Comparison of two numerical schemes for solving the 1D shallow water equations

Aukje de Boer

Chair of Numerical Analysis and Computational Mechanics,
Faculty of Mathematical Sciences,
University of Twente, The Netherlands.

August 28, 2003

Supervisors:

Dr. Ir. O. Bokhove, University of Twente,
Dr. Ing. D. Schwanenberg, WL | Delft Hydraulics,
Dr. Ir. I. Wenneker, WL | Delft Hydraulics ,
Dr. R.M.J. van Damme, University of Twente.

WL | Delft Hydraulics
Rotterdamseweg 185
2629 HD Delft
The Netherlands

Abstract

Shallow water flows in rivers and coastal environments are characterized by complex boundaries demarcating wet and dry topography. Unstructured grids are ideally suited to model those complex computational domains, because of their large geometric flexibility.

The aim of this M.Sc. thesis work is to compare two numerical methods that are able to compute 1D shallow water flows. The methods must in principle be applicable to unstructured grids when extending them to two dimensions.

The first method, the so-called Collocated Coupled Solution Method (CCSM), is a collocated, vertex-centred finite volume method, in which a linear approximation of the primary variables is used. Due to the use of a first order upwind scheme in the momentum equation the method is first order accurate in space. Discretization of the advection term in the continuity equation is either done by means of a central or a first order upwind approach. The time marching procedure is based on solving the coupled matrix system of equations at once by means of a θ-method. It is quite simple to make changes in the discretization, because it only implies a change in the computation of certain matrix elements and not of the solution procedure. Picard linearization or an iterative process is used to deal with the nonlinear terms and for this reason the CCSM-method is semi-implicit which allows for relative large time steps. The choice for the time step is only limited by accuracy reasons.

The Runge-Kutta Discontinuous Galerkin finite element method is the second method examined in this M.Sc. thesis. It uses a discontinuous piecewise polynomial approximation for the variables, and the method is very local. More degrees of freedom are involved relative to finite volume methods. The HLLC approximate Riemann solver is used to locally solve the Riemann problem between two cells, as is common in most finite volume methods. A diagonal mass matrix is obtained by the use of orthogonal Legendre polynomials and this avoids the necessity to solve a local matrix system. For the solution procedure an explicit TVB Runge-Kutta scheme is used and this gives a CFL-restriction on the time step, which becomes more severe when the order of the method increases. A slope limiter can be applied to avoid unphysical oscillations in higher order schemes.

Some numerical test cases are performed to test the numerical methods and to compare them with each other. They include a linear wave problem, a problem with one of the Riemann variables being constant, Riemann problems, tidal movement and flow over an isolated ridge. The second order RKDG-method gives the best results in all test cases. The first order RKDG-method and the CCSM-method give most of the times results of comparable quality, being more diffusive than the second order RKDG-method. A slope limiter has to be used for the second order RKDG-method when large gradients in the solution are present. The upwind approach in the continuity equation has to be applied in the CCSM-method when the flow is supercritical or in the presence of large gradients in the solution.

It can be concluded that both the RKDG-method and the CCSM-method are capable of computing 1D shallow water flows. At least for flow involving low Froude numbers the CCSM-method introduced in this report can use larger time-steps than the RKDG-method used in this M.Sc. thesis, but the RKDG-method is better suited to deal with discontinuities in the flow.

Preface

This report is the result of a seven months project at WL | Delft Hydraulics. The project is performed as part of the curriculum of the study Applied Mathematics at the University of Twente and is the final stage in obtaining the grade of Master of Science. The goal of this final project is to apply the gained experience and skills to a well defined problem of sufficient mathematical level.

The Faculty of Mathematical Sciences (MS) has a five-year scientific based engineering education at a Technical University. The graduates at this faculty find a job in Industry. Complex problems from technical science areas as well as real world problems in Industry and Public Administration inspire MS.

Research at WL | Delft Hydraulics involves all kinds of flow problems occurring in real live situations. Most of the problems are solved by numerical models and this gave me the opportunity to do a project within my specialization: numerical modelling of physical phenomena.

