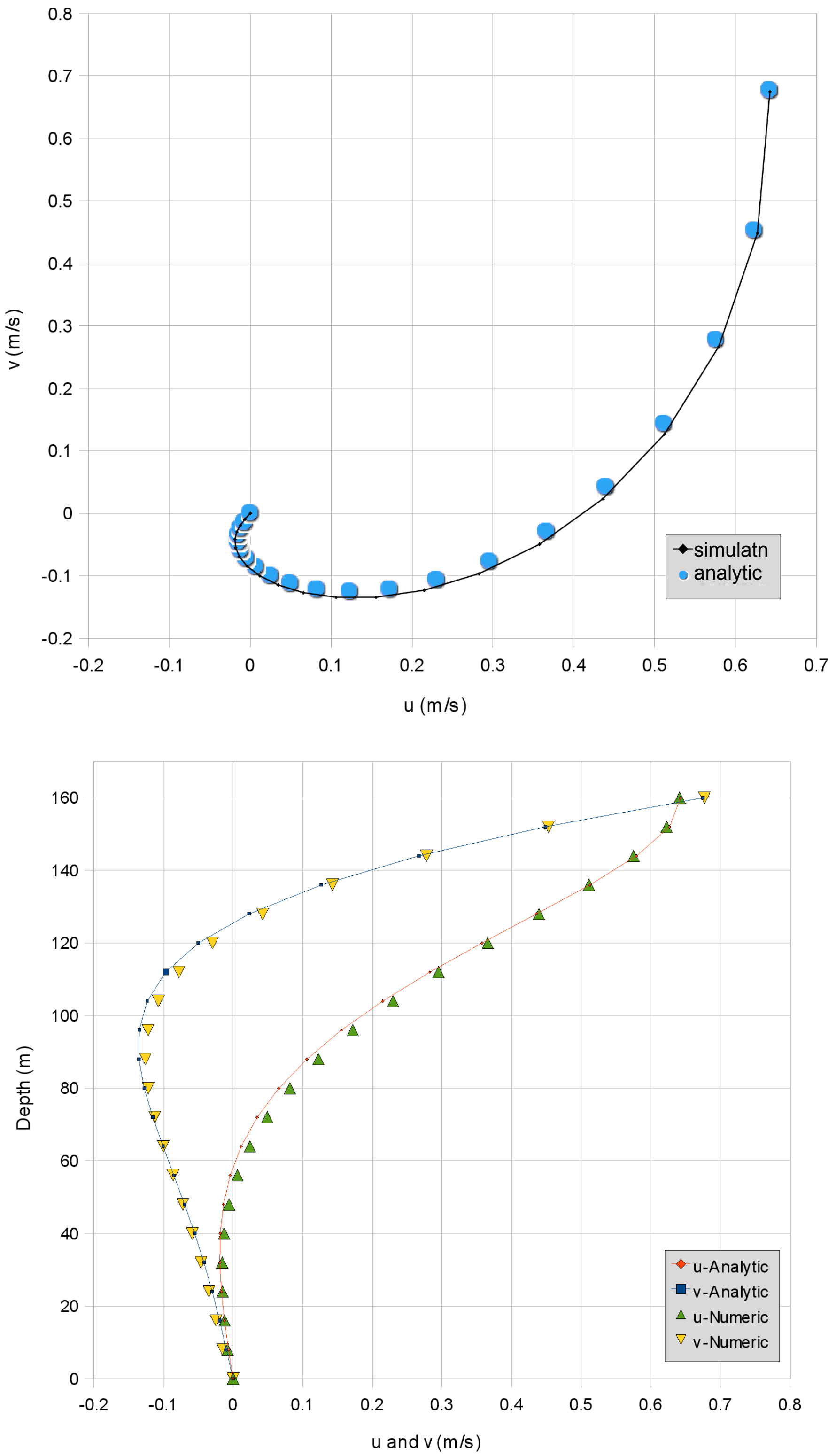


Figure 8.11. Sensitivity analysis of the Ekman profile due to the water depth depth in numerical simulation = 160 m, depth in analytical solution = 159 (compare with figure 8.10)

8.2 Hydrodynamic Pressure - Wind Driven Circulation

A typical example that illustrates the effect of non-hydrostatic pressure effect is the wind-driven circulation in a closed basin (Huang, 1993; Tsuruya et al., 1985; Jankowski, 1999; Kocyigit M., 2002). A square basin measuring 10m by 10m with 5m depth has been considered here. The water body is influenced by wind shear stress due to the wind velocity in the x-direction.

The problem has been simulated for wind speed of 5 m/s. The vertical eddy viscosity is 0.001 m^2/s. The horizontal eddy viscosity is zero in one case and 0.001 m^2/s in another case. The computational mesh is similar to figure (8.1). However the dimensions are different. There are 21 computational nodes along the vertical direction which have distributed evenly. The vertical resolution is 0.25m. The Coriolis force is neglected. The bed condition is no-slip and the wall condition is free-slip. The simulation time step is 1s.

The simulation was carried out for 1000 seconds at first, but it was not steady state. Then, the non-hydrostatic simulation was repeated for 4000 seconds where it was close to the steady state condition. Figures (8.12) to (8.15) illustrate the simulation results after 1000 seconds, in order to compare the hydrostatic and non-hydrostatic simulations. Figure (8.16) illustrates the simulation result after 4000 seconds for comparison with the analytical steady state solution.

Figure (8.12) compares the results of hydrostatic and hydrodynamic pressure assumptions for the case of 0.001 m^2/s horizontal eddy viscosity. Figure (8.13) compares the results of hydrostatic and hydrodynamic cases for zero horizontal eddy viscosity. It is clear from the figures that the near-wall vertical velocity in the hydrostatic pressure case is considerably more than the hydrodynamic pressure case. The non-hydrostatic solution results in a smoother circulation over the entire area. Increasing the horizontal eddy viscosity helps for reducing the vertical velocity near the wall.

The underwater circulation in the hydrostatic case is due to the free surface slope which results in hydrostatic pressure gradient along the horizontal layers. However, the underwater circulation in the hydrodynamic pressure case is partly due to the hydrostatic pressure gradient and partly due to the hydrodynamic pressure gradients in the horizontal and vertical directions. Figure (8.14) illustrates the effective pressure and velocity field. The effective pressure is the sum of the hydrodynamic pressure and the hydrostatic excess pressure. The hydrostatic excess pressure is the difference in hydrostatic pressure at each point relevant to the still water condition. It is clear from the figure that there is no velocity gradient when there is no effective pressure gradient. The maximum hydrodynamic pressure occurs at the upper corners. The near-wall velocity in the non-hydrostatic solution is completely due to the hydrodynamic pressure gradient. The vertical velocity component in the hydrostatic solution is obtained from the continuity equation only.

Figure (8.16) illustrates the predicted vertical distribution of the horizontal velocity at the middle of the basin after 4000 seconds. The continuous line in the figure shows the analytic solution after (Huang, 1993). for comparison with the analytical steady state solution. The analytic solution for a constant vertical eddy viscosity is given as (Huang 1993, Kocyigit 2002)

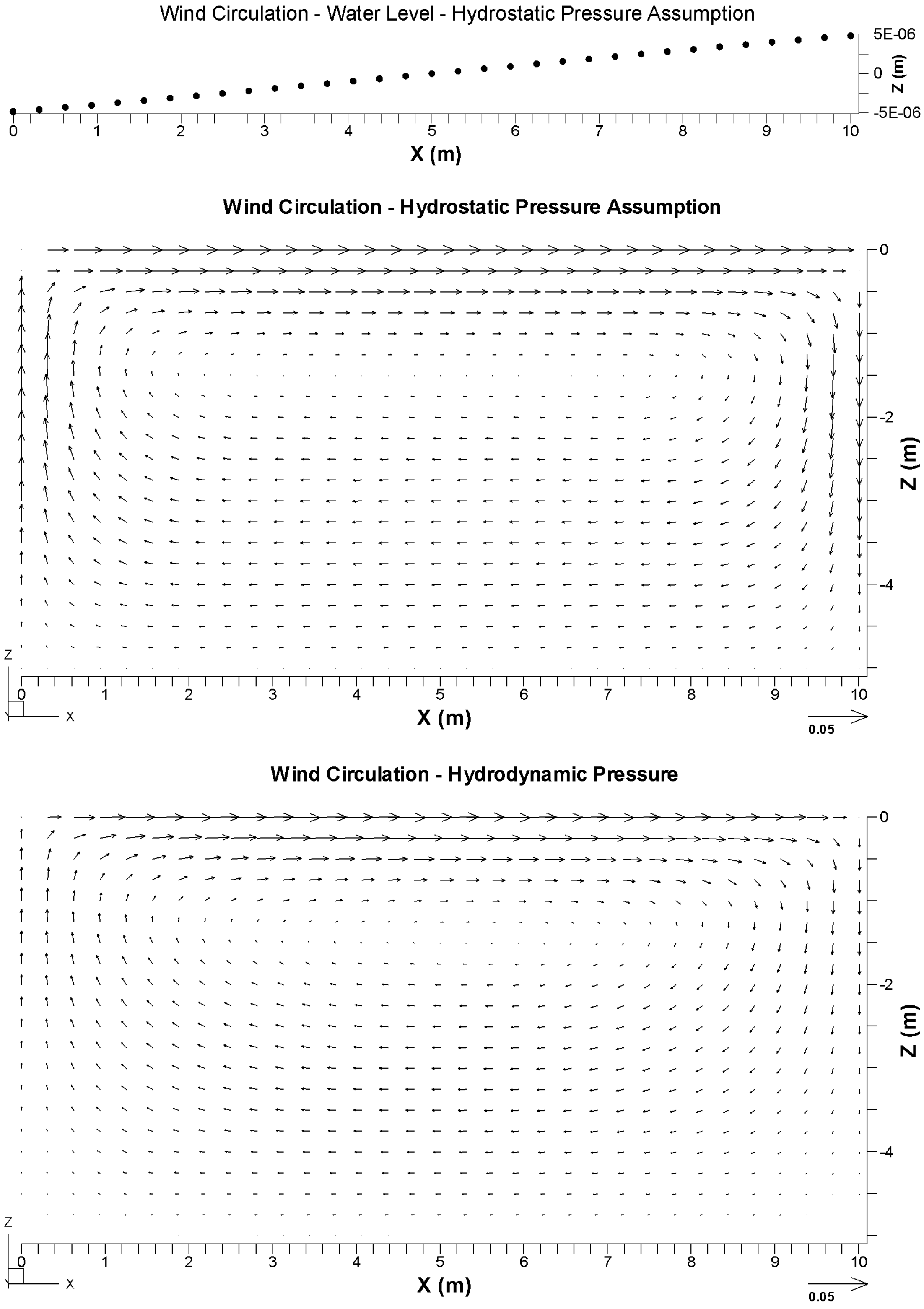
$$u=\frac{1}{6\nu_v}g\frac{\partial\eta}{\partial x}\left(3z^2-H^2\right)+\frac{\tau_w}{2\rho\nu_v}\left(H+2\mathrm{z}\right) \qquad (8.9)$$

where

$$\frac{\partial\eta}{\partial x}=\frac{3}{2}\frac{\tau_w}{\rho g H}\left(\frac{2\nu_v+K_l H}{3\nu_v+K_l H}\right) \qquad (8.10)$$

and k_l is the linearized bottom friction coefficient.

Figure (8.16) shows that the predicted results are in good agreement with the analytic solution. However, the predicted velocity is not yet completely steady state and the analytic solution can not takes the left and right boundaries into account. The velocity error at the water surface is about 2%.



Wind Circulation - Water Level - Hydrostatic Pressure Assumption

Wind Circulation - Hydrostatic Pressure Assumption

Wind Circulation - Hydrodynamic Pressure

Figure 8.12. Wind driven circulation in a 10x10x5m pool after 1000 seconds.

(a) water level for the hydrostatic pressure case, (b) velocity pattern for the hydrostatic pressure case,

(c) velocity pattern for the non-hydrostatic case. v_v= 0.001 m^2/s, v_h= 0.001 m^2/s.

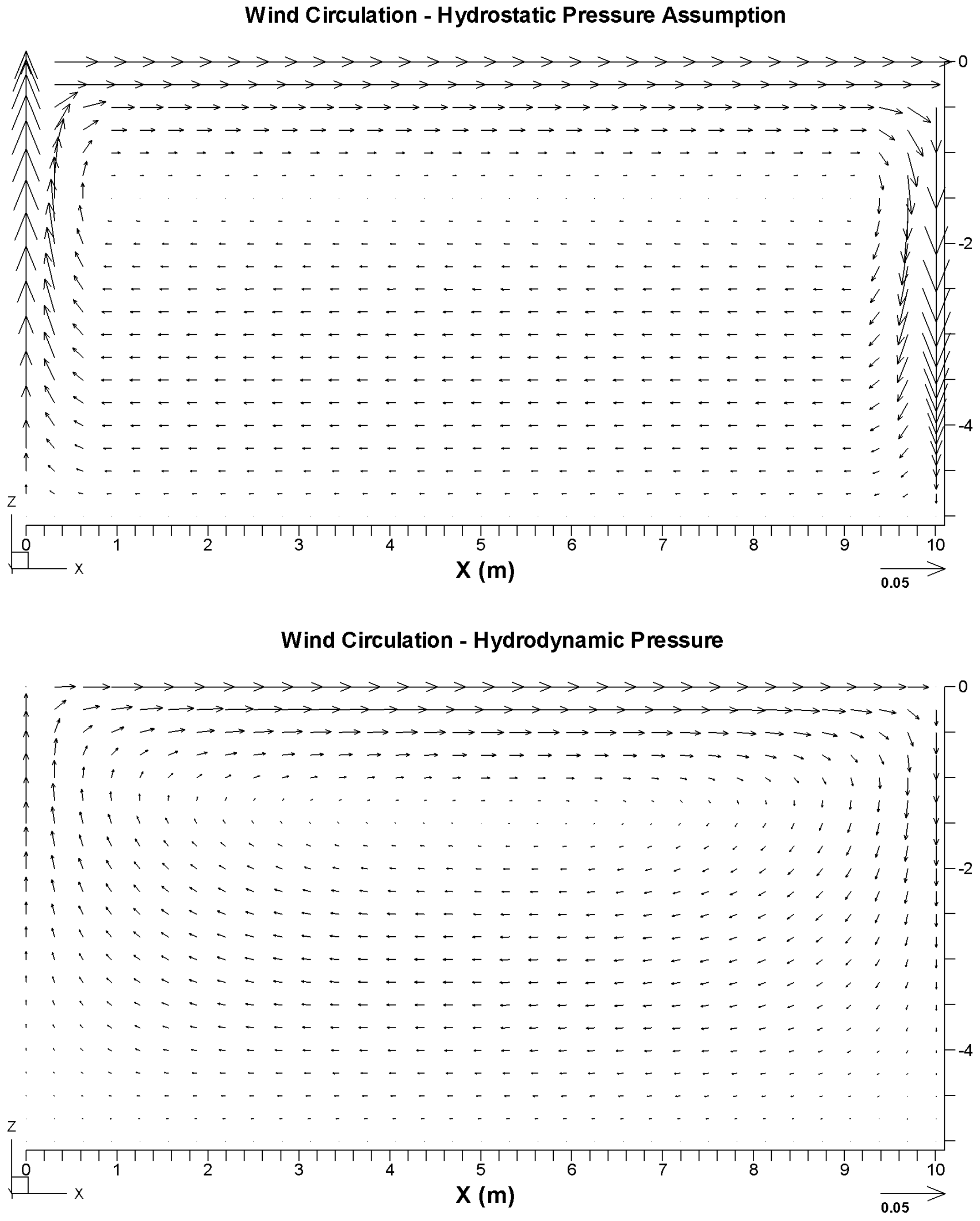


Figure 8.13. Wind driven circulation in a 10x10x5m pool after 1000 seconds.
(a) velocity pattern for the hydrostatic pressure case,
(b) velocity pattern for the non-hydrostatic case. v_v= 0.001 m^2/s, v_h= 0.0 m^2/s.

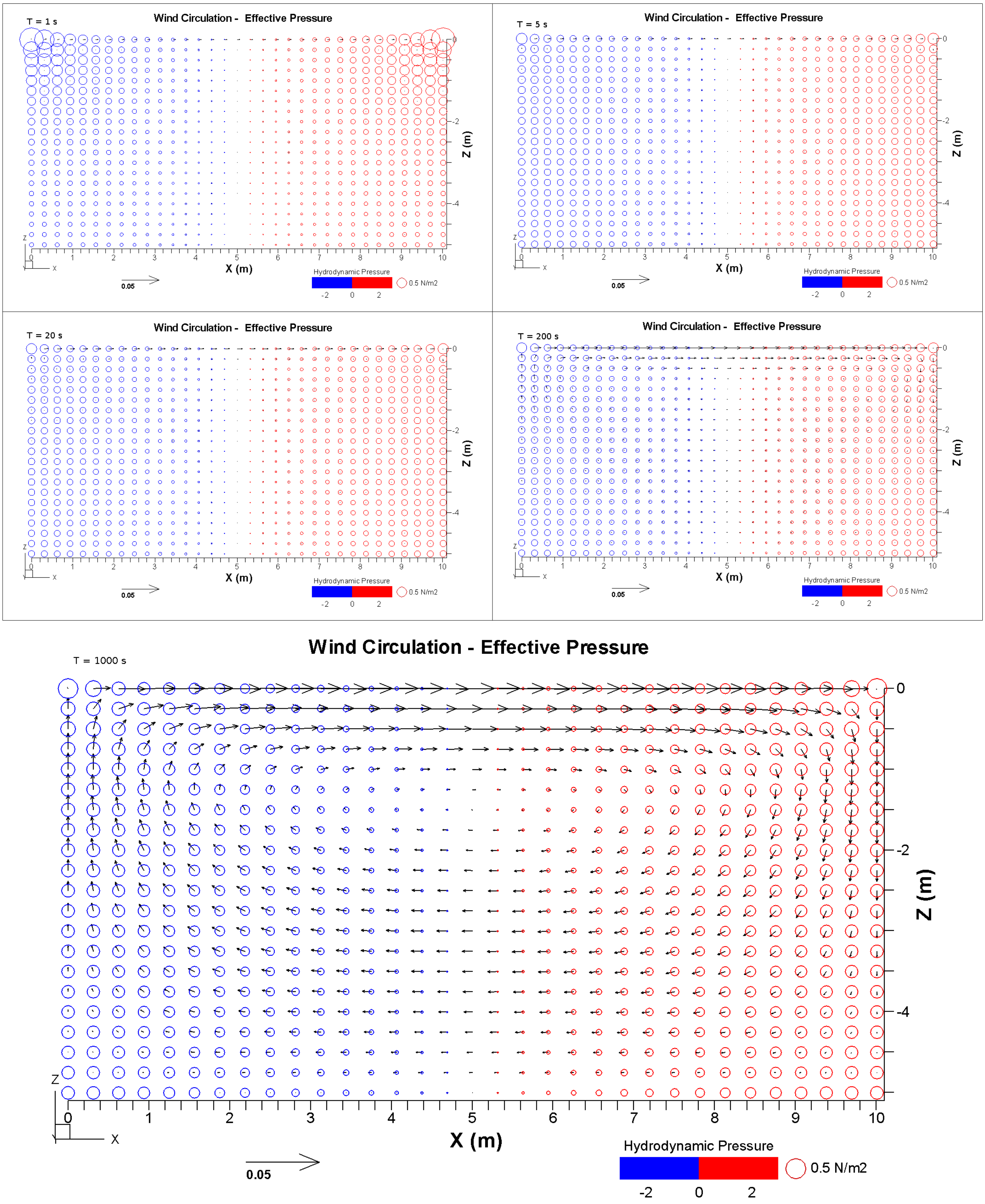


Figure 8.14. Effective pressure and velocity filed relevant to the wind driven circulation

Effective pressure = Hydrodynamic pressure + Hydrostatic excess pressure.

Hydrostatic excess pressure is the difference in hydrostatic pressure relevant to the still water condition.

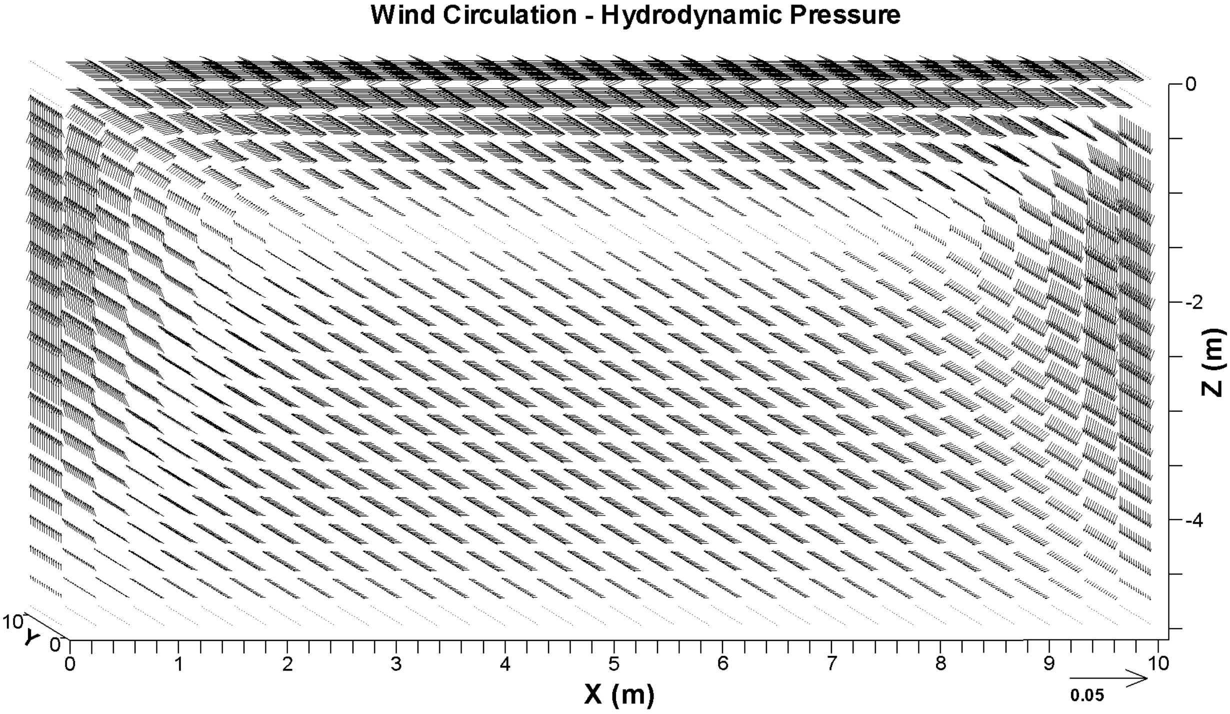


Figure 8.15. A 3D view of the wind driven circulation (v_h= 0.001 m^2/s).

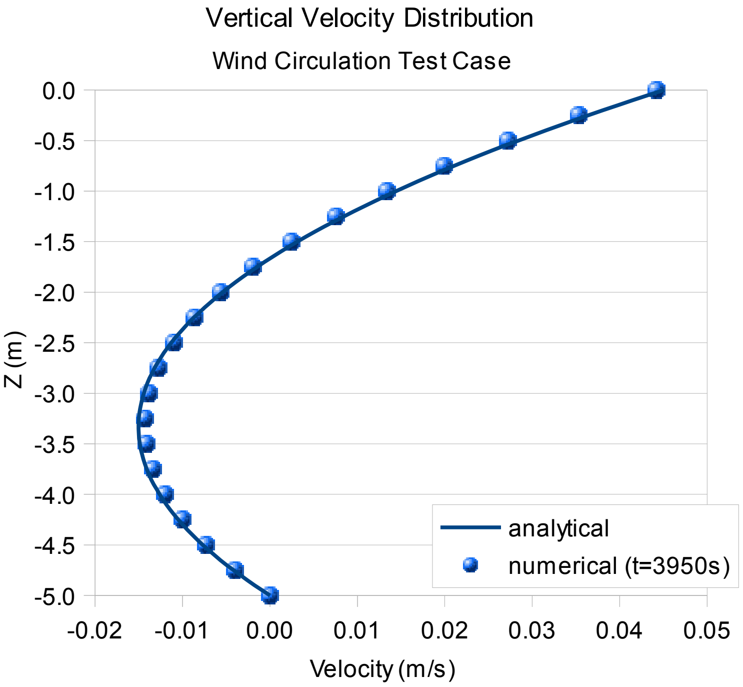


Figure 8.16. Comparison of the predicted velocity distribution at the middle of basin with the analytical solution of (Huang, 1993) for the non-hydrostatic case.

8.3 Lock exchange flow

The lock exchange flow is a well-known test case to verify the modelling of density-buoyancy current (Benjamin, T., 1968; Turner, J., 1973; Yih, C.-S., 1980; Huppert and Simpson, 1980; Jankowski, J. A., 1999; Namin, 2003). In this case, two incompressible free surface fluids of slightly different densities and equal depth are initially divided in a rectangular basin by a thin wall in the middle of the basin. When the wall is removed, a buoyancy driven flow is developed. The denser fluid flows towards the other side along the bed, while the lighter fluid flows in the opposite direction along the water surface.

The shape of the simulated interface between the fluids depends on considering or ignoring the hydrodynamic pressure component. The hydrostatic pressure assumption results in a rectangular-type pattern which is not observed in nature or laboratory. The noses of density currents advancing along a flat bottom have a slope of approximately $\pi/3$ to zero (Turner J. 1973). This kind of interface shape is reproduced by simulation when the hydrodynamic pressure component is taken into account.

The original driving force for the denser fluid is the horizontal pressure gradient due to heavier water column in the denser part. With hydrostatic pressure assumption in mind, the driving force for the lighter fluid is just the pressure gradient due to the water surface gradient that happens after dense fluid flow. But in fact the hydrodynamic pressure component has considerable role in driving the fluid and shaping the velocity field.

It is possible to find analytically the mean velocity of the fluids movement during the transient time in a simplified condition where there is no viscosity and energy loss. Equation (8.11) illustrates the result of such a solution (for details, see Jankowski J. A. 1999).

$$u=\frac{\sqrt{2}}{2}\sqrt{\frac{\rho_2-\rho_1}{\rho_2+\rho_1}gH}=0.71\sqrt{\frac{\rho_2-\rho_1}{\rho_2+\rho_1}gH}\approx 0.5\sqrt{\frac{\rho_2-\rho_1}{\rho_2}gH}=0.5\sqrt{g'H} \qquad (8.11)$$

where $g'=g\cdot\Delta\rho/\rho_2$ is the reduced gravity acceleration.

The laboratory experiments of Yih (Yih, 1980) yield the flow velocity as given by the equation (8.11) with the coefficient being 0.67 instead of 0.71. Barr observed the factor of 0.44 in a closed conduit. He observed slightly asymmetrical flow with factors of 0.47 and 0.59 for the underflow and overflow respectively (Jankowski, 1999).

The lock exchange flow case has been simulated here using the present model. The basin is 30m long and 3m wide. The water depth is 4m. The water temperature is 15°C and the initial salinity at the left hand side and right hand sides are 11 psu and 10 psu, respectively. Therefore the initial water density is 1007.55 kg/m3 at the left hand side and 1006.78 kg/m3 at the right hand side. The vertical and horizontal eddy viscosities are set to zero, in order to compare the results with the analytical case.

Figure (8.17) illustrates a part of the computational mesh for this test. The distance between the nodes is 0.35m in the central area. The number of computational nodes along each column is 21. The simulation time step is set to 1s.

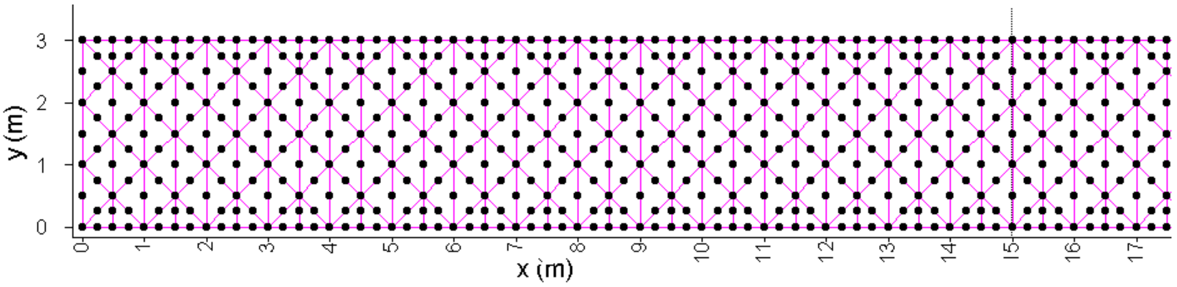


Figure 8.17. Plan view of the computational grid for the lock-exchange test case.

Figure (8.18) illustrates some results at selected times during the simulation when the hydrodynamic pressure component is taken into account. It shows that the smooth penetration of the salty water below the fresh water and also the smooth penetration of the fresh water above the salty water. The frontal angle of the salt-fresh water is from about $\pi/3$ to zero, similar to the physical modelling results (Turner J. 1973).

Figure (8.19) compares the behaviour of the hydrostatic and hydrodynamic pressure assumptions. It shows that when the pressure is assumed to be hydrostatic, the vertical velocities at the fronts are unrealistically high, the horizontal velocities are under-estimated, and the general shape of the salt-fresh boundary is wrong. The reason is that with this assumption the horizontal velocity components are forced just by the baroclinic and hydrostatic pressure gradients, the vertical velocity component is defined just by the continuity equation, and the effect of the hydrodynamic pressure gradient has been ignored both on the horizontal and vertical velocity components.

The present model has the option of switching each physical process on or off separately. It is interesting to see the effect of pure advection on the results of the lock exchange problem.

Figure (8.20) compares the result of the lock exchange problem after 100 seconds when advection is ignored, when the horizontal advection is solved but the vertical advection is ignored, and finally when the advection process is solved completely. The figure shows that the horizontal advection has considerable effect on the velocity of the salt-fresh water penetration. The vertical velocity has considerable effect on the height of the front and its shape at the centre-line.

Figure (8.20) shows the simulation result of a non-hydrostatic version of Telemac3D for the same lock exchange problem after 100 seconds (Jankowski, 1999). The initial salinities in the Jankowski's test case are 0 and 1 instead of 10 and 11, but the initial density gradient is the same. Comparing figures (8.17) and (8.20) shows that there is a good agreement between the two simulations. The salt water front movement after 100 seconds is predicted consistently in the two models. Comparing the vertical distribution of the salinity at the mid-side shows that the numerical diffusion in the present model is less than that version of Telemac3D.

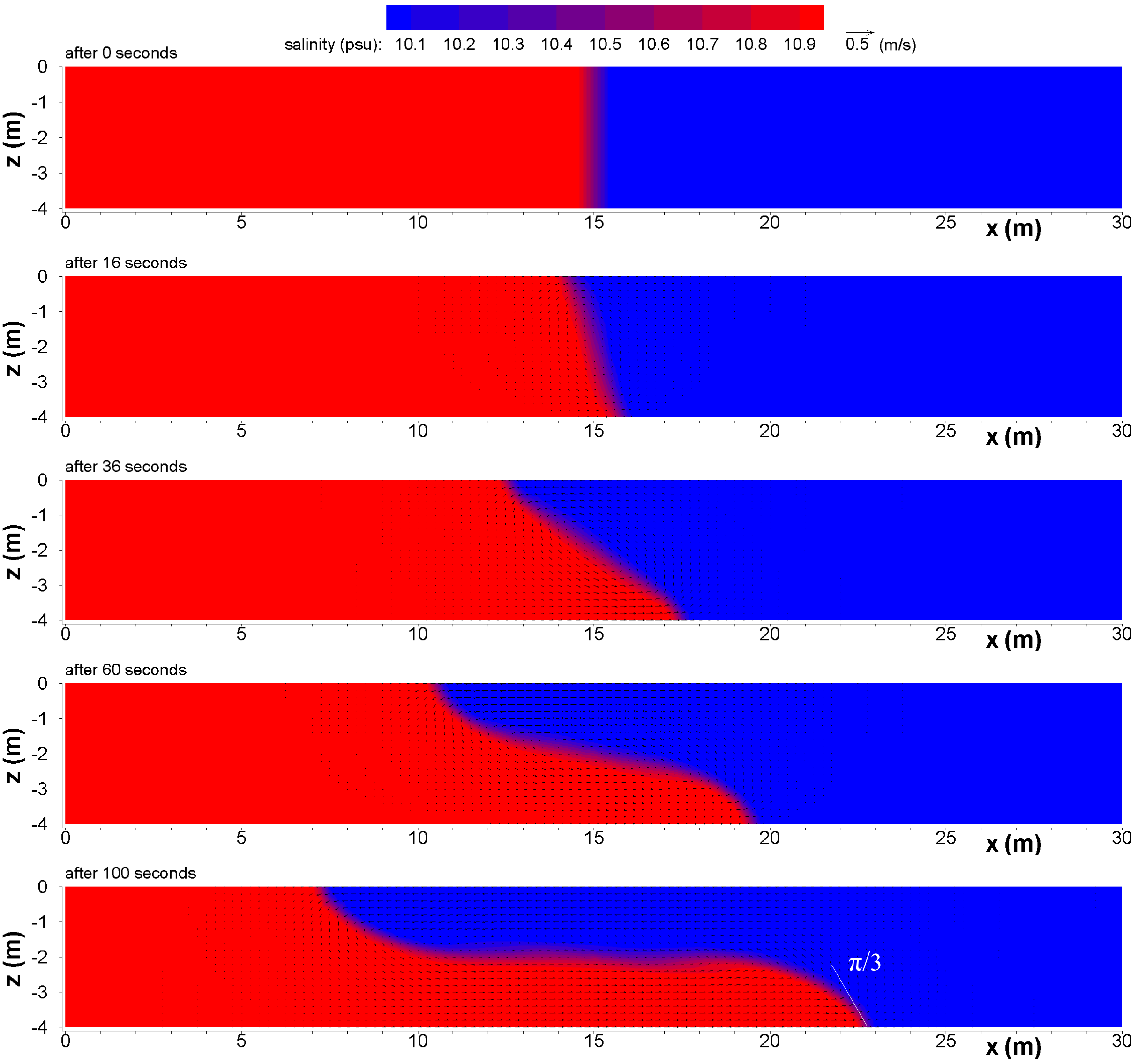


Figure 8.18. Selected simulation results from the hydrodynamic pressure simulation.