At WL | Delft Hydraulics and the chair Numerical Analysis and Computational Mechanics (NACM) at the University of Twente, exploratory research has started on the development of unstructured grid methods. The aim of my work is to get insight in the performance and accuracy of two numerical methods for solving the shallow water equations on unstructured grids.

During the project I gained the support of a lot of people. At first I want to thank my direct supervisors Onno, Dirk and Ivo for all their good advise and reading my report over and over again. I want to thank Karin for the ability to share all the frustrations and also the highlights with somebody in the same position. The distraction of my roommate at WL, Tessa, during our chitchat about everything, gave me the opportunity to think of something else then working on the project. I also will remember the lunches with the multi-cultural students at WL, and our dinners in Delft. My visits to Onno in Enschede gave me the opportunity to meet all the friends that are still left over there. And finally, this project wouldn't been possible without the support of my boyfriend, Martijn, and my family, although they hardly understood what I've been doing.

Contents

Chapter 1

Introduction

Most flows on the surface of the earth, for example in rivers, seas and the atmosphere, are shallow water flows in which the horizontal length and velocity scales of interest are much larger than the vertical ones. The mathematical formulation of these flows, the so-called shallow water equations, are already known for over a century, but the equations involved are far too complex to be solved analytically in the majority of cases. In the past this meant that physical experiments had to be carried out to model the flow, or very simplified mathematical models were used.

The advance in computing power gave rise to an alternative way to solve the equations: the use of numerical simulations became possible. Nowadays, experiments are most of the times far too expensive compared to the use of numerical models. In the past decades a large number of different numerical methods are developed to solve the shallow water equations. At present, these models are used in all kinds of applications such as flood warning systems, design of harbours, impact of changes in water systems, climate predictions and reducing water pollution. And due to the increasing computer power and speed, more complicated problems can be solved by the day.

Closely related to this development is the generation of grids for the numerical methods. In the past the computations were carried out on a simple structured grids because of the relative simplicity of the computational domains that were involved. Thanks to the advances in computer technology, problems on more and more complex domains can be solved. Unstructured grids are ideally suited to model those complex computational domains, because of their large geometric flexibility. The grid generation can also be automated to a larger extent than for structured grids, and local and adaptive grid refinement is easier. However, the discretization of the flow equations is quite difficult and as a result the computational cost are higher.

At WL | Delft Hydraulics and the chair Numerical Analysis and Computational Mechanics (NACM) at the University of Twente, exploratory research has started on the development of unstructured grid methods. The aim of this M.Sc. thesis work is to get insight in the performance and accuracy of two numerical methods for solving the shallow water equations on unstructured grids.

Formulation of the problem

Two numerical methods that are able to compute 1D shallow water flows are compared. The methods must in principle be applicable to unstructured grids when extending them to two dimensions.

Methodology

The research consists of the following steps:

- Understanding of the shallow water equations and the problems involved with unstructured grids.
- Investigation of numerical methods which are able to solve the shallow water equations.
- Performing a literature study on flooding and drying algorithms.
- Developing of a new finite volume method, the CCSM-method, for solving the one-dimensional shallow water equations with first order accuracy.
- Applying the first order RKDG-method to the one-dimensional shallow water equations.
- Implementing both methods in Fortran 90 and testing them by means of several test cases.
- Extending the RKDG-method to second order accuracy, including a slope limiter.

Structure of the M.Sc. thesis

The derivation of the shallow water equations (SWE) from the Navier-Stokes equations is given in Chapter 2. The final result of this derivation are the one-dimensional shallow water equations which are the equations of interest in this work.

In Chapter 3 the contrast between structured and unstructured grids is outlined, the difference between staggered and collocated grids is explained and the distinction between a vertex-centred and cell-centred approach is given. The motivation to prefer one approach over the other is also included.

Four different numerical methods that can be used to solve the shallow water equations are outlined in Chapter 4. The two methods that are compared in this work are the CCSM finite volume method and the RKDG finite element method. They are examined in detail in Chapter 5.

A literature survey of various flooding and drying algorithms is given in Chapter 6.