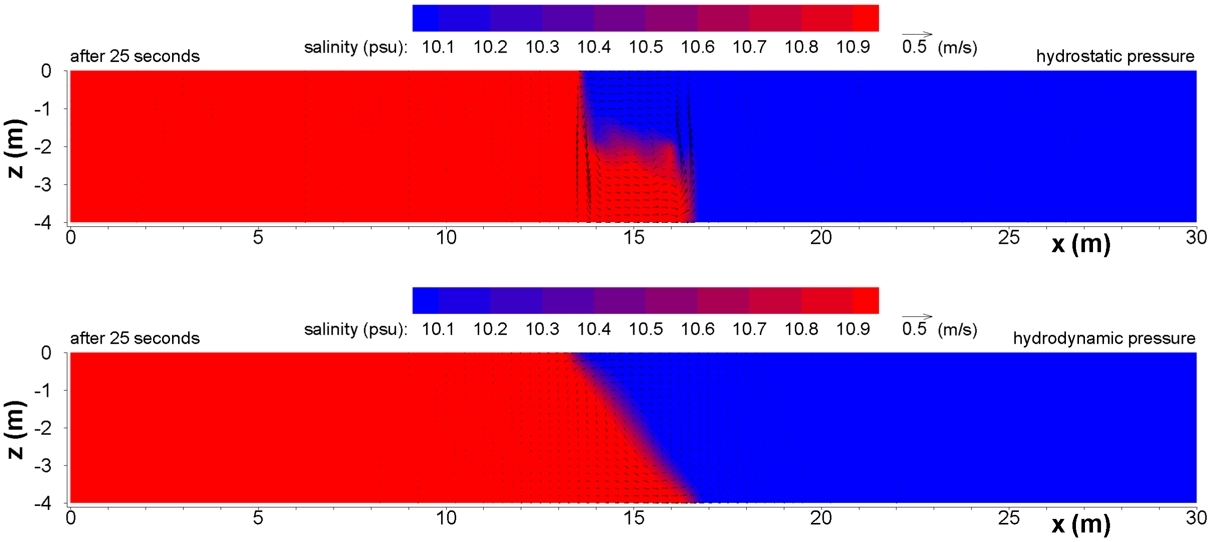


Figure 8.19. Difference between the hydrostatic and hydrodynamic pressure simulations.

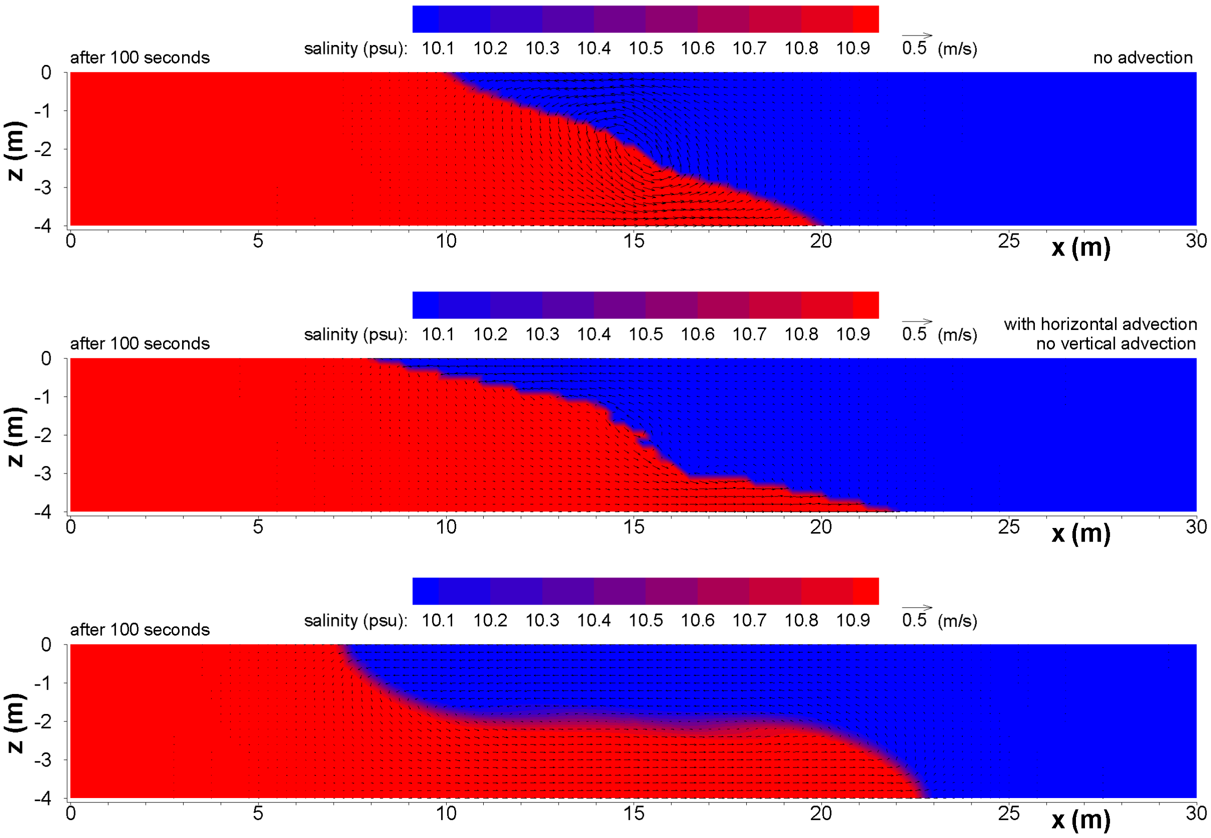


Figure 8.20. The effect of horizontal and vertical advection on the lock exchange problem.

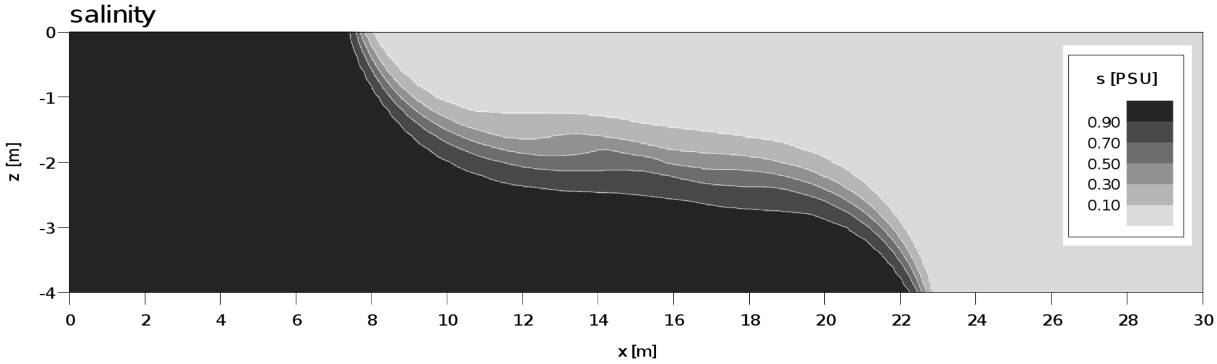


Figure 8.21. The simulation result of a non-hydrostatic version of Telemac3D for the same lock exchange problem after 100 seconds (Jankowski, 1999). (The salinities in this simulation are 0 and 1 instead of 10 and 11, but the initial density gradient is the same)

8.4 Basin Wave Movement by Integrated 2D-3D Simulation

The following test case shows the smooth integration and implicit dynamic data exchange between the 2DH and the 3D models within a basin. A semi-enclosed basin with the length of 200 meters and width of 100 meters is considered, with the initial water depth equal to 20 meters. The left side of the basin is simulated by the 2DH model while the right sidez is modelled by the 3D one. The two models are connected to each other at the middle of the basin. Figure (8.22) illustrates the basin and the distinct computational grids of the 2D and 3D domains. The left boundary of the basin (located in the 2DH domain) is open from which water enters into the basin with a constant rate of 200 m3/s uniformly along the boundary line. Other boundaries of the basin are closed. The computational time step in this simulation is 0.5 seconds with Courant number more than 2.0. The computational grids of the two domains are independent. Figure (8.23) illustrates the numbering systems of the grids. There is not any global numbering system for the computational nodes.

Figures (8.24) to (8.28) illustrate selected results of the simulation. Figure (8.24) shows the simulated water-level change in the test basin. It shows that the incoming wave moves smoothly towards the right side up to the end of the basin. Then a backwater wave is generated at the right-end of the basin and moves back up to the input boundary, where the second input wave is generated on top of the new water level. The movement of the the second input wave and backwater is illustrated too. It is clear that no partial reflection occurs at the interface. Figures (8.25) to (8.27) illustrates the simulated velocity field within the test basin (upper segment: depth averaged horizontal velocity, below: local velocity). It shows that the velocities are identical at the interface during the simulation period. Figure (8.28) illustrates the instant when the input wave crosses the interface in more details. There is no numerical partial reflection or discontinuity in the free surface slope or local oscillations when the input waves and the backwater waves cross the 2D-3D domains interface. This result is achieved without a global computational grid that solves all or part of the governing equations on the whole basin. The integration process does not restricts the simulation time step and does not enforce any iterative computation to the simulation procedure. There is no predicted velocity at the right side of the backwater wave, which is consistent with the analytical solution of such a backwater. This consistency shows that the conservations lows for mass and momentum have been fulfilled with the same accuracy as simulating on a single mesh with no multi dimensional coupling. It also signs that no numerical partial reflection has happened at the interface.

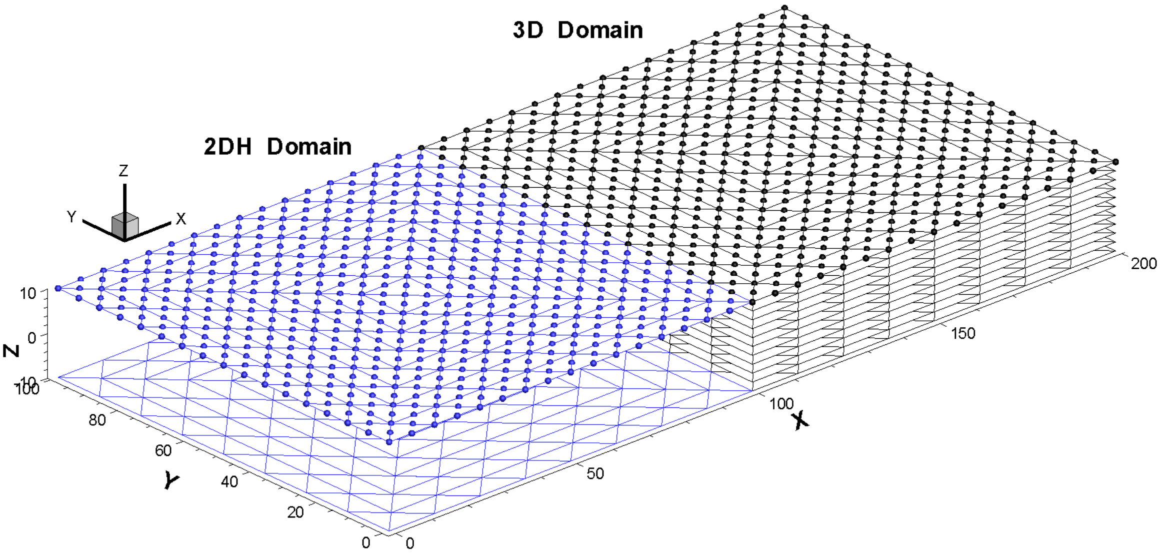


Figure 8.22. The basin with the distinct computational grids for the 2D and 3D domain parts.

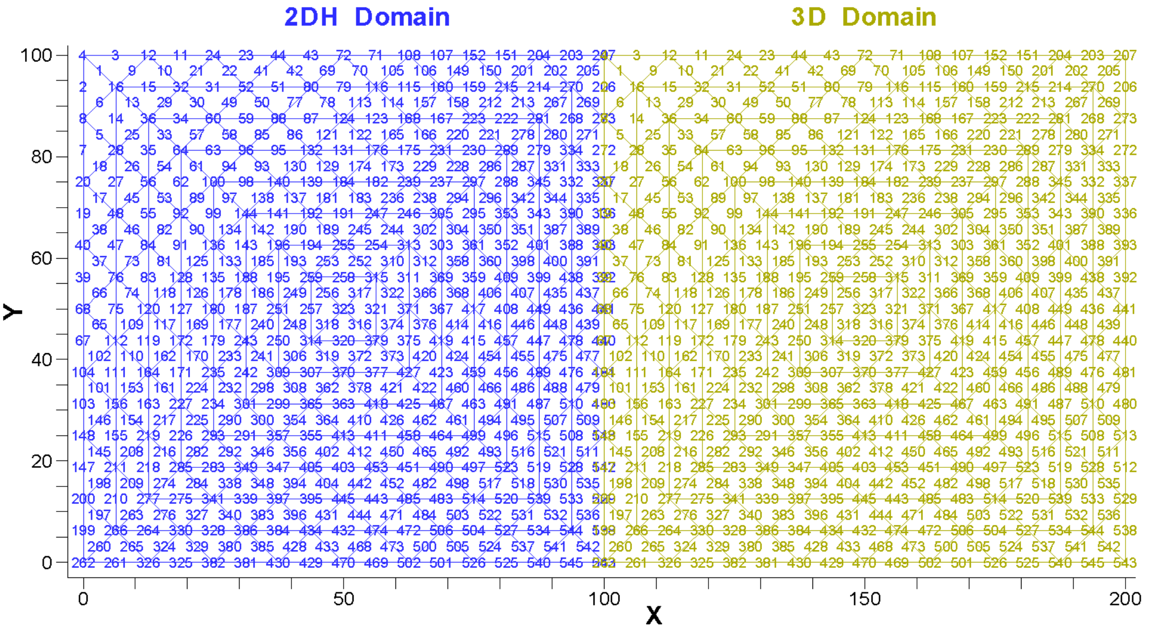


Figure 8.23. The independent numbering system of the computational grids.

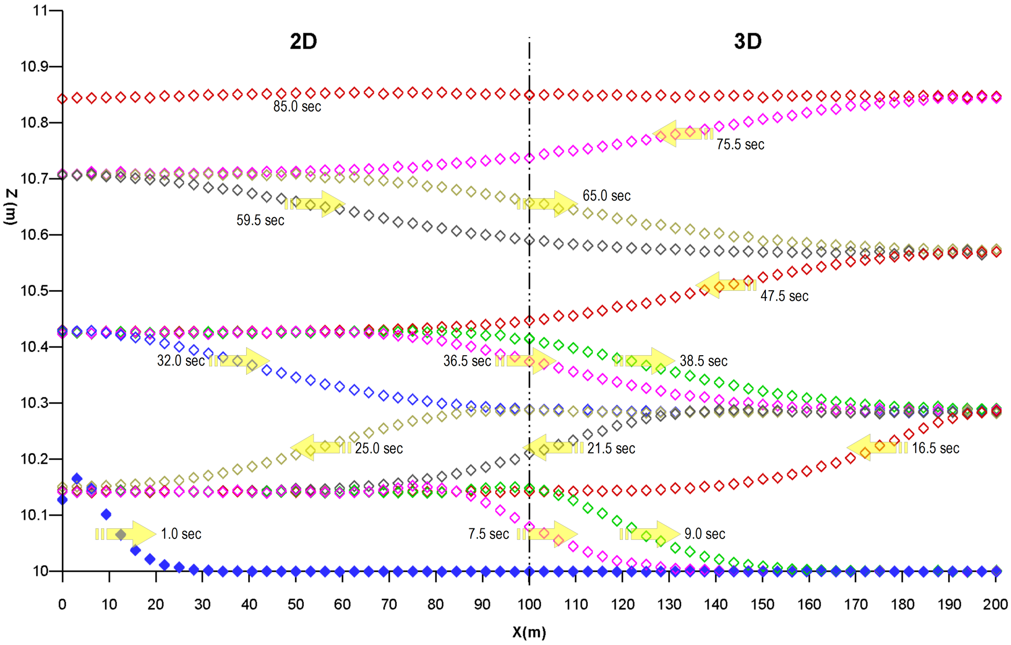
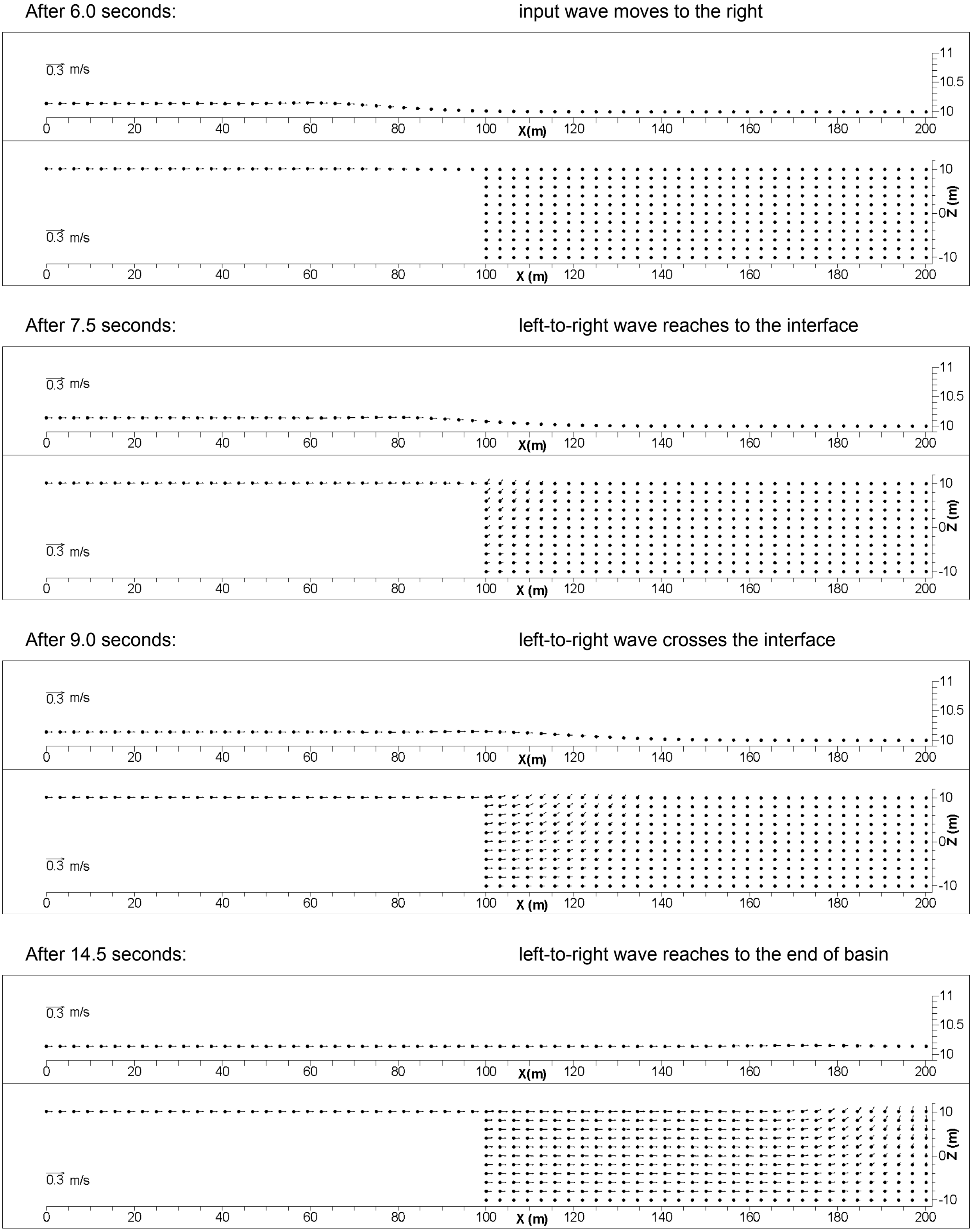


Figure 8.24. Water-level change in the test basin, simulated by 2D-3D integration.