In Chapter 7 some numerical test cases are performed to test the numerical methods and to be able to compare them with each other. They include a linear wave problem, a problem with one of the Riemann variables being constant, Riemann problems, tidal movement and flow over an isolated ridge.

The final chapter, Chapter 8, contains the conclusions of the research and some recommendations for further investigations.

Chapter 2

Derivation of the shallow water equations

The Navier-Stokes equations are the governing equations to model fluid flow in many applications, such as flow through pipes and around aircrafts. In most flows on the surface of the earth, for example in seas, rivers and the atmosphere, the horizontal length and velocity scales are much larger than the vertical ones. In this case a simplification can be made by assuming that the vertical acceleration is negligible compared to the horizontal accelerations. This results in a hydrostatic pressure distribution, and the shallow water equations can be used instead of the more complicated full Navier-Stokes equations. In this chapter the shallow water equations (SWE) will be derived from the Navier-Stokes equations. First the three-dimensional (3D) SWE will be derived in Section 2.1. These will be integrated over the water depth to obtain in Section 2.2 the two-dimensional (2D) depth-integrated SWE. In for example rivers and channels the length scales of the flow are also much larger than the scales related to the width variations of the flow. In this case the equations can be simplified even more, by looking at equations for the cross-section of the flow, to obtain the so called Saint-Venant equations in Section 2.3. If the width of the channel is constant, the Saint-Venant equations reduce to the one-dimensional (1D) SWE as described in Section 2.4. The derivation of the SWE, as given in this chapter, is based on the derivations found in Young et al. (1997), Johnson (1998), Toro (2001), Vreugdenhil (1994) and Schwanenberg (2003). Finally the boundary conditions for the one-dimensional case are examined in Section 2.5.

2.1 Three-dimensional shallow water equations

The Navier-Stokes equations are the equations that govern flows in many realistic situations. For water, we can reasonably assume that the density is constant, which means that water is incompressible, and therefore we only look at the incompressible Navier-Stokes equations. They consist of the equations for the conservation of mass and momentum:

$$\text{mass:} \qquad \partial_{x_j} v_j = 0, \tag{2.1}$$

$$\text{momentum:} \qquad \partial_t v_i + \partial_{x_j}(v_j v_i - \frac{1}{\rho}\sigma_{ij}) = g_i, \tag{2.2}$$

where ρ is the density of water, v_j the velocity components, x_j the directional indices, σ_{ij} the stresses of deformation and g the acceleration due to gravity. The indices $i, j \in \{x, y, z\}$ indicate the spatial directions and in addition, partial derivatives are denoted as $\partial_t = \partial/\partial t$.

The stresses of deformation can be expressed as

$$\sigma_{ij} = -p\delta_{ij} + \mu\left(\partial_{x_j} v_i + \partial_{x_i} v_j\right), \tag{2.3}$$

where p is the pressure in water, μ its dynamic viscosity (assumed to be constant) and δ_{ij} the Kronecker delta which has the properties: $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. Consequently (2.1) and (2.2) can be written as

$$\partial_x u + \partial_y v + \partial_z w = 0, \tag{2.4}$$

$$\partial_t u + \partial_x u^2 + \partial_y(uv) + \partial_z(uw) = -\frac{1}{\rho}\partial_x p + \nu\left(\partial_x^2 u + \partial_y^2 u + \partial_z^2 u\right), \tag{2.5}$$

$$\partial_t v + \partial_x(vu) + \partial_y v^2 + \partial_z(vw) = -\frac{1}{\rho}\partial_y p + \nu\left(\partial_x^2 v + \partial_y^2 v + \partial_z^2 v\right), \tag{2.6}$$

$$\partial_t w + \partial_x(wu) + \partial_y(wv) + \partial_z w^2 = -g - \frac{1}{\rho}\partial_z p + \nu\left(\partial_x^2 w + \partial_y^2 w + \partial_z^2 w\right), \tag{2.7}$$

where u, v and w represent the x-, y- and z- velocity components respectively and $\nu = \mu/\rho$ is the kinematic viscosity. Here we also assumed that $g_x = g_y = 0$ and $g_z = -g$.