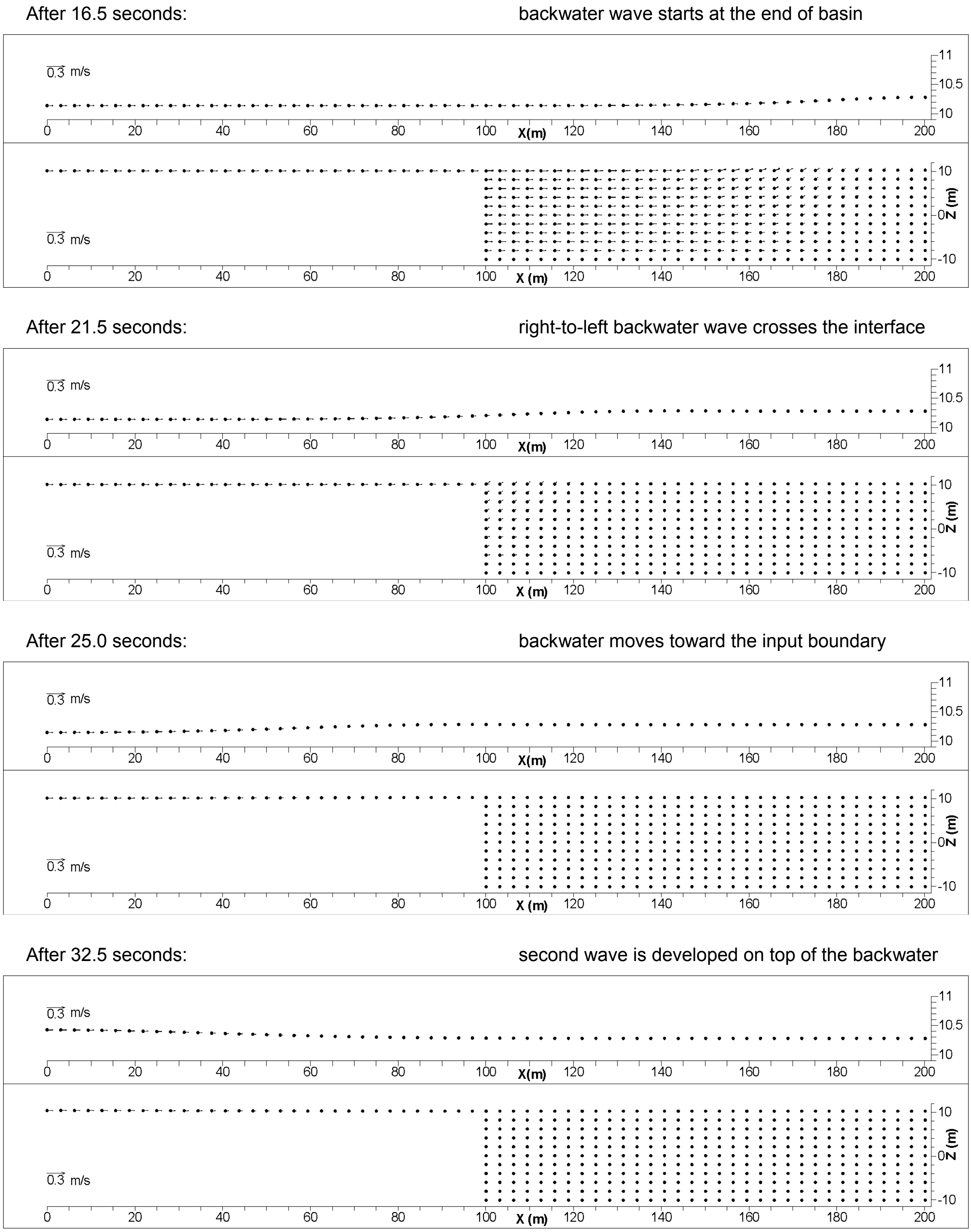
After 6.0 seconds: input wave moves to the right

After 7.5 seconds: left-to-right wave reaches to the interface

After 9.0 seconds: left-to-right wave crosses the interface

After 14.5 seconds: left-to-right wave reaches to the end of basin

Figure 8.25. Velocity field within the test basin simulated by 2D-3D integration.
(upper segment: depth averaged horizontal velocity, below: local velocity)



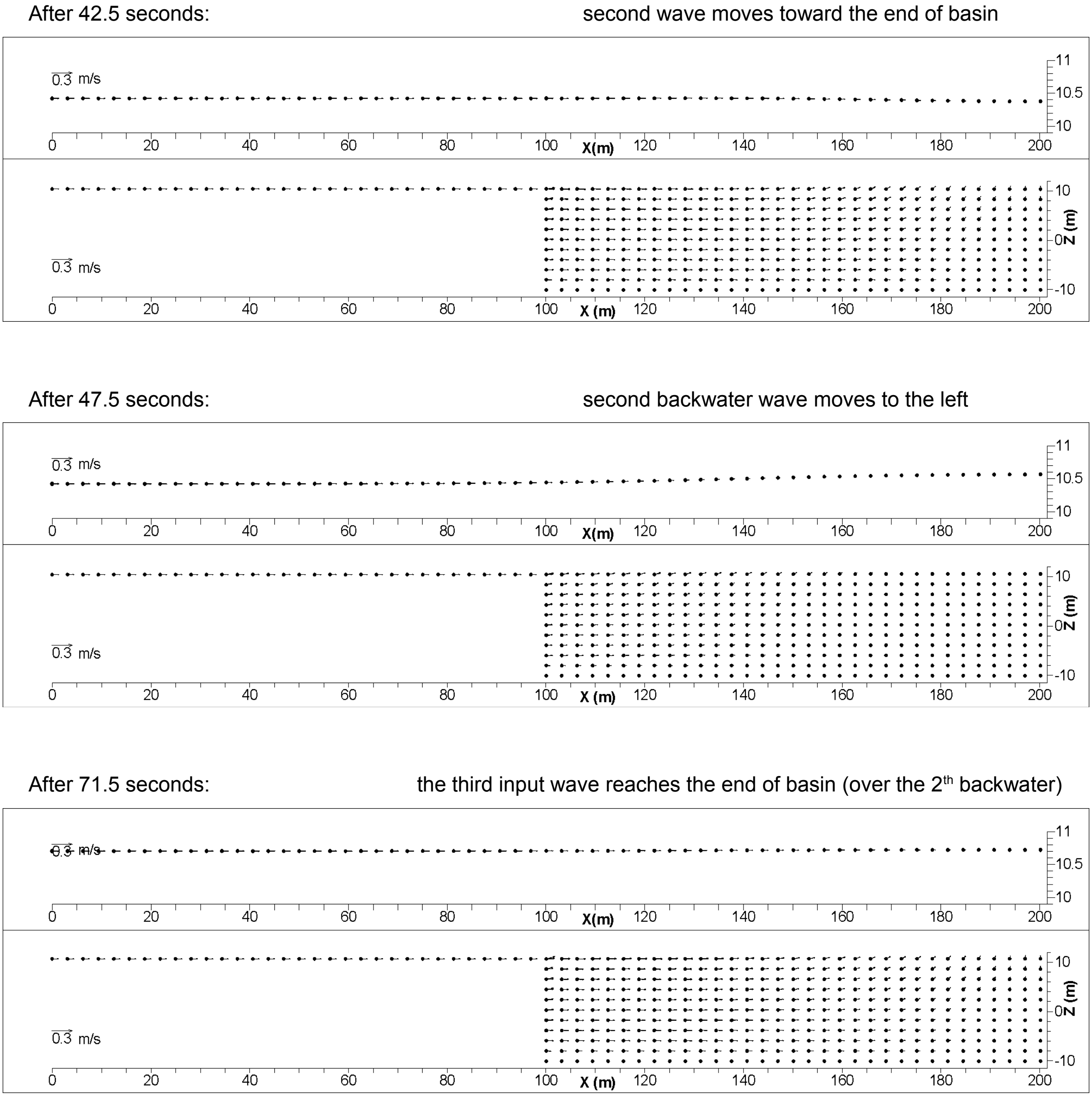
After 16.5 seconds: backwater wave starts at the end of basin

After 21.5 seconds: right-to-left backwater wave crosses the interface

After 25.0 seconds: backwater moves toward the input boundary

After 32.5 seconds: second wave is developed on top of the backwater

Figure 8.26. Velocity field within the test basin simulated by 2D-3D integration (continue). (upper segment: depth averaged horizontal velocity, below: local velocity)



After 42.5 seconds: second wave moves toward the end of basin

After 47.5 seconds: second backwater wave moves to the left

After 71.5 seconds: the third input wave reaches the end of basin (over the 2th backwater)

Figure 8.27. Velocity field within the test basin simulated by 2D-3D integration (continue).
(upper segment: depth averaged horizontal velocity, below: local velocity)

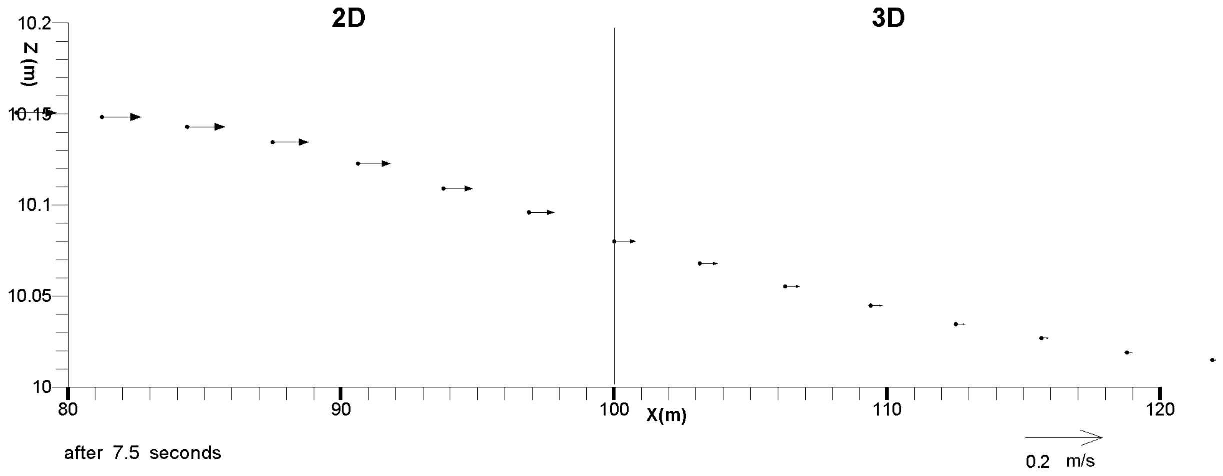


Figure 8.28. Predicted input wave crossing the interface with continuous free surface slope without domain-overlapping or iteration.

8.5 Conclusion

In this chapter, samples of the verification of the 3D model and the integration against analytical data are given. The hydrodynamic pressure, density current, diffusion effect, Coriolis force and 2D-3D flow integration have been examined.

In the first part of the chapter, ...A comparison of the model predictions with analytical solutions showed very good agreement.

The numerical model was then applied to study the wind induced current circulation in a closed basin, where an analytical solution is available for the vertical velocity distribution. It was shown that the model predictions were again in good agreement with the analytical solution.

Then the model capability was checked for simulating the density currents in conjunction with the hydrodynamic pressure. The results compared and were in good agreement with the available data and results from another world-class sophisticated model, while the numerical diffusion in the present model was less than the compared one.

Finally the capability of the model for implicit coupling of 2D-3D domains was checked throughout simulating the water wave and backwater wave motion in a semi-enclosed basin. The results showed that no numerical reflection or local oscillation has been produced at the interface and the velocity behind the backwater wave is always zero (as expected), which signs the accurate mass and momentum balance within the domains and at the interface.

9 Discussion

In this research program, an integrated horizontally two dimensional and fully three dimensional model system has been developed based on a combined unstructured and σ-coordinate grid to simulate the flow and water quality process in large water bodies with a focus on the three dimensional behaviours at specific areas. Each model is robust and efficient and can be used independently. Nevertheless, the two models are optimised for coupling to each other and give a robust and efficient integration procedure. The two models interact dynamically during the solution procedure with no additional time-step restriction or iteration introduced due to the integration algorithm.

Both of the models have been developed from scratch, in order to fulfil the proposed framework completely. The programming language C++ has been used because of its flexibility in object-oriented programming, flexibility in mixing the object-oriented program with procedural code segments of Fortran and C in order to gain more efficiency, and its power in memory management.

The models are based on the time dependent Reynolds-Averaged Navier-Stokes equations with a non-hydrostatic pressure distribution and a baroclinic force being incorporated in the three dimensional (3D) model. The horizontal gradients of the water density and shear stress are computed on the true horizontal plains in order to prevent inaccuracy in density flow simulation when the slope of near-bed sigma layers varies significantly.

A novel coupling algorithm has been introduced in order to overcome some known restrictions of the previous coupling algorithms (see example see Wu W., 2007; Sebastian and Shu, 2003). There are not many integrated 2DH-3D models available in the literature. However, existing ones use the interpolation/averaging algorithm at the interface in order to exchange data back and forth between the models. This procedure does not guarantee the conservation low at the interface and may cause inaccuracy when the spatial gradients of the flow parameters are not gentle. In the present model, an interpolation/averaging algorithm at the interface is not required due to unstructured computational grid and introducing hybrid finite volume cells. The hybrid finite volume cells are partly in the 2D domain and partly in the 3D domain area.

The present model uses the fractional step algorithm in both of the models and solves the common fractions of the two sub models together with common algorithms. Dynamic data exchange between the models are handled fraction by fraction. Each fraction concerns one or more physical processes and is solved by its most efficient algorithm. Most of the fractions are solved throughout the cell-vertex finite volume method except for advection which is solved by the semi-Lagrangian method. The 2D and 3D models are coupled dynamically through applying the hybrid cells to the computational nodes on the interface. A hybrid finite volume cell produces a unique discretised equation for a computational node, while the discretised equation is partly supported by nodes within the 2DH domain and partly by the 3D domain.

The 3D model uses the projection method for pressure calculation. The projection method is a specific type of the fractional step algorithm and is consistent with the overall solution strategy.

The solution fractions of the 3D flow model are: the advection, horizontal diffusion, free surface, vertical diffusion, and projection steps respectively. The solution fractions of the 2DH flow model are: the advection, horizontal diffusion and free surface steps. The common fractions of the two models are solved together with similar procedures to enable data to be exchanged directly.

The advection step is solved by the semi-Lagrangian method. It is unconditionally stable, which is an important requirement for advection dominated problems. The method is also efficient and simple for predicting the advection effect at the interface, and id unconditionally stable.

The horizontal diffusion equation is solved explicitly via the finite volume (FV) method. An explicit solution of the horizontal diffusion does not restrict the model's overall solution time step. The computational nodes on the interface are handled by the hybrid FV cells. In this way, data on a link node is predicted by nodal data from both 2D and 3D domains. This algorithm enables the smooth transmission of the diffusion process at the interface.

The horizontal gradients of the water density and shear stresses are calculated on the true horizontal planes, instead of transforming their terms into the σ-coordinate space. The aim is to avoid artificial velocity and diffusion that could form near the steep bed slope in highly stratified flow regions, such as density current in a reservoir. The present study has used a new approach for discretisating of the horizontal diffusion terms. It computes the shear stresses in the Cartesian coordinate system, and then computes the gradients of the shear stresses in the σ-coordinate system.

The free surface slope has a major role in the flow field in most of the practical flow problems. Determining the free surface needs special attention when two models are coupled, particularly when an implicit procedure is used. The present study has introduced a new implicit algorithm, in order to guaranty the mass and momentum conservation at the interface of the unstructured grids. The discretisation of the depth integrated forms of the continuity and momentum equations in the 2D and 3D domains results in two distinctive matrix equations. However, each matrix (and right hand side vector) involves just half of the data regarding to the hybrid cells on the interface. Then the matrices are linked at the link nodes in order to be solved as a global matrix equation. There is no need to have a global computational grid or a global numbering system, of the nodes in this regard.

The vertical diffusion sub-model is solved implicitly, column by column, using a one dimensional finite volume method. This sub-model also involves the barotropic and Coriolis terms. The Cartesian coordinate system is used herein instead of the sigma coordinate system. In this way, not only the velocity but also its vertical gradient remains continuous at the element boundaries. At the lateral free-slip closed boundaries, the vertical diffusion equation is solved for the tangential direction.

The pressure correction equation is solved in the Cartesian coordinate system using the three dimensional finite volume method. The bi-conjugate gradient stabilised (BI-CGSTAB) method (van der Vorst, 1992) has been used in this regard. The hydrodynamic pressure at the 2D-3D interface is set to zero.

The finite volume method used in the present model is of the so called cell-vertex type. A hybrid finite element – finite volume algorithm has been used herein. Several estimations have been proposed in the literature to estimate the FV cell-area and cell-boundary integrals using nodal values, even for the hybrid FE – FV cells. The present study is based on a new algorithm. The integrals have been calculated analytically using shape functions. It has been done without projecting the real elements and cells into a master element. In this way, the coefficients that link a nodal value to its surrounding nodes (the matrix elements) are exactly accurate for the selected finite element shape functions. This technique has eliminated one of the numerical diffusion sources which is common in existing models. Six-node finite element triangles and three-node sub elements have been used in the present study. The divergence theorem has been used for predicting the spatial gradients on the cells. The unstructured finite difference method has also been incorporated as a fast but non-conservative option for predicting the nodal gradients.

The above model system has been verified using analytical solutions and benchmark test cases. The first test addresses the wind driven current, with and without the Coriolis force, in a large three dimensional domain (200km by 200km with depths from about 70 to 160 m). The dimensions of the domain were chosen such that the development of the Ekman profile in the ocean bodies can be verified. The simulation of the pure wind driven current (without the Coriolis force) resulted initially in a curved vertical velocity distribution which converged to a linear profile and finally reached to a steady state condition. This was then repeated for different viscosities. It showed that the numerical model results were in close agreement with the analytical ones. Then the Coriolis force was activated in the model to compare the predicted velocity distribution with the analytical Ekman profile. This test case was simulated for three water depths in order to check different developments of the spiral-type Ekman velocity profile. The numerical simulation results were again in close agreement with the analytical ones.

The next test case was simulating wind driven circulation in a closed square basin (10m by 10m with 5m depth) in order to show the flow field difference predicted with the hydrostatic and non-hydrostatic pressure assumptions. The simulation results showed that the near-wall vertical velocity in the hydrostatic pressure case was considerably larger than the hydrodynamic pressure case. The non-hydrostatic solution resulted in a smoother and reasonable circulation over the entire area. Then the simulated vertical velocity distribution at the middle of the basin was compared with an analytical solution (Huang, 1993), which showed a close agreement.

The next test case was the well-known lock exchange flow test case in order to verify the model capability for predicting the density-buoyancy current. A basin with 30m long and 3m wide with 4m water depth was selected in this regard. The results showed a smooth penetration of the salty water below the fresh water and also the smooth penetration of the fresh water above the salty water, with the frontal angle of the salt-fresh water equal to $\pi/3$, similar to the physical modelling observations (Turner J. 1973). The results were also compared with a similar simulation by a non-hydrostatic version of Telemac3D and the two are consistent. The numerical diffusion in that version of Telemac was more than the present model. The reason may be that Telemac model was based on linear elements (FE) while the present model uses a second-order approximation. The present model has the option of switching each physical process on or off separately. Therefore the simulation was repeated for several assumptions by activating and de-activating individual processes. Comparing the results showed the importance of the non-hydrostatic pressure distribution and the horizontal and vertical advection processes in this test case.

The next test case was to check the smoothness of integration and implicit dynamic data exchange between the 2DH and the 3D models. A semi-enclosed basin with the length of 200 meters and width of 100 meters was considered, with the initial water depth equal to 20 meters. The left side of the basin was simulated by the 2DH model while the right side was modelled by the 3D model. The two models were coupled at the middle of the basin. The predicted incoming wave moved smoothly towards the right side with no partial reflection or local oscillations when it was crossing the 2D-3D domains interface. The simulated reflected backwater wave also moved back towards the left side with no oscillations, false velocity or partial reflection at the coupling boundary and within each domain. Then the second input wave developed on top of the new water level towards the right end and the second backwater developed and moved to the left and so on. The implicit solution was as smooth, accurate and conservative as a single mesh simulation with no multi-dimensional coupling.

The test cases show that the proposed model is capable of simulating both the flow processes in 3D and integrated 2D/3D domains. Indeed, it must be applied to real lakes and reservoirs having field measurements and to test cases having experimental data for further investigation. The model already satisfies the necessary requirements for performing a smooth and adequate solution in real-world problems. This includes, among others, identical discretisation of the source terms to the main equation, unconditionally stable solution algorithm for the advection terms, modelling separately the vertical turbulence and horizontal turbulence, accurate discretisation of the baroclinic forces and shear stresses in the real horizontal plane, implicit discretisation of the bed shear stresses, implicit algorithm for the water surface and the vertical diffusion, stable procedure for the wetting-drying area and so on.

10 Conclusions and Recommendations

10.1 Conclusions

In the present research a new approach has been proposed using a 3D numerical model in conjunction with a 2D model. The integration of the two models is based on the method of fractional steps, an unstructured computational grid and hybrid 2D/3D finite volume cells.

This research has shown that the proposed approach is capable of solving integrated problems dynamically and efficiently without restricting the efficiency and robustness of the models involved. The fractional steps algorithm is a suitable framework for dynamic model integration in such a way that not only the coupling but also each one of the models may be efficient and implicit. It has also shown that it is possible and adequate to couple the 2DH and 3D models in a fraction by fraction manner with the most efficient solution algorithm being used for each physical process. The idea of using hybrid 2D/3D finite volume cells at the interface is a suitable tool for coupling the two models at each fraction and to solve the free-surface via a unified matrix equation for the total domain.

A crucial benefit of this integration approach is that most of the existing advanced numerical techniques (for example for shock capturing etc.) may be used within this framework without introducing new problems.

The main novelties of this research study are:

1- A dynamically integrated 2DH-3D model with a fully implicit solution method based on an unstructured computational grid has been developed.

2- Hybrid 2D-3D finite volume cells have been introduced on the unstructured grid which applies to the discretised equations of the link cells, involving nodes from both the 2D and the 3D domains. This method removed the need for an interpolation/averaging procedure at the interface and domain overlapping.

3- The 2D and 3D models have been coupled fraction by fraction within a fractional step algorithm in order to use the most efficient coupling algorithm for each fraction and each physical process.

4- The finite volume integrals have been derived analytically on the cell-areas and the cell-boundaries in order to eliminate the numerical diffusions regarding to the integral estimations. This procedure is based on the hybrid finite element – finite volume method in the model. Analytical integrations have been derived on the real cells and elements without projecting them to a master element.

5- A new set of the solution algorithms has been implemented for the 2DH and 3D model components, which are efficient both for the multi-dimensional modelling and also for the single-dimensional modelling.

6- The implicit behaviour of the fluid flow solution has been preserved by solving the free surface through a global matrix equation, which has been assembled by linking the local free surface matrices of the domains without using a global computational grid. The local matrices are linked throughout the hybrid FV cells. In this way, the accuracy and smoothness of coupling, i.e., without any numerical partial reflections at the domain interface are guaranteed.

7- Neumann interpolation elements with virtual nodes have been introduced at the boundaries, in order to avoid dense grid near a boundary with Neumann boundary condition.

10.2 Recommendations for further studies

In extending this research study in the future, the following aspects are worthy of further considerations:

- **Quadrilateral elements.** The triangular elements are not the most efficient elements in narrow and long water bodies like rivers. By adding quadrilateral finite element - finite volume cells to the model, it will be possible to use distorted elements with small node distances along the river's width and large distance along the river's length.
- **1D integration.** The present model may be coupled with a one dimensional model in order to simulate the rivers at the upstream and downstream of a reservoir within an integrated environment. This capability is useful for simulating the flood events.
- **2DV integration.** The present model may be coupled with a vertically two dimensional model in order to be more efficient in the narrow parts of the mountainous reservoirs and also in the deep but narrow parts of the rivers.
- **3D conservative semi-Lagrangian methods.** It is useful to add some kind of 3D conservative semi-Lagrangian advection method as an option for the problems where the 3D model must be run for a long time period.

- **Further calibration and validation of the model.** The numerical model developed should be applied to real lakes and reservoirs having field measurement data for further investigation of the flow patterns, sensitivity of model parameters and calibration and validation of the model.

11 References

Abbott M.B. and Basco D.R. (1989), “Computational Fluid Dynamics”, Addison Wesley Longman, England.

Abbott M.B., (1979), "Computational Hydraulics. Elements of the theory of free surface flow.", Pitman, London.

Abgrall R. (1994), “On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation”, J. Comput. Phys. 114, 45.

Agarwal R.K. and Halt D.W. (1999), “A Compact High-Order Unstructured Grids Method For The Solution of Euler Equations”, International Journal For Numerical Methods In Fluids, 31: 121–147.

Arandiga F., Belda A. M. (2004), “Weighted ENO interpolation and applications”, Communications in Nonlinear Science and Numerical Simulation 9, 187–195.

Armfield S., Street R. (2004), “Modified fractional-step methods for the Navier-Stokes equations”, ANZ IAM J. 45 (E) pp C364–C377.

Atkinson E. (1995), “A Numerical Model for Predicting Sediment Exclusion at Intakes”, HR Wallingford.

Atkinson E. (1996), “The Feasibility of Flushing Sediment from Reservoirs”, HR Wallingford.

Auclair F., Marsaleix P., Estournel C. (2000), “Sigma coordinate pressure gradient errors: evaluation and reduction by an inverse method”, J. Atmos. Ocean. Technol. 17. 1348-1367.

Baird & Associates Coastal Engineers LTD. (2008), “Final Report Of 3d Hydrodynamic Modelling of Chabahar Bay”, Canada.

Bakhvalov N.S. (1966), "On the Convergence of a Relaxation Method With Natural Constrains on the Elliptic Operator", USSR Computational Mathematics and Mathematical Physics, Vol. 6, No. 5, pp 101-135.

Baldwin B. S. and Lomax H. (1978), "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, Huntsville, AL.

Baldwin B.S. and Barth T.J. (1990), A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows, NASA TM 102847.

Bangerth, W. and G. Kanschat (2000). Concepts for Object-Oriented Finite Element Software – the deal.II Library, Institute of Applied Mathematics, University of Heidelberg, Germany.

Barth T. and Frederickson P. (1990), “High order solution of the Euler equations on unstructured grids using quadratic reconstruction”, AIAA Paper No. 90-0013.

Barth T. J. (1993), “Recent developments in high order K-exact reconstruction on unstructured meshes”, AIAA-93-0668, NASA Ames Research Centre, 1993.

Barth, T. J. (1995), “Aspects of unstructured grids and finite volume solvers for the Euler and Navier-stokes equations”, Lecture Notes Presented at the VKI Lecture Series 1994-05 Revised February 1995, Advanced Algorithms and Applications Branch, NASA Ames Research Center, Moffett Field, CA 94035.

Bates P.D. and J.M. Hervouet (1999), "A new method for moving-boundary hydrodynamic problems in shallow water", Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 455, 3107-3128.

Batteen M.L. (1988a), “Model simulations of a coastal jet and undercurrent in the presence of eddies and jets in the California Current System”, The Springer-Verla Lecture Note Volume.

Batteen, M.L. (1988b), “On the use of sigma coordinates in large-scale ocean circulation models”, Ocean Modelling, No 27.

Batten P, Lambert C, Causon DM., (1996), “Positively conservative high-resolution convection schemes for unstructured elements.”, International Journal for Numerical Methods in Engineering 39: 1821–1838.

Batuca, D.G. and Jordaan, J.M. (2000), Silting and desilting of reservoirs, A.A.Balkema, Rotterdam, Netherlands.

Becker, E. B., Carey, G. F., Oden, J. T., (1981). “Finite Elements; An Introduction – Vol 1”, Prentice-Hall.

Benjamin, T. (1968), “Gravity currents and related phenomena”, Journal of Fluid Mechanics, 31(2), 209–248.

Bernard P., Chevaugeon N., Legat V., Deleersnijder E. and Remacle J. (2005), “High-order h-adaptive discontinuous Galerkin methods for ocean modeling”, In Diez, P. and Wiberg, N.-E., editors, Proc. of ADMOS 2005, International Conference on Adaptive Modeling and Simulation, Barcelona, Spain. CIMNE.

Binning P. , Celia M.A. (2002), “A forward particle tracking Eulerian–Lagrangian Localized Adjoint Method for solution of the contaminant transport equation in three dimensions”, Advances in Water Resources 25, 147–157.

Bradford, S.F. and Sanders, B.F. (2002), “Finite-volume model for shallow-water flooding of arbitrary topography” J. Hydraul. Eng., 128(3), 289–298.

Bradshaw P. (1967), “The turbulence structure of equilibrium boundary layers”, J. Fluid Mech. (1967), vol. 29, part 4, pp. 625-645.

Bradshaw P. (1969) "The Analogy Between Streamline Curvature and Buoyancy in Turbulent Shear Flow," J.Fluid Mechanics, vol 36, pp. 177-191. -

Bradshaw P., Ferriss, D. H. and Atwell, N. P. (1967), "Calculation of Boundary Layer Development Using the Turbulent Energy Equations," Journal of Fluid Mechanics, Vol. 23, Pt. 3, pp. 593-616.

Branski J. M. and Holley E. R. (1986), “Advection calculations using spline schemes”, Water Forum '86: World Water Issues in Evolution, Proceedings of the Conference. ASCE, New York, NY, USA, Long Beach, CA, USA, 1807-1814.

Brufau P. , Garcia Navarro P. (2003), “Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique”, Journal of Computational Physics 186 503–526.

Brufau, P., Garcia-Navarro, P. and Vazquez-Cendon, M. E. (2002), “A numerical model for the flooding and drying of irregular domains”, Int. J. Numer. Methods Fluids, 39, 247–275.

Brufau P. and Garcia-Navarto P. (2000), “Two dimensional dam break flow simulation”, International Journal for Numerical methods in Fluids, 33: 35-57.

Cambier, L. (1999). The elsA project. First ONERA-DLR Aerospace Symposium. Paris.

Casulli V, Stelling GS (1998), “Numerical simulation of three dimensional, quasi-hydrostatic free-surface flows”, Journal of Hydraulic Engineering, 124(7), 678–686.

Casulli, V., and Stelling, G. (1995), “Simulation of three-dimensional, non-hydrostatic free-surface flows for estuaries and coastal seas”, In Estuarine and Coastal Modeling. Proceedings of the 4th International Conference, San Diego, M. Spaulding and R. Cheng, Eds., ASCE, New York, pp. 1–12.

Cebeci T. and Smith A.M.O. (1974), “Analysis of turbulent boundary layers”,. Academic Press, ISBN 0-12-164650-5

Cesare G. (1998), “Alluvionnement des Retenues par Courants de Turbidité”, , Thesis Report, Department of Civil Engineering Federal Polytechnic School of Lausanne.

Chan, C.T. and Anastasiou, K. (1999), “Solution of incompressible flows with or without a free surface using the finite volume method on unstructured triangular meshes”, Int. J. Numer. Methods Fluids, 29, 35–57.

Chand, K. K. (2003). Unstructured Hybrid Mesh Support for Overture, Livermore National Laboratory, Livermore, CA.

Chang, C.T., Kojasoy, G. and Landis, F. (1995), "Confined Single- and Multiple-Jet Impingement Heat Transfer-I. Turbulent Submerged Liquid Jets," Int. J. of Heat and Mass Transfer, Vol. 38(5), pp. 833-842.

Chen X. J. (2003), "A free-surface correction method for simulating shallow water flows", Journal of Computational Physics 189, 557–578.

Chen X. J. (2003), "A fully hydrodynamic model for three-dimensional, free-surface flows", Int. J. Numer. Meth. Fluids; 42 :929–952.

Chen Z. (2002), "Characteristic mixed discontinuous finite element methods for advection-dominated diffusion problems", Comput. Methods Appl. Mech. Engrg. 191: 2509–2538.

Chen Y. N., Yang S. C. and Yang J. Y. (1999), "Implicit Weighted Essentially Non-Oscillatory Schemes for the Incompressible Navier – Stokes Equations", Int. J. Numer. Meth. Fluids 31: 747 – 765.

Chen Ch., Liu H., Beardsley R. C. (2003), "An Unstructured Grid, Finite-Volume, Three-Dimensional, Primitive Equations Ocean Model: Application to Coastal Ocean and Estuaries", American Meteorological Society.

Chesshire, G. S. and W. D. Henshaw (2002), "The Overture Grid Classes Users' Guide, Version 1.0"

Chien, K.Y. (1982), "Predictions of channel and boundary-layer flows with a low-Reynolds-number turbulence model", AIAA Journal, 20, 33-38.

Chorin, A. (1967), "A numerical method for solving incompressible viscous flow problem", Journal of Computational Physics, 2, 12–26.

Chorin, A. (1968), "Numerical solution of the Navier-Stokes equations", Math. Comp., 22, 745–762.

Chukkapalli G. and Turan O.F. (1995), "Structural parameters and prediction of adverse pressure gradient turbulent flows: An improved k-e model", ASME Journal of Fluids Engineering, vol. 117, pp. 424-432.

Chung T.J. (2002), "Computational fluid dynamics", Cambridge University Press.

Corby, G. A., Gilchrist A. and Newson R. L. (1972), "A general cir-culation model of the atmosphere suitable for long period integrations", Quart. J. Roy. Meteor. Soc., 98, 809̃-832.

Daly B. J. and Harlow F. H. (1970), "Transport Equations in Turbulence," Physics of Fluids, Vol. 13, pp. 2634-2649

Degani D. and Schiff L.B. (1986), “Computation of turbulent supersonic flows around pointed bodies having cross-flow separation”, J. Comp. Phys., V.66, 1986, pp173-196.

Dietrich J.C., Kolar R. L. and Westerink J.J. (2006), “Refinements in Continuous Galerkin Wetting and Drying Algorithms”, Estuarine and Coastal Modeling, Spaulding, M. (ed.), ASCE, 637-65.