We consider shallow water flow with a free surface under gravity for which the geometry for a fixed value of y is depicted in Figure 2.1. We introduce a reference level at $z = 0$ and the bottom is assumed to be fixed in time and is defined by a given function

$$z = -h(x, y). \tag{2.8}$$

The free surface is defined by

$$z = \zeta(x, y, t), \tag{2.9}$$

and the total water depth is defined as

$$H(x, y, t) = h(x, y) + \zeta(x, y, t). \tag{2.10}$$

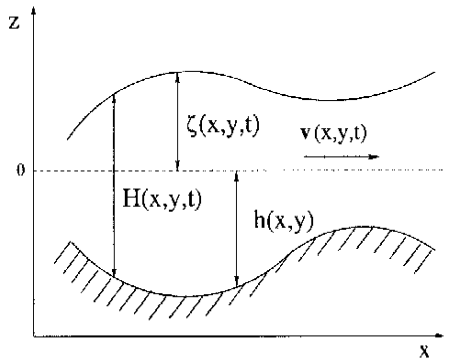


Fig. 2.1: Flow with a free surface.

In shallow water flow, by definition, the horizontal length and velocity scales are much larger than the vertical ones. Therefore the vertical acceleration is assumed to be negligible compared to the other terms appearing in the z-momentum equation (2.7), so $|Dw/Dt| << |g + \partial_z p/\rho|$ and $|\nu\nabla^2 w| << |g + \partial_z p/\rho|$, where $D/Dt = \partial_t + u\partial_x + v\partial_y + w\partial_z$ is the material derivative

and $\nabla^2 = \partial_x^2 + \partial_y^2 + \partial_z^2$. This implies that the z-momentum equation reduces to the hydrostatic pressure distribution

$$\frac{1}{\rho}\partial_z p = -g. \tag{2.11}$$

This equation can be integrated, by assuming the atmospheric pressure to be constant, without loss of generality taken to be zero, to yield

$$p = \rho g(\zeta - z). \tag{2.12}$$

Substituting equation (2.12) in (2.5) and (2.6) yields

$$\partial_t u + \partial_x u^2 + \partial_y(uv) + \partial_z(uw) = -g\partial_x\zeta + \nu\left(\partial_x^2 u + \partial_y^2 u + \partial_z^2 u\right), \tag{2.13}$$

$$\partial_t v + \partial_x(vu) + \partial_y v^2 + \partial_z(vw) = -g\partial_y\zeta + \nu\left(\partial_x^2 u + \partial_y^2 u + \partial_z^2 u\right). \tag{2.14}$$

Equations (2.4), (2.12), (2.13) and (2.14) are the incompressible hydrostatic Navier-Stokes equations also called the 3D SWE. The unknowns are the velocity vector $\mathbf{v} = (u, v, w)$ and the surface elevation ζ.

The conditions on the free surface and the bottom are given by the assumption that any fluid particle located at the surface or the bottom remains at the surface or the bottom respectively and the bottom boundary is assumed not to change in time. The vertical velocity of the particle at the surface and the bottom is then given by

$$w_{z=\zeta} = \left.\frac{D\zeta}{Dt}\right|_{z=\zeta} = \left[\partial_t\zeta + u\partial_x\zeta + v\partial_y\zeta\right]_{z=\zeta}, \tag{2.15}$$

$$w|_{z=-h} = -\left.\frac{Dh}{Dt}\right|_{z=-h} = -\left[u\partial_x h + v\partial_y h\right]_{z=-h}, \tag{2.16}$$

where the notation $f|_{z=\alpha}$ means that $f(z)$ is evaluated in the point $z = \alpha$. Equations (2.15) and (2.16) are called the kinetic boundary conditions for the incompressible hydrostatic Navier stokes equations.