Dietrich J.C., Kolar R.L. and Luettich R. A. (2004), “Assessment of ADCIRCs Wetting and Drying Algorithm”, Proceedings of the XV International Conference on Computational Methods in Water Resources, Miller, C.; Farthing, M.; Gray, W., and Pinder, G. (ed.), 2, 1767-177.

Donea, J., (1983), “Arbitrary Lagrangian-Eulerian Finite Element Methods”, Computational Methods for Transient Analysis, 1, 101–119.

Dubois-Pelerin, Y. and P. Pegon (1997). "Improving Modularity In Object-Oriented Finite Element Programming." Communications In Numerical Methods In Engineering 13: 193-198.

Durbin, P. A. (1993), “A Reynolds stress model for near-wall turbulence”, J. Fluid Mech. 249, 465-498.

Einfeldt, B. (1988), “On Godunov-type methods for gas dynamics”, SIAM J. Numer. Anal. 25, 294.

Evans, R. A., and Roig, L. C. (1995). “Two-dimensional hydrodynamic modeling of wetland surface flows.” Proceedings of the National Workshop on Engineering for Wetlands Restoration, St. Louis, Missouri, 2 - 5 August 1993, TR-WRP-RE-8, USAE Waterways Experiment Station, Vicksburg, MS.

Ewing R. E., Wang H. (2001), “A summary of numerical methods for time-dependent advection-dominated partial differential equations”, Journal of Computational and Applied Mathematics 128 , 423–445.

Eyheramendy, D. and T. Zimmermann (1996). "Object-oriented finite elements II. A symbolic environment for automatic programming." Computer Methods in Applied Mechanics and Engineering 132: 277-304.

Falconer R. A., George D.G., and Hall P. (1991), “Three dimensional numerical modelling of wind driven circulation in a shallow homogeneuos lake”, ACSE Journal of Hydrology, Vol. 124, pp.59-79.

Falconer RA and Y Chen (1991), “An improved representation of flooding and drying and wind stress effects in a two-dimensional tidal numerical model”, Proceedings of the Institution of Civil Engineers 91:659–678.

Falconer, R. (1980), “Numerical modeling of tidal circulation in harbore”, J. Wtrwy., Prot.Coast., and Oc. Div., ASCE, 106(1), 31-48.

Fang H.W. and Wang G. Q. (2000), “Three-dimensional Mathematical Model for Suspended Sediment Transport”, Journal of Hydraulic Engineering, ASCE, 126(8), 578–592.

Ferziger J.H. and Peric M. (2002), “Computational Methods for Fluid Dynamics”, Springer, USA.

Fletcher, C., (1988), “Computational techniques for fluid dynamics”, Springer Verlag, Berlin Heidelberg New York.

Fletcher, R. (1976), “Conjugate Gradient Methods for Indefinite Linear Systems”, Lecture Notes in Mathematics 506, Springer-Verlag Berlin, 73-89.

Fluent Inc. (1998), “Fluent Manual”, Release 5.5.

Fortunato A. and Baptista A. (1996), “Evaluation of horizontal gradients in sigma – coordinate sha llow water models”, Atmosphere and Ocean, 34 :489 – 514.

Friedrich O. (1997), “Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids”, Hamburger Beitrage zur Angewandten Mathematik.

Friedrich O. (1998), “Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids”, Journal of Computational Physics 144 (1): 194-212.

Fromm, J.E. (1968), "A method for reducing dispersion in convective difference schemes", J. Comp. Phys., Vol. 3, p.176,

Gary J.M. 1973, “Estimate of truncation error in transformed coordinate primitive equation atmospheric models”, J. Atmos. Sci., 30:223-233.

Gil, L. and G. Bugeda (2001). "A C+ + object-oriented programming strategy for the implementation of the finite element sensitivity analysis for a non-linear structural material model." Advances in Engineering Software 32: 927-935.

Godunov, S. (1959), “A difference scheme for numerical computation of discontinuous solution of hydrodynamic equations”, Math. Sbornik, 43, 271-306.

Goldberg U., Apsley D. (1997), “A wall-distance-free low Re k - ϵ turbulence model”, Computer Methods in Applied Mechanics and Engineering, Volume 145, Issues 3–4, 30 June 1997, Pages 227–238.

Goldberg U.C. (1996), “Exploring a 3 EquationR-k”, ASME J. Fluid Mech., 12.

GoldbergU.C. (1991), “Derivation and testing of a one-equation model based on two time scales”, AIAA J. 29, p. 1337.

Gopalakrishnan T. (1989), "A moving boundary circulation model for regions with larg tidal flats", International Journal for Numerical Methods in Engineering, 28: 245-260.

Gresho P. M. and Chan S. T. (1990), “On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduce s a nearly consistent mass matrix. Part 2”, Int. J. Numer. Meth. Fluids, 11(5): 621-660.

Gresho P. M., Sani R. L. (2000), “Incompressible Flow and the Finite Element Method”, John Wiley and Sons, England.

Gresho, P., and Sani, R., (1987), “On pressure boundary conditions for the incompressible Navier Stokes equations”, International Journal for Numerical Methods in Fluids, 7, 1111–1145.

Guermond J.L., Shen J. (2003), “A new class of truly consistent splitting schemes for incompressible flows”, Journal of Computational Physics 192, 262–276.

Han Q. W. and He M. M. (1997), “2-D non-equilibrium transportation equation of non-uniform suspended load and its boundary condition.”, Chinese Journal of Hydraulic.

Haney RL. (1991), “On the pressure gradient force over steep topography in sigma co-ordinate ocean models”, AMS Journal of Physical Oceanography; 21(4):610–619.

Hanjalic K. & Launder B.E. (1980), "Sensitizing the dissipation equation to irrotalional strains", J. Fluids Eng., Vol.102, pp.34-40

Hanjalic K. and Launder B.E. (1972), “ A Reynolds Stress Model of Turbulence and its Application to Thin Shear Flows,” Journal of Fluid Mechanics, vol. 52, No. 4, pp. 609-638. -

Hanjalic K. and Launder B.E. (1976), "Contribution towards a Reynolds-stress closure for low-Reynolds-number turbulence", J. Fluid Mech. 74. 593-610

Hanjalic, K., Launder, B.E. and Schiestel, R., (1980), “Multiple-time-scale concepts in turbulent transport modelling, in Turbulent Shear Flows 2”, (eds. L.J.S. Bradbury, F. Durst, B.E. Launder, F.W. Schmidt, J.H. Whitelaw), Springer-Verlag.

Harten A. and Chakravarthy S. (1991), “Multi-dimensional ENO Schemes for General Geometries”, ICASE Report, No. 91-76.

Harten A., Engquist B., Osher S. (1997), “Uniformly High Order Accurate Essentially Non-oscillatory Schemes”, Journal of Computational Physics 131, 3–47.

Harten A., Osher S. (1987), “Uniformly high-order accurate non-oscillatory schemes. I”, SIAM J. Numer. Anal. 24 (2): 279–309.

Harten, A. (1983), “High resolution schemes for hyperbolic conservation laws”, J. Computational Phys., 49, 357-393.

Harten, A. and Zwas, G. (1972), “Self-adjusting hybrid schemes for shock computations”, J. Comput. Phys., 9, p. 568.

Harten, A., Lax, P.D. and van Leer, B. (1983), “On upstream differencing and Godunov-type schemes for hyperbolic conservation laws”, SIAM Rev., 25, 35–61.

Hatakeyama, M., Watanabe M. and Suzuki T. (1998). "Object-Oriented Fluid Flow Simulation System." Computers and Fluids 27(5): 581-597.

Healy R.W., Russell T.F. (1993), “A Finite Volume Eulerian-Lagrangian Localized Adjoint Method for Solution of the Advection-Dispersion Equation”, Water Resources Research, Vol. 29, No. 7, p2399-2413.

Henshaw W. D. (2002), “A Primer for Writing PDE Solvers with Overture”, Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, 94551.

Henshaw W. D. (2003), “OverBlown : A Fluid Flow Solver For Overlapping Grids - Reference Guide”, Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA.

Henshaw W. D. (2003), “OverBlown : A Fluid Flow Solver For Overlapping Grids - User Guide”, Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA.

Henshaw W. D. (2003), “OverBlownINS: The Incompressible Navier–Stokes Solver”, Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA.

Hirsch, C. (1988), Numerical computation of internal and external flows, Vol. I. John Wiley & Sons, Inc., New York, N. Y.

Holly F. M. Jr., and Preissmann A. (1977), “Accurate calculation of transport in two dimensions”, J. Hydraul. Div., Am. Soc. Civ. Eng., 103(11), 1259–1277.

Hsu, M. H., Kuo, W. C., Liu, W. C., and Kuo, J. T., (198X), “Numerical Simulation of Circulation and Salinity distribution in the Tanshui Estuary”, Proc. Natl. Sci. Cownc. ROC (A), 23(2), 259-273.

Hsu, M. H., Kuo, Y., Kuo, J. T., and Liu, W. C., “Procedure to Calibrate and Verify Numerical Models of Estuarine Hydrodynamics”, Journal of Hydraulic Engineering, ASCE, 125(2), 16-182.

Hu, K., Mingham, C.G. and Causon, D.M. (1998), “A bore-capturing finite volume method for open-channel flows”, Int. J. Numer. Methods Fluids, 28, 1241–1261.

Huang R. X. (1993), "Real freshwater flux as a natural boundary condition for the salinity balance and thermohaline circulation forced by evaporation and precipitation", J. Phys. Oceanogr., 23, 2428-2446.

Huang W., and Spaulding M. (1995), “3D model of estuarine circulation and water quality induced by surface discharges”, ACSE Journal of Hydraulic Engineering, Vol. 121, No. 4, pp.300-311.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.C., Tung, C. C. & Liu, H. H. (1998), “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis”, Proc. R. Soc. London, Ser. A 454, 903.

Huppert H.E. and Simpson J.E. (1980), The Slumping of Gravity Current, Journal of Fluid Mechanics, 99, pp 785-799.

Hwang Y. H. (2003), “Upwind scheme for non-hyperbolic systems”, Journal of Computational Physics 192, 643–676.

Iqbal M. (1983), "An introduction to solar radiation", Academic Press.

Jankowski, J. A. (1999), “A non-hydrostatic model for free surface flows”, Thesis Report, University of Hannover, Germany.

Jasak H. (1996), "Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows", PhD Thesis, Imperial College, UK.

Jawahar, P., and Kamath, H. (2000), “A high-resolution procedure for Euler and Navier-Stokes computations on unstructured grids”, J. Comput. Phys., 164, 165–203.

Jia Y. and Wang S. S. Y. (1999), “Numerical Model for Channel Flow and Morphological Change Studies”, Journal of Hydraulic Engineering, Vol. 125, No. 9, September.

Jiang Y.W. and Wai O.W.H. (2005), “Drying-Wetting Approach for 3D Finite Element Sigma Coordinate Model for Estuaries with Large Tidal Flats”, Advances in Water Resources, 28: 779-792.

Jingzhi Zhu J., Chen L., Shen J., Tikare V. (2001), “Computing the effective diffusivity using a spectral method”, Materials Science and Engineering A311, 135 – 141.

Kaazempur-Mofrad M.R. (1999), "A Characteristic Finite Element For 3-D Advection Dominated Transport Phenomena Using Unstructured Grids", Ph.D. Thesis, University of Toronto.

Kaazempur-Mofrad M.R., Ethier C.R. (2002), “An efficient characteristic Galerkin scheme for the advection equation in 3-D”, Comput. Methods Appl. Mech. Engrg. 191, 5345–5363.

Kantha and Clayson (2000), “Small Scale Processes in Geophysical Fluid Flows”, International Geophysics Series, Volume 67.

Kashefipour S.M., Lin B. and Falconer R.A. (2006), "Modelling the fate of faecal indicators in a coastal basin", Water Res. 40, 1413–1425.

Kashefipour S.M., Lin B., Harris E. and Falconer R.A. (2002), "Hydro-environmental modelling for bathing water compliance of an estuarine basin", Water Res. 36, 1854–1868.

Katopodes N. D., Kao K. Ch. (2003), “Nested Grid Model for Nearshore Hydrodynamics”, 16th Asce Engineering Mechanics Conference, Seattle.

Keats, W. A. and F. S. Lien (2004), "Two-dimensional anisotropic Cartesian mesh adaptation for the compressible Euler equations" International Journal for Numerical Methods in Fluids 46: 1099–1125.

Khan, L. A., and Liu, P., L.-F. (1998), “An operator splitting algorithm for the three-dimensional advection-diffusion equation.” International Journal for Numerical Methods in Fluids, Vol. 28, 461–476.

Khangaonkar TP, SA Breithaupt, and FC Kristanovich (2006), "Restoration of Hydrodynamic and Hydrologic Processes in the Chinook River Estuary, Washington – Feasibility Assessment", In 9th International Conference on Estuarine Coastal Modeling Conference , ed. ML Spaulding. American Society of Civil Engineering, New York, NY.

Kim D. H., Cho Y. S. and Kim W. G. (2004), “Weighted Averaged Flux-Type Scheme for Shallow-Water Equations with Fractional Step Method”, Journal of Engineering Mechanics, ASCE, pp 152-160.

Kim S. W. and Chen C. P. (1989), “A multiple-time-scale turbulence model based on variable partitioning of the turbulent kinetic energy spectrum”, Numerical Heat Transfer, No:16:, pp193-211.

Kim; D., Cho, Y. and Kim, W. (2004), “Weighted averaged flux-type scheme for shallow-water equations with fractional step method”, J. Hydraul. Eng., 130(2), 152–160.

King, I. P. (1988), "A Finite Element Model for Three Dimensional Flow," RMA report prepared for U.S. Army Corps of Engineers.

King, I.P. (1993), "RMA-10, a finite-element model for three dimensional density stratified flow", Dept. Civil Environ. Engrg., University of California, Davis.

King, I.P. (1998), "A Finite Element Model for Stratified Flow – RMA-10: User’s Guide – Version 6.5", Department of Civil and Environmental Engineering, University of California, Davis.

Knabner P., Angermann L. (2003), “Numerical Methods for Elliptic and Parabolic Partial Differential Equations”, Springer-Verlag New York.

Kobayashi, M. H., Pereira, J. M. C., and Pereira J. C. F. (1999), “A Conservative Finite-Volume Second-Order- Accurate Projection Method on Hybrid Unstructured Grids.” Journal of Computational Physics 150, 40–75.

Kocyigit M. B. , Falconer R. A. and Lin B. (2002), “Three-dimensional numerical modelling of free surface flows with non-hydrostatic pressure”, Int. J. Numer. Meth. Fluids; 40:1145–1162.

Kolmogorov, A. (1942), “Equations of Turbulent Motion of an Incompressive Fluid”, Izv. Akad. Nauk. SSR, Seria Fizicheska Vi, No, 1-2, 56-58, (English translation: Imperial College, Mechanical Engineering Department Report. ON/6, 1968).

Komatsu T., Holly Jr. F. M., Nakashiki N. and Ohgushi K. (1985), “Numerical calculation of pollutant transport in one and two dimensions”, J. Hydrosci. Hydr. Eng., 3(2), 15–30.

Komatsu, T., Ohgushi, K, and Asia, K. (1997), “Refined Numerical Scheme for Advective Transport in Diffusion Simulation”, Journal of Hydraulic Engineering, Vol. 123, No. 1, 41-50.

Krone R. B. (1962), "Flume Studies of the Transport of Sediment in Estuarial Shoaling Processes, Final Report", Hydraulic Engineer Laboratory and Sanitary Engineering Research Laboratory, University of California, Berkeley, Berkeley, CA.

Kruger, S., Rutschmann, P. (2006), “Modeling 3D Supercritical Flow with Extended Shallow-Water Approach.”, J. Hydr. Engrg., ASCE, 132(9), 916-926.

Lam, C.K.G. and Bremhorst, K.A. (1981), “Modified form of the k-e model for predicting wall turbulence”, Journal of Fluids Engineering, 103, 456-460.

Langseth, J. O. (1996), "On an implementation of a front tracking method for hyperbolic conservation laws", Advances in Engineering software 26: 45-63.

Laprise J. and Plante A. (1995), “A class of semi-Lagrangian integrated mass (SLIM) numerical transport algorithms” , Monthly Weather Review, 123, 2551-2566.

Launder B. E., Reece G. J. and Rodi W. (1975), "Progress in the Development of a Reynolds-Stress Turbulence Closure," Journal of Fluid Mechanics, Vol. 68, Pt. 3, pp. 537-566.

Launder, B. E. and Spalding, D. B. (1972), Mathematical Models of Turbulence, Academic Press, London.

Launder, B.E. and Sharma, B.I. (1974), “Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc”, Letters in Heat and Mass Transfer, 1, 131-138.

Lehfeldt R.,Bloss S. (1988), “Algebraic turbulence model for stratified tidal flows”, Physical processes in estuaries, J. Dronkers & W. Leussen van (eds.),Springer Verl.,278–291.

Leonard B. P. (1979), “A stable and accurate convective modelling procedure based on quadratic upstream interpolation”, Comput. Methods Appl. Mech. Eng., 19, 59–98.

Leonard, B. P. , Lock, A. P. and MacVean, M. K. (1995), "The NIRVANA scheme applied to one–dimensional advection", International Journal of Numerical Methods in Heat and Fluid Flow,5:341–377.

Leonard, B. P. , MacVean, M. K. and Lock, A. P. (1993), "Positivity-preserving numerical schemes for multidimensional advection.", Technical Memorandum TM-106055 ICOMP-93-05, NASA.

Leonard, B. P. (1988), "Elliptic systems: Finite-difference method IV.", In W. J. Minkowycz, E. M. Sparrow, G. E. Schneider, and R. H. Pletcher, editors, Handbook of Numerical Heat Transfer, pages 347–378. Wiley, New York.