2.2 Two-dimensional shallow water equations

The incompressible hydrostatic Navier stokes equations can be simplified by integrating them over the total water depth resulting in the 2D SWE, also called the depth-averaged or depth-integrated SWE. Integrating the individual terms of the continuity equation (2.4) over the total water depth yields

$$\int_{-h}^{\zeta} \partial_x u\, dz = \partial_x \int_{-h}^{\zeta} u\, dz - u|_{z=\zeta}\cdot\partial_x\zeta - u|_{z=-h}\cdot\partial_x h, \tag{2.17}$$

$$\int_{-h}^{\zeta} \partial_y v\, dz = \partial_y \int_{-h}^{\zeta} v\, dz - v|_{z=\zeta}\cdot\partial_y\zeta - v|_{z=-h}\cdot\partial_y h, \tag{2.18}$$

$$\int_{-h}^{\zeta} \partial_z w\, dz = w|_{z=\zeta} - w|_{z=-h}, \tag{2.19}$$

where we used Leibniz's formula

$$\partial_\alpha \int_{\xi_1(\alpha)}^{\xi_2(\alpha)} f(\xi,\alpha)\, d\xi = \int_{\xi_1(\alpha)}^{\xi_2(\alpha)} \partial_\alpha f\, d\xi + f(\xi_2,\alpha)\partial_\alpha\xi_2 + f(\xi_1,\alpha)\partial_\alpha\xi_1. \tag{2.20}$$

Define the depth-averaged velocity terms

$$\bar{u} = \frac{1}{H}\int_{-h}^{\zeta} u\,dz \quad \text{and} \quad \bar{v} = \frac{1}{H}\int_{-h}^{\zeta} v\,dz \tag{2.21}$$

to remove the integrals in equations (2.17) and (2.18). Substituting these in the integrated continuity equation yields

$$\begin{aligned}\partial_x(\bar{u}H) - u|_{z=\zeta}\cdot\partial_x\zeta - u|_{z=-h}\cdot\partial_x h + \partial_y(\bar{v}H) - v|_{z=\zeta}\cdot\partial_y\zeta \\ - v|_{z=-h}\cdot\partial_y h + w|_{z=\zeta} - w|_{z=-h} = 0.\end{aligned} \tag{2.22}$$

The physical conditions on the free surface and the bottom are implemented by substituting the kinetic boundary conditions (2.15) and (2.16) in (2.22) to yield

$$\partial_t\zeta + \partial_x(\bar{u}H) + \partial_y(\bar{v}H) = 0. \tag{2.23}$$

Recalling that $\zeta = H - h$ and h independent of time, the final form of the two-dimensional continuity equation can be expressed as

$$\partial_t H + \partial_x(\bar{u}H) + \partial_y(\bar{v}H) = 0. \tag{2.24}$$

The momentum equation can be depth-integrated using the same procedure. Integrating the individual terms of the x-component of the momentum equation, (2.5), over the total depth yields

$$\int_{-h}^{\zeta}\partial_t u\,dz = \partial_t\int_{-h}^{\zeta} u\,dz - u|_{z=\zeta}\cdot\partial_t\zeta, \tag{2.25}$$

$$\int_{-h}^{\zeta}\partial_x u^2\,dz = \partial_x\int_{-h}^{\zeta} u^2\,dz - u^2|_{z=\zeta}\cdot\partial_x\zeta - u^2|_{z=-h}\cdot\partial_x h, \tag{2.26}$$

$$\int_{-h}^{\zeta}\partial_y uv\,dz = \partial_y\int_{-h}^{\zeta} uv\,dz - uv|_{z=\zeta}\cdot\partial_y\zeta - uv|_{z=-h}\cdot\partial_y h, \tag{2.27}$$

$$\int_{-h}^{\zeta}\partial_z uw\,dz = uw|_{z=\zeta} - uw|_{z=-h}, \tag{2.28}$$

$$\int_{-h}^{\zeta} g\partial_x\zeta\,dz = g\partial_x\zeta(\zeta + h) = gH\partial_x\zeta = gH\partial_x\left(H - h\right), \tag{2.29}$$

and we define the depth-integrated friction term S_{fx} of the x-component of the momentum equation as follows

$$S_{fx} = \int_{-h}^{\zeta}\nu\left(\partial_x^2 u + \partial_y^2 u + \partial_z^2 u\right)\,dz. \tag{2.30}$$

We assume that the distribution of the velocity is uniform in the z-plane, which results in the fact that the integrands in equations (2.25), (2.26) and (2.27) can be taken out of the integral and we can write $\bar{u} = u$ and $\bar{v} = v$. Then along with the kinetic boundary conditions (2.15) and (2.16), the x-momentum equation yields

$$\partial_t(uH) + \partial_x\left(u^2H\right) + \partial_y\left(uvH\right) = gH\partial_x(h - H) + S_{fx}. \tag{2.31}$$

Following the same procedure, the y-component of the momentum equation becomes

$$\partial_t(vH) + \partial_x\left(uvH\right) + \partial_y\left(v^2H\right) = gH\partial_y(h - H) + S_{fy}. \tag{2.32}$$

Equations (2.24), (2.31) and (2.32) are now called the 2D depth-integrated shallow water equations. These equations can be written in differential conservative law form

$$\partial_t \mathbf{u} + \nabla \cdot \mathbf{f}(\mathbf{u}) = \mathbf{s}(\mathbf{u}), \tag{2.33}$$

with the differential operator defined as $\nabla \cdot \mathbf{f} = \partial_x \mathbf{f}_x + \partial_y \mathbf{f}_y$. The vector containing the unknown variables $\mathbf{u}$, the flux vector $\mathbf{f} = (\mathbf{f}_x, \mathbf{f}_y)$ and the vector containing the source terms $\mathbf{s}(\mathbf{u})$ are given by

$$\mathbf{u} = \begin{bmatrix} H \\ uH \\ vH \end{bmatrix}, \qquad \mathbf{f}_x(\mathbf{u}) = \begin{bmatrix} uH \\ u^2H \\ uvH \end{bmatrix},$$
$$\mathbf{f}_y(\mathbf{u}) = \begin{bmatrix} vH \\ uvH \\ v^2H \end{bmatrix}, \qquad \mathbf{s}(\mathbf{u}) = \begin{bmatrix} 0 \\ -gH\partial_x\zeta + S_{fx} \\ -gH\partial_y\zeta + S_{fy} \end{bmatrix},$$

or

$$\mathbf{u} = \begin{bmatrix} H \\ uH \\ vH \end{bmatrix}, \qquad \mathbf{f}_x(\mathbf{u}) = \begin{bmatrix} uH \\ u^2H + \frac{1}{2}gH^2 \\ uvH \end{bmatrix},$$
$$\mathbf{f}_y(\mathbf{u}) = \begin{bmatrix} vH \\ uvH \\ v^2H + \frac{1}{2}gH^2 \end{bmatrix}, \qquad \mathbf{s}(\mathbf{u}) = \begin{bmatrix} 0 \\ gH\partial_x h + S_{fx} \\ gH\partial_y h + S_{fy} \end{bmatrix},$$

depending on the kind of solution procedure that will be used.

2.3 Saint-Venant equations

In flows in rivers or channels, for example, the main flow is in the x-direction. This means that both the acceleration in the y- and z-direction are assumed to be negligible. In this case the equations can be simplified even more, by looking at equations for the cross-section of the flow instead of the total water depth.

To get a one dimensional formulation of the shallow water equations we consider the mass and momentum equation for an infinitesimal control volume. The geometry and quantities of the infinitesimal control volume are illustrated in Figure 2.2. We assume that the velocity is uniform

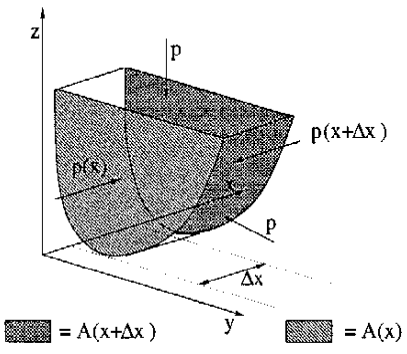


Fig. 2.2: Infinitesimal control volume for the Saint-Venant equations.