Leonard, B. P. (1991), "The ULTIMATE conservative difference scheme applied to unsteady one–dimensional advection", Computer Methods in Applied Mechanics and Engineering, 88:17–74.

LeVeque, R.J. (1998), “Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm”, J. Comput. Phys., 146, 346-365.

LeVeque, R.J. (2002), “Finite volume methods for hyperbolic problems”, Cambridge University Press, UK.

Levy D., Tadmor E. (1997), “Non-Oscillatory Central Schemes for the Incompressible 2-D Euler Equations”, Mathematical Research Letters 4, 321–340.

Li B., Feleming C.A. (2003), “Three dimensional hydrodynamic model for free surface flow”, Journal of Hydraulic Research, Vol. 41, No. 4, pp. 367-377.

Li C.W. and Falconer R.A. (1995), “Depth-integrated modelling of tide-induced circulation in a square harbour”, J. of Hydraulic Research, 33(3), pp. 321-332.

Li J.G. and Atkinson B. W. (1999), “Transition regimes in valley air flows”, Bound.-Layer Meteor., 91, 385–411.

Li Z, Johns B. (2001), “A numerical method for the determination of weakly non-hydrostatic non-linear free surface wave propagation”, International Journal for Numerical Methods in Fluids; 35:2997–3317.

Lim Y.I., Lann J.M. L., Joulia X. (2001), “Accuracy, temporal performance and stability comparisons of discretization methods for the numerical solution of Partial Differential Equations (PDEs) in the presence of steep moving fronts”, Computers and Chemical Engineering 25, 1483 – 1492.

Lin B, Falconer RA. (1997), “Three-dimensional layer integrated modelling of estuarine flows with flooding and drying”, Estuarine, Coastal and Shelf Science; 44: 737 – 751

Lin B., Wicks J. M., Falconer R. A. and Adams K., (2006), “Integrating 1D and 2D hydrodynamic models for flood simulation”, Proceedings of the Institution of Civil Engineers, Water Management 159, March 2006 Issue WM1, Pages 19–25.

Lin L., Cialone M. A. and Bass G. P. (2004), “Hurricane Isabel storm surge model simulation for Chesapeake Bay”, Hurricane Isabel in Perspective Conference, 15-17 November 2004, ASCE. Virginia Beach, VA.

Liu H.H. and J.H. Dane. (1996), "An interpolation corrected modified method of characteristics to solve advective-dispersion equations", Advances in Water Resour. 19:359-368.

Liu, S.K. and J.J. Leendertse. (1978), “Multidimensional Numerical Modeling of Estuaries and Coastal Seas” , Advances in Hydroscience, 11, 95-164, Academic, New York

Liu. W.C., Hsu, M.H., and Kuo. A.Y. (2001), “Application of Different Turbulence Closure Model for Stratified Tidal Flows and Salinity in an Estuarine System”, Mathematics and Computers in Simulation, 59, 437-451.

Liu. W.C., Hsu, M.H., and Kuo. A.Y. (2001), “Application of Different Turbulence Closure Model for Stratified Tidal Flows and Salinity in an Estuarine System”, Mathematics and Computers in Simulation, 59, 437-451

Lo D.C., Young D.L. (2004), “Arbitrary Lagrangian–Eulerian finite element analysis of free surface flow using a velocity–vorticity formulation”, Journal of Computational Physics 195, 175–201.

Lohner, R., Morgan, K., Perier, J. and Vahdati, M. (1987), “Finite element flux-corrected transport (FEM-FCT) for Euler and Navier Stokes equations”, Int. J. Numer. Methods in Fluids, 7, 1093-1109.

Lu Q. and Wai W.H. (1998), “An Efficient Operator Splitting Scheme for Three-Dimensional Hydrodynamic Computations”, Int. J. Numer. Meth. Fluids 26: 771 – 789.

Lu Q. (2003), "A Three-Dimensional Modelling of Tidal Circulation in Coastal Zones With Wetting and Drying Process", International Conference on Estuaries and Coasts November 9-11, Hangzhou, China.

Lumley J. L. (1978), "Computational Modelling of Turbulent Flows," Adv. Appl. Math., Vol. 18, pp. 123-176.

Luyten P. J., Jones J. E., Proctor R., Tabor A., Tett P. and Wild-Allen K. (1999), "COHERENS – A coupled hydrodynamical-ecological model for regional and shelf seas: user documentation", MUMM Rep., Management Unit of the Mathematical Models of the North Sea.

MacArthur R. C., Dexter J. R., Smith D. J. and King I. P. (1990), "Two Dimentional Finite Element Simulation of The Flooding Characteristics in Kawainui Marsh, Hawai", Proceedings of 1990 National Hydraulic Engineering Conference, ASCE, New York, 0-87262-774-8:664-669

Magesh R., Ruhle R. (2003), "Quadratic reconstruction on arbitrary polygonal grids for 2nd-order conservation laws", in: R. Herbin, D. Kroner (Eds.), "Finite Volumes for Complex Applications III", Kogan Page Science, London, Sterling, 2003.

Mahrer Y. (1984), “An improved numerical approximation of the horizontal gradients in a terrain-following coordinate system”, Mon. Wea. Rev., 112, 918-922.

Malan, A. G. and R. W. Lewis (2004). "On the development of high-performance C++ object-oriented code with application to an explicit edge-based fluid dynamics scheme." Computers and Fluids 33: 1291–1304.

Mc Anally W.H., Letter J.V. and Thomas W.A. (1986), “Two- and three-dimensional modeling systems for sedimentation”, Proc. Third Int. Symp. on River Sedimentation, The University of Mississippi, USA, pp. 400–411.

Mehta A.J., Hayter E.J., Parker W.R., Krone R.B., Teeter A.M. (1989), “Cohesive Sediment Transport. I: Process Description”, Journal of Hydraulic Eng., Vol. 115, No. 8, pp. 1076-1093.

Mellor G. L., Ezer T. and Oey L.-Y. (1994), “The pressure gradient conundrum of sigma coordinate ocean models”, J. Atmos. Oceanic Technol., 11, 1126 – 1134.

Mellor, G. L., Oey L.-Y. and Ezer T. (1998), “Sigma coordinate pressure gradient errors and the seamount problem”, J. Atmos. Oceanic Technol., 15, 1122 – 1131.

MellorG.L., Blumberg, A.F. (1985), “Modeling vertical and horizontal diffusivities with the sigma coordinate system”, Mon. Weather Rev. 113, 1380–1383.

Menter F.R. (1997), "Eddy viscosity transport equations and their relation to the k-e model", Journal of Fluids Engineering-Transactions of the ASME, 118(3):876-884

Menter F.R., and Scheuerer, G. (1998), “Turbulence Modelling with Perspective to CFD,”Lecture Notes, AEA Technology Users Conference, Wilmington, DE. -

Mingham, C.G. and Causon, D.M. (1998), “High-resolution finite-volume method for shallow water flows”, Journal of Hydraulic Engineering, 124(6), 605-614.

Mohamadian M. (2006), “Conservative Characteristic-Based Schemes for Shallow Flows”, Thesis Report, University of Quebec.

Mohammadian A., Le Roux D. (2006), “A Conservative Semi-Lagrangian Method for the Shallow Water Equations”, Computer Physics Communications, Vol. 174 (2), pp.99-108.

Munk, W.H. and Anderson E.R. (1948), “Notes on the Theory of the Thermocline”, J. Marine Res., 1, 7276-7295.

Nagano Y. and Hishida M. (1987), “Improved form of the k–ε model for wall turbulent shear flows”, J. Fluid Eng., pp. 109–156

Namin M. M. , Lin B. and Falconer R. A. (2001), “An implicit numerical algorithm for solving non-hydrostatic free-surface flow problems”, Int. J. Numer. Meth. Fluids; 35: 341 – 356.

Namin M. M. (2003), “A Fully Three-Dimensional non-Hydrostatic Free Surface Flow Model for Hydro-Environmental Predictions”, Thesis Report, Cardiff University, UK.

Nielsen C. and C. Apelt (2003), "Parameters affecting the performance of wetting and drying in a two dimensional finite element long wave hydrodynamic model", Journal of Hydraulic Engineering, 129, 628-636

Nujic, M. (1995), “Efficient implementation of non-oscillatory schemes for the computation of free surface flows”, J. Hydr. Res., 33(1), 101-111.

Oliveira A., Babtista A.M. (1995), “A Comparison of Integration and Interpolation Eulerian-Lagrangian Methods”, International Journal for Numerical Methods in Fluids Vol. 21, 183-204.

Olliver-Gooch C. F. (1997), "High-order ENO schemes for unstructured meshes based on least-squares reconstruction", AIAA paper 97-0540.

Ollivier-Gooch, C. F, and Van Altena, M. (2002), “A High-Order Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation”, University of British Columbia, Department of Mechanical Engineering, Revised June 25, 2002

Ollivier-Gooch, C.F. (1997), “Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction”, J. Comput. Phys. 133, pp 6–17.

Olsean N. R. B. (2003), "3D CFD Modelling of a Self-Forming Meandering Channel", ASCE Journal of Hydraulic Engineering, No. 5.

Olsean N. R. B. (2003), "A Three-Dimensional Numerical Model for Simulation of Sediment Movements in Water Intakes with Multi-block Option, User Manual", Department of Hydraulic and Environmental Engineering, The Norwegian University of Science and Technology.

Olsen N.R.B. (2004), “Hydro-informatics, Fluvial Hydraulics and Limnology”, Norwegian University of Science and Technology.

Olsen, N. R. B. (1999), “Computational Fluid Dynamics in Hydraulic and Sedimentation Engineering”, The Norwegian University of Science and Technology.

Olsen, N. R. B. (2000), “CFD Algorithms for Hydraulic Engineering”, The Norwegian University of Science and Technology.

Olsen, N. R. B. (2009), “Numerical Modelling and Hydraulics”, The Norwegian University of Science and Technology.

Olsen, N. R. B. and Haun, S. (2010), "Free surface algorithms for 3D numerical modelling of reservoir flushing", River Flow 2010, Braunschweig, Germany.

Olsen, N. R. B., Jimenez, O. F., Abrahamsen, L. and Løvoll, A. (1999), "3D CFD modelling of water and sediment flow in a hydro-power reservoir", International Journal of Sediment Reserach, Vol. 14, No. 1

Omelchenko Y.A., Karimabadi H. (2006), “Self-adaptive time integration of flux-conservative equations with sources”, Journal of Computational Physics 216, 179–194.

Osher, S. and Solomone, F. (1982), “Upwind difference schemes for hyperbolic systems of conservation laws”, Math. Comput., 38, 339–374.

Pacanowski, R. C., and S.G.H. Philander (1981), “Parameterisation of vertical mixing in numerical models of Tropical Oceans”, J. Phys. Oceanogr., 11, 1443-1451

Pan D, Cheng JC (1993), “Upwind finite volume Navier Stokes computations on unstructured triangular meshes”, AIAA Journal 31(9): 1618 – 1625.

Pan H. and Damodaran M. (2002), “Computation of unsteady viscous incompressible flows in generalized non-inertial co-ordinate system using Godunov-projection method and overlapping meshes”, Int. J. Numer. Meth. Fluids; 40 :1365–1380.

Park S. H., Kwon J. H. (2003), “On the dissipation mechanism of Godunov-type schemes”, Journal of Computational Physics 188 (2003) 524–542

Park K., Kuo A.Y. (1993), “A vertical two-dimensional model of estuarine hydrodynamics and water quality”, Spec. Rep. App. Mar. Sci. Ocean Eng. No. 321, Virginia Institute of Marine Science, Gloucester Point, VA

Park K., Kuo A.Y. (1994), “Numerical modeling of advection and diffusion transport in the Rappahannock estuary”, Proceedings of 3rd International Conference of Estuarine and Coastal Modeling, ASCE, New York, pp. 461–474.

Patankar S.V. (1980), “Numerical Heat Transfer and Fluid Flow”, Hemisphere , Washington D.C.

Patankar S.V. and D.B. Spalding. (1970), “Heat and Mass Transfer in Boundary Layers”, 2nd ed., Intertext, London, UK.

Perrels P.A.J. , Karelse M. (1981), “ A two-dimensional, laterally averaged model for salt intrusion in estuaries Transport Models for Inland and Coastal Waters, Academic Press, New York, pp. 483–535

Prandtl L. (1925), "Uber die ausgebildete Turbulenz," ZAMM, Vol. 5, pp. 136-139

Prandtl L. (1942), “Fuhrer durch die Stromungslehre”, Vieweg & Sohn, Braunschweig, 519 pp.

Prandtl L. (1945), "Uber ein neues Formelsystem fur die ausgebildete Turbulenz," Nacr. Akad. Wiss. Gottingen, Math-Phys. Kl. 1945, pp6-19.

Restelli, M., Bonaventura, L., and Sacco, R. (2006), “A semi-Lagrangian discontinuous Galerkin method for scalar advection by incompressible flows.” Journal of Computational Physics 216, 195–215

Reynolds W. C. (1987), "Fundamentals of Turbulence for Turbulence Modeling and Simulation," In Lecture Notes for von Karman Institute, AGARD Lecture Series No.86, pp. 1-66, New York: NATO.

Reynolds O. (1895), “On the dynamical theory of incompressible viscous fluids and the determination of the criterion”, Philosophical Transactions of the Royal Society of London, A186, 123–164.

Roberts S. and Zoppou C. (2000), “Robust and efficient solution of the 2D shallow water equation with domains containing dry beds”, ANZIAM J. 42 (E) ppC1260-C1282.

Robertson A. W., Ghil M. and Latif M. (2000), “Interdecadal changes in atmospheric low-frequency variability with and without boundary forcing”, J. Atmos. Sci., 57, 1132–1140.

Rodi W. (1993), "Elements of the theory of turbulence", In Coastal, estuarial and harbours engineers' reference book (ed. Abbott M.B. and Price W.A.), London:Spon

Rodi, W. (1980), “Turbulence Models and Their Application in Hydraulics-A State of the Art Review”, International Association for Hydraulic Research, Delft, Netherlands.

Rodi W. (1984), “Turbulence models and their applications in Hydraulics” (2nd edn). IAHR Publication: Delft.

Rodi W. (1984), “Examples of Turbulence-Model Applications”, Simulation of Turbulence Models and Their Application, 2, Collection de la Direction des Etudes et Recherches, Electricite de France, editions Eyrolles, Paris, France

Rodi W. (1991), “Experience with Two-Layer Models Combining the k-e Model with a One-Equation Model Near the Wall”, AIAA Paper 91-216, Reno, NV. -

Roe P.L. (1981), “Approximate Riemann solvers, parameter vectors and difference schemes”, J. Computational Phys., 43, 357-372.

Roig L. C. (1995), "Mathematical theory and numerical methods for the modeling of wetland hydraulics", In Water Resources Engineering, Proceedings of the 1995 First International Conference San Antonio, Texas, August 14–18, 1995. New York: ASCE, 249–253.

Roig L. C. (1994). “Hydrodynamic modeling of flows in tidal wetlands”, Ph.D. Diss., University of California, Davis.

Rotta J. C. (1951), "Statistische Theorie nichthomogener Turbulenz," Zeitschrift fur Physik, Vol. 129, pp. 547-572.

Rubinstein R. and Barton J.M. (1990), “Nonlinear Reynolds stress models and the renormalisation group”, Physics of Fluids, series A, 2, 1472-1476.

Saint Venant B. (1871), “Theorie de mouvement Non-permanant des eaux avec application aux crues des rivieres et a l’introduction des marees dans leur lit” Acad. Sci., Paris, C. R., 73, 148–154, 237–240.

Sebastian K., Shu C. (2003), “Multidomain WENO Finite Difference Method with Interpolation at Subdomain Interfaces”, Journal of Scientific Computing,Vol.19, Nos.1–3.

Shchepetkin A. F. and McWilliams J. C. (2003), “A method for computing horizontal pressure-gradient force in an oceanic model with a nonaligned vertical coordinate”, J. Geophys. Res., 108(C3), 3090.

Shen . (1994), “Remarks on the pressure error estimates for the projection methods”, Numer. Math. Vol. 67, 513-520.

Shen, J., (1993), “A remark on the projection-3 method”, International Journal for Numerical Methods in Fluids, 16, 249–253.

Shih T.H. (1996), “Developments in Computational Modeling of Turbulent Flows,” NASA Contract Report No.198458, February, Lewis Research Center.

Shih T.H., Zhu J., and Lumley J. L. (1993), "A Realisable Reynolds Stress Algebraic Equation Model," NASA TM105993. -

Shih, T-H., Zhu, J. and Lumley, J.L. (1995), “A new Reynolds stress algebraic equation model”, Computer Methods in Applied Mechanics and Engineering, 125, 287-302.

Shima, N. (1998), “Low-Reynolds-number second-moment closure without wall-reflection redistribution terms”, International Journal of Heat and Fluid Flow, 19, 549-555. -

Shu C. (1997), “Preface to the Republication of “Uniformly High Order Essentially Non-oscillatory Schemes”, Journal of Computational Physics 131, 1–2.

Shu C., Chew Y. T. and Niu X. D. (2001), “Least-squares-based lattice Boltzmann method: A meshless approach for simulation of flows with complex geometry”, Physical Review E, Volume 64, 045701.

Simmons A. J., Burridge D. M. (1981), “An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates”, Mon. Wea. Rev. 109, 758-766.

Singh C.B., Prasad N. (2001), “Three dimensional Numerical Simulation for an Open Channel Flow with a Constriction”, Journal of Hydraulic Research, Vol. 39, 2001, Issue 2, pp. 187-201.

Sleigh P.A., Gaskell P.H., Bersins M. and Wright N.G. (1998), "An unstructured finite-volume algorithm for predicting flow in rivers and estuaries", Computers and Fluids, Vol. 27, No. 4, 479-508.

Smagorinsky, J (1963), "General circulation experiments with the primitive equations, i. the basic experiment. Monthly Weather Review", 91: pp 99-164.

Sonar T. (1997), “On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conser-vation laws on general triangulations: Polynomial recovery, accuracy and stencil selection”, Comput. Methods Appl. Mech. Engrg. 140, 157.

Song Y. T. and Wright D. G. (1998), “A general pressure gradient formulation for the ocean models. part 2: Momentum and bottom torque consistency”, Mon. Weather Rev., 126, 3213 – 3230.

Song, Y. T. (1998), “A general pressure gradient formulation for ocean models. Part 1: Scheme design and diagnostic analysis”, Mon. Weather Re v. , 126.

Spalart P.R. and Athmaras S.R. (1994), “La Recherche”, Aérospatiale, Vol. 1, pp. 5-21.

Speziale C. G. (1985), "Modeling the Pressure-Gradient-Velocity Correlation for Turbulence," Physics of Fluids, Vol. 28, pp. 69-71.

Speziale C. G., (1998) “Turbulence Modeling for Time-Dependent RANS and VLES: A Review,” AIAA Journal, Vol. 36, No. 2, pp. 173-184.

Speziale C.G., Abid R. and Anderson E.C. (1992), “Critical Evaluation of Two Two-Equation Models for Near-Wall Turbulence”, AIAA Journals, 30, pp 324-331.

Stansby. P.K, Zhou. J. G, (1998), “Shallow- Water Flow Solver with Non-Hydrostatic Pressure: 2D Vertical Plane Problems”, Int. J. Numer. Methods Fluids, 28, 541–563.

Stansby. P.K. (1997), “Semi-Implicit Finite Volume Shallow-Water Flow and Solute Transport Solver with k– ε Turbulence Model” , Int. J. Numer. Methods Fluids, 25, 285–313.

Stefanovic, D. L., and Stefan, H. G. (2001). “Accurate two-dimensional simulation of advective-diffusive-reactive transport.” Journal of Hydraulic Engineering, Vol. 127, No. 9, 728-737.

Steinhorn I. (1991), "Salt flux and evaporation", J. Phys. Ocesnogr. 21 (1991) 1681–1683.

Stelling. G, van Kester. A.T.M. (1994), “On the Approximation of Horizontal Gradients in Sigma Co-ordinates for Bathym etry with Steep Bottom Slopes” , Int. J. Numer. Methods Fluids, 18, 915–935.

Sun Y. and Wang Z.J. (2003), “Evaluation of Discontinuous Galerkin and Spectral Volume Methods for Conservation Laws on Unstructured Grids”, 41st Aerospace Sciences Meeting and Exhibit 6-9 January, 2003 Reno, Nevada, AIAA-2003-0253.

Takizawa K., Yabe T., Nakamura T. (2002),”Multi-dimensional semi-Lagrangian scheme that guarantees exact conservation”, Computer Physics Communications 148 (2002) 137–159.

Tanaka R., Nakamura T., and Yabe T. (2000), “Constructing exactly conservative scheme in a non-conservative form”, Computer physics communications, 126, 232-243.

Thomas, T., Leslie, D., and Williams, J. (1995), “Free surface simulations using a conservative 3D code”, Journal of Computational Physics, 116, 52–68.

Thompson J. F., Warsi Z. U. A., Mastin C. W. (1985), "Numerical Grid Generation: Foundations and Applications", North-Holland, 1985.

Thompson, J. F., Soni, B. and Weatherrill, N. P. (1999), “Handbook of Grid Generation”, CRC Press

Titov, V. V., and Synolakis, C. E. (1995). “Modeling of breaking and nonbreaking long-wave evaluation and runup using VTCS-2.” Journal of Waterway, Port Coast. Ocean Engineering, 121(6), 308–316.

Toro, E. F. (1999), “Riemann solvers and numerical methods for fluid dynamics”, Springer, New York.

Toro, F.E. (2001), “Shock-capturing methods for free-surface shallow flows”, John Wiley & Sons, Chichester, England.

Trottenberg U., Oosterlee C.W. and Schuller A. (2001), “Multi-Grid”, Academic Press, UK.

Tsai, T. L., Yang, J. C, and Huang, L. H. (2004), “Characteristics Method Using Cubic–Spline Interpolation for Advection–Diffusion Equation.” Journal of Hydraulic Engineering, Vol. 130, No. 6, 580-585.

Tsuruya H., Nakamo S., and Kato H. (1985), “Experimental study on wind driven current in a wind-wave tank-effect of return flow on wind driven current”, The Ocean Surface, pp. 425-430.

Turner, J. (1973), “Buoyancy effects in fluids”, Cambridge University Press. 367 pp.

Unesco (1987), “International Oceanographic Tables, Vol. 4”, Unesco technical papers in marine science, No. 40.

Váchal, P., Liska, R. and Wendroff, B. (2004), “Fully Two Dimensional HLLEC Riemann Solver”, Proceedings of Czech-Japanese Seminar in Applied Mathematics 2004 August 4-7, 2004, Czech Technical University in Prague pp. 195-206.

Valiani, A., Caleffi, V. and Zanni, A. (2002), “Case study: Malpasset dam-break simulation using a two-dimensional finite volume method”, Journal of Hydraulic Engineering, 128(5), 460-472.

Van Altena, M. (1999), “High-order finite-volume discretisations for solving a modified advection-diffusion problem on unstructured triangular meshes”, Mater of Applied Science Thesis, University of British Columbia, Department of Mechanical Engineering, October,1999

van der Vorst, H.: BiCGStab: A Fast and Smoothly Converging Variantof Bi-CG for the Solution of Nonsymmetric Linear Systems, SIAM J.Sc. Stat. Comp. 13, 631-644 (1992).

Van Driest E. R. (1956), "On Turbulent Flow Near a Wall," Journal of the Aeronautical Sciences, Vol. 23, p. 1007.

van Leer B. (1977), “Toward the ultimate conservative difference scheme. IV: A new approach to numerical convection”, J. Comput. Phys., 23, 276–299.

van Leer B. (1979), “Toward the ultimate conservative difference scheme. V: A second order sequel to Godunov’s method”, J. Comput. Phys., 32, 101–136.

van Leer, B. (1973), “Towards the ultimate conservative difference scheme. I. The quest of monotonicity”, Lecture Notes in Phys., 18, 163-168.

van Leer, B. (1982), “Flux vector splitting for the Euler equations”, Proc., 8th Int. Conf. on Numer. .Methods in Fluid Dyn., E. Krause, ed., Springer Verlag, Berlin, Germany, 507-512.

van Rijn, L.C. (1986), “Mathematical modeling of suspended sediment in non-uniform flows”, Journal of Hydraulic Engineering, 112(6), 433-455.

Van Rijn. L. C (1985), “Two-Dimensional Ver tical Mathematical Model for Suspended Sediment Transport by Currents and Waves”, Delft Hydraulics Communication.

Van Rijn. L. C (1987), “Mathematical Modeling of Morhological Processes in the case of Suspended Sediment Transport”, Delft Hydraulics Communication No. 382.

Vasiliev, O. F., (2000), “Vertical two dimensional hydrodynamic Models for Water Bodies”, the state of the art and current issues, Russia

Versteeg H. K. and Malalasekera W. (1995), “An Introduction to Computational Fluid Dynamics, the Finite Volume Method”, Addison Wesley Longman, England.

Verwey A. (2001), “Latest Developments in Floodplain Modelling – 1D/2D Integration”, 6th Conference on Hydraulics in Civil Engineering, Hobart, Australia.

Wai O., Lu Q. (2000), “An efficient parallel model for coastal transport process simulation”, Advances in Water Resources 23, 747-764.

Wai O.H. And Lu Q. (1999), ”Gradient Adaptive Sigma (GAS) grid for 3D mass transport modelling”, ACSE Journal of Hydraulic Engineering, Vol. 125, No. 2, pp.141-151.

Wang H., Dahle H. K., Ewing R. E., Espedal M. S., Sharpley R. Sh. And Man Sh. (1999), “An ELLAM Scheme For Advection-Diffusion Equations In Two Dimensions”, Siam J. Sci. Comput. Mechanics, 99, pp 785-799.

Wang J. D., Cofer-Shabica S. V. and Chin Fatt J. (1988). "Finite element characteristic advection model.", J. Hydr. Engrg., ASCE, 114(9), 1098–1114.

Wang J. S. , Ni H. G. and He Y. S. (2000), “Finite-Difference TVD Scheme for Computation of Dam-Break Problems”, Journal of Hydraulic Engineering, April 2000.

Wang Z.J. , Zhang L., Liu Y. (2004), “Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems”, Journal of Computational Physics 194, 716–741.

Wang Z.J. (2003), “High-Order Spectral Volume Method for Benchmark Aeroacoustic Problems”, 41st Aerospace Sciences Meeting and Exhibit 6-9 January, 2003 Reno, Nevada

Wang J. W. and Liu R. X. (2000), “A comparative study of finite volume methods on unstructured meshes for simulation of 2D shallow water wave problems”, Mathematics and Computers in Simulation, 53, 171–184.

Wang J. W. and Liu, R. X. (2001), “The composite finite volume method on unstructured meshes for the two-dimensional shallow water equations”, Int. J. Numer. Meth. Fluids; 37: 933–949

Wang J.D. and J.J. Connor. (1975), “Mathemati cal Modeling of Near Coastal Circulation”. Report No. 217, Massachusetts Institute of T echnology, Department of Civil Engineering, Cambridge, MA.

Wang Z.J. (2000), “A fast nested multi-grid viscous flow solver for adaptive Cartesian/Quad grids”, Int. J. Numer. Meth. Fluids, 33: 657–680.

Wesseling P. (2001), "Principles of Computational Fluid Dynamics", Springer, Germany.

Wierse M., (1997), “A new theoretically motivated higher order upwind scheme on unstructured grids of simplicies.”, Advances in Computational Mathematics; 7: 303–335.

Wilcox D. (1994), "Turbulence Modeling for CFD", DCW Industries, Inc., California.

Wilcox D. C. (1988a), "Reassessment of the Scale Determining Equation for Advanced Turbulence Models," AIAA Journal, Vol. 26, No. 11, pp. 1299-1310.

Wilcox, D.C. (1988), “Reassessment of the Scale-Determining Equation for Advanced Turbulence Models”, AIAA Journal 26: 1299-1310. 39

Wilcox, D.C. (1994), “Simulation of Transition with a Two-Equation Turbulence Model”, AIAA J. 32: 247-254.

Wood, W., A., and Kleb, W., L. (1999), “Diffusion Characteristics of Finite Volume and Fluctuation Splitting Schemes”, Journal of Computational Physics 153, 353–377.

Wu W. (2007), “Computational River Dynamics”, Taylor & Francis Group, London.

Wu W. and Li Y. (1992), "One- and two-dimensional nesting mathematical model for river flow and sedimentation", The Fifth Int. Symp. on River Sedimentation, Karlsruhe, Germany, 1, 547-554.

Wu, J. K., (1997). “A wave equation model to solve the multi dimensional transport equation.” International Journal for Numerical Methods in Fluids, Vol. 24, 423–439

Xiao F. (2002), “Profile modifiable conservative transport schemes and a simple multi-integrated moment formulation for hydrodynamics”, Computational Fluid Dynamics 2002, Proceedings of the 2nd international conference on CFD, Sydney, Australia, Springer, 2003, ISBN 3-540-00739-3.

Xiao F. and Yabe T. (2001), “Completely conservative and oscillation-less semi-Lagrangian schemes for advection transportation”, Journal of computational physics, 170, 498-522.

Yabe T., Mizoe H., Takizawa K., Moriki H., Im H., Ogata Y. (2004), “Higher-order schemes with CIP method and adaptive Soroban grid towards mesh-free scheme”, Journal of Computational Physics 194, 57–77.

Yabe T., Tanaka R., Nakamura T. and Xiao F. (2001), “Exactly conservative semi-Lagrangian methods (CIP-CSL) in one dimension”, Monthly Weather Review, 126, 232.

Yakhot V., Orszag S.A., Thangam S., Gatski T.B. And Speziale C.G. (1992), “Development of Turbulence models for Shear Flows by a Double Expansion Technique”, Physics of Fluids, 4, pp 1510-1519.

Yanenko N.N. (1971), “The Method of Fractional Steps”, Springer, New York.

Yang, J. and Hsu, C. (1993), “Computation of free surface flows”, J. Hydr. Res., Delft, The Netherlands, 31(3), 403-413.

Ye H.C. (1997), “Explicit and Implicit Multidimensional Compact High-Resolution Shock-Capturing Methods”, Journal of Computational Physics 131, 216–232.

Ye J., Mccorquodale J.A., Barron R.M. (1998), “A Three-Dimensional Hydrodynamic Model in Curvilinear Co-Ordinates with Collocated Grid”, Int. J. Numer. Meth. Fluids 28: 1109 – 1134.

Ye J.Y., McCorquodale J. A. (1998), “Simulation of Curved Open Channel Flows by 3D Hydrodynamic Model”, Juornal of Hydraulic Engineering.

Yih, C. S. (1980), “Stratified flows”, Academic Press, London, New York. 418 pp.

Yoon, T.H. and Kang, S. (2004), “Finite volume model for two-dimensional shallow water flows on unstructured grids”, Journal of Hydraulic Engineering, 130(7), 678-688

Younga D.L., Changa J.T., Eldho T.I. (2004), “Solution of three-dimensional unsteady external flow using a coupled arbitrary Lagrangian FEM – BEM model”, Engineering Analysis with Boundary Elements 28, 711–723.

Yu L., Geng Z., Roma W.N.L., Righetto A.M. and Xiong S. (1998), "Two- and Three-Dimensional Nested Simulation by Using FEM and FVA to Analyze Flows in an Estuary", Mathl. Comput. Modelling, Vol. 28, No. 11, pp. 115-134

Zalesak ST. (1979), "Fully multidimensional flux-corrected transport algorithms for fluids", Journal of Computational Physics 31: 335–362.

Zängl G., Gantner L., Hartjenstein G. and Noppel H. (2004), “Numerical errors above steep topography. A model intercomparison”, Meteorol. Z., 13, 69-76.

Zerroukat M, Wood N, Staniforth A. (2002), “SLICE: A semi-Lagrangian inherently conserving and efficient scheme for transport problems”, Q. J. R. Meteorol. Soc. 128: 801 – 820.

Zhang J. (1996), “A cost-effective multi-grid projection operator”, Journal of Computational and Applied Mathematics 76, 325-333.

Zhang S.Q. (1999), “One-D and two-D combined model for estuary sedimentation”, Int. J. Sediment Res., 14(1), 37–45

Zhang X., Takeuchi, K. and Ishidaira, H. (2004), “Simulation of floodplain morphological variation to sediment laden inundation flows in the Lower Yellow River basin”, Advances in Hydro-Science and Engineering, Volume VI, 1-10.

Zhao D.H., Shen H.W., Lai J.S. and Tabious G.Q. (1996), “Approximate Riemann solvers in FVM for 2D hydraulic shock wave modeling”, ASCE J. Hydraul. Eng., 122(12), 692–702.

Zhao D.H., Shen H.W., Tabious G.Q., Lai J.S. and Tan W.Y. (1994a), “Finite-volume two-dimensional unsteady flow model for river basins”, ASCE J. Hydraul. Eng., 120(7), 864–883.

Zheng L., Chen C. and Liu H. (2003), "A 3-D modeling study of the Satilla River estuarine System. Part I: simulation of flooding/drying processes", Estuaries, 26(3), 651-669.

Zheng L., Chen C., Alber M. and Liu H. (2003), "A 3-D modeling study of the Satilla River estuarine System. Part II: suspended sedimentation", Estuaries, 26(3), 670-679.

Zhou JG, Stansby PK. (1999), “An arbitrary Langrangian–Eulerian sigma (ALES) model with non-hydrostatic pressure for shallow water flows”, Computational Methods in Applied Mechanical Engineering 1999; 178:199–214

Zhou JG. (1995), “Velocity-depth coupling in shallow-water flows”, Journal of Hydraulic Engineering, ASCE, 121: 717 – 724.

Zhou, J.G., Mingham, C.G. Causon,D.M. and Ingram, D.M. (2002), “Accurate Numerical Treatment of the Source Terms in the Nonlinear Shallow Water Equations”, Proceedings of the Fifth International Conference on Hydroinformatics, Cardiff, UK, 207-211.

Zhu J. (1991), “Low Diffusive and oscillation-free convection scheme”, Communications and Applied Numerical Methods, 7, N3. 225-232.

Zhu J., Rodi W. (1991), "A low dispersion and bounded convection scheme", Comp. Meth. Appl. Mech.& Engng, Vol. 92, p 225.

Zhu J., Rodi W. (1991), “A low dispersion and bounded discretization schemes for finite volume computations of incompressible flows”, Computational Methods for Applied Mechanics and Engineering, 92, 87-96

Ziegler U. (2004), “An ADI-based adaptive mesh Poisson solver for the MHD code NIRVANA”, Computer Physics Communications 157, 207–216.

Zienkiewicz O.C. and Taylor R.L. (2000), "The Finite Element Method - Volume 3: Fluid Dynamics", Butterworth-Heinemann, Oxford.

Zienkiewicz O., and Taylor R. (1991), “The finite element method”, McGraw-Hill, London, New York. Fourth edition, Vol. I, Vol. II.

Zienkiewicz, O., and Codina R. (1995), “A General Algorithm for Compressible and Incompressible Flow. Part I- The Split Characteristic Based Scheme”, Int. J. Num. Meth. In Fluids, 20:869-885.

Zyserman J.A. And Fredsoe J. (1994), “Data Analysis of Bed Concentration of Suspended Sediment”, Journal of Hydraulic Engineering, Vol. 120, No. 9, pp. 1021-1042.