in the y- and z-direction, because the accelerations in the y- and z- direction are assumed to be

small. The mass and momentum equation for this infinitesimal control volume, without viscous terms are

$$\partial_t(A\Delta x) - (Au)(x) + (Au)(x+\Delta x) = 0, \tag{2.34a}$$

$$\begin{aligned}\partial_t(Au\Delta x) - (Au^2)(x) + (Au^2)(x+\Delta x) = &\frac{1}{\rho}\left[(Ap)(x) - (Ap)(x+\Delta x)\right] \\ &+ \frac{1}{\rho}p\left[A(x+\Delta x) - A(x)\right],\end{aligned} \tag{2.34b}$$

where $A(x)$ is the cross-section, Δx the infinitesimal length of the control volume and p the pressure on the control volume. The density ρ is again assumed to be constant. The last term on the right hand side of the momentum equation (2.34b) can be derived by dividing the control volume in even smaller sub-volumes by dividing it also in infinitesimal pieces Δy and Δz in the y- and z- direction and assuming that this infinitesimal volumes have quadrilateral sides. The contribution of the pressure in the x-direction in each sub-volume can be simply calculated and the total contribution on the infinitesimal volume is then the sum of all the sub-volumes, by taking the limit $\Delta y, \Delta z \to 0$.

Dividing equations (2.34) by Δx and taking the limit $\Delta x \to 0$ will give the following differential equations

$$\partial_t A + \partial_x(Au) = 0, \tag{2.35a}$$

$$\partial_t(Au) + \partial_x(Au^2) = -\frac{1}{\rho}A\partial_x p. \tag{2.35b}$$

By assuming again that the pressure is hydrostatic as given in (2.11), due to the shallow water assumption, (2.35) reduce to

$$\partial_t A + \partial_x(Au) = 0, \tag{2.36a}$$

$$\partial_t(Au) + \partial_x(Au^2) = gA\partial_x(h-H). \tag{2.36b}$$

Equations (2.36) are the 1D cross-sectional shallow water equations that are commonly referred to as the Saint-Venant equations.

By making use of the chain rule to evaluate $A\partial_x H$, this equations can also be written in (quasi-) conservation law form

$$\partial_t A + \partial_x(Au) = 0, \tag{2.37a}$$

$$\partial_t(Au) + \partial_x(Au^2 + gAH) = gA\partial_x h - gH\partial_x A. \tag{2.37b}$$

A problem is that there is now a differential term in the unknown variable A on the right hand side as part of the source terms and this means that the equations are not completely in conservative form.

2.4 One-dimensional shallow water equations

If we look at a very simple channel with a constant width, W, then $A = HW$ and we get the following simplification of equations (2.36):

$$\partial_t H + \partial_x(uH) = 0, \tag{2.38a}$$

$$\partial_t(uH) + \partial_x(u^2 H) = gH\partial_x(h-H). \tag{2.38b}$$

Introducing the discharge $q = uH$ the equations become

$$\partial_t H + \partial_x q = 0, \tag{2.39a}$$
$$\partial_t q + \partial_x (uq) = -gH\partial_x \zeta. \tag{2.39b}$$

In this way the equations express that the term $gH\partial_x\zeta$, which represents the water surface slope, is the driving force of variations in the discharge.

If however a Riemann problem has to be solved the momentum flux of the 'real' conservative form is needed and then the equations (2.38) are usually written in the following form

$$\partial_t H + \partial_x q = 0, \tag{2.40a}$$
$$\partial_t q + \partial_x \left(q^2/H + \tfrac{1}{2}gH^2\right) = gH\partial_x h. \tag{2.40b}$$

In this case, the bottom slope is represented as a source term instead of the water surface slope, so that the source term does not contain a derivative of a primary variable.

The equations can be written in conservative law form as follows

$$\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) = \mathbf{s}(\mathbf{u}), \tag{2.41}$$

with

$$\mathbf{u} = \begin{bmatrix} H \\ q \end{bmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} q \\ uq \end{bmatrix}, \quad \mathbf{s}(\mathbf{u}) = \begin{bmatrix} 0 \\ -gH\partial_x\zeta \end{bmatrix} \tag{2.42}$$

for equations (2.39), and

$$\mathbf{u} = \begin{bmatrix} H \\ q \end{bmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} q \\ q^2/H + \frac{1}{2}gH^2 \end{bmatrix} \quad \text{and} \quad \mathbf{s}(\mathbf{u}) = \begin{bmatrix} 0 \\ gH\partial_x h \end{bmatrix} \tag{2.43}$$

for equations (2.40).

2.5 Boundary conditions in one dimension

To solve a set of differential equations, appropriate boundary conditions have to be specified. There can be made the distinction between two kinds of boundaries. Real boundaries can obviously be coastlines, or other structures, but if a computational domain covers a part of the sea or river, then there is also an 'open' boundary which is artificial in the sense that it is not a physical boundary: it is just a line drawn on the map. We will only look at boundary conditions for the 1D SWE, equation (2.41), which means that there is only a left and a right boundary. We assume that the fluid flows from the left to the right, meaning that at the left boundary the flow is directed inwards and at the right boundary directed outwards. The 1D SWE are hyperbolic partial differential equations and the number of boundary conditions that have to be imposed depends on whether the flow is directed in or out of the domain and whether the flow is sub- or supercritical, see Table 2.1. The flow is subcritical if the Froude number, $Fr = |u|/c$

Table 2.1: Number of boundary conditions to be prescribed.

	$Fr < 1$	$Fr > 1$
Inflow	1	2
Outflow	1	0

with $c = \sqrt{gH}$ the celerity, is smaller than one and supercritical if it is larger than one. The Froude number in shallow water flow is the equivalent of the Mach number in aerodynamic

applications. Most shallow water flows are subcritical and Fr rarely exceeds 0.1 in the ocean or 0.8 in rivers. Only in special cases (e.g. flow over dams, rivers with large water surface gradient) supercritical flow can occur, but this will mostly be local. According to Table 2.1, in normal cases one boundary condition must be specified at both boundaries. In the next part of the section this will be explained further.

The number and type of boundary conditions needed in hyperbolic systems is closely related to the behaviour of characteristics. The quasi-linear form of the one-dimensional system (2.41) with (2.43) is

$$\partial_t \mathbf{u} + B\partial_x \mathbf{u} = \mathbf{s}(\mathbf{u}), \tag{2.44}$$

with the Jacobi matrix B given by

$$B = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{bmatrix} 0 & 1 \\ -q^2/H^2 + gH & 2q/H \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{bmatrix}, \tag{2.45}$$

and $c = \sqrt{gH}$ again the celerity. The eigenvalues, λ_i, and corresponding eigenvectors, R_i, of matrix B are given by

$$\lambda_1 = u - c, \qquad R_1 = \frac{H}{2c}\begin{bmatrix} 1 \\ u - c \end{bmatrix}, \tag{2.46}$$

$$\lambda_2 = u + c, \qquad R_2 = \frac{H}{2c}\begin{bmatrix} 1 \\ u + c \end{bmatrix}. \tag{2.47}$$

The matrix R with the eigenvectors (R_1, R_2) as its columns diagonalizes the matrix B as follows

$$R^{-1}BR = \Lambda \quad \Leftrightarrow \quad B = R\Lambda R^{-1}, \tag{2.48}$$

where the diagonal matrix Λ is given by

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \tag{2.49}$$

Multiplying (2.44) by R^{-1} and using (2.48) yields

$$R^{-1}\partial_t \mathbf{u} + \Lambda R^{-1}\partial_x \mathbf{u} = R^{-1}\mathbf{s}. \tag{2.50}$$

This equation can be written in characteristic form

$$\partial_t \mathbf{r} + \Lambda \partial_x \mathbf{r} = \hat{s}, \tag{2.51}$$

where $\hat{s} = R^{-1}\mathbf{s}$ and $\mathbf{r}$ are the characteristic variables, which are constant along the characteristics. Because (2.44) is nonlinear, R is dependent of $\mathbf{u}$ and the characteristic variables $\mathbf{r}$ cannot be computed directly by putting $\mathbf{r} = R^{-1}\mathbf{u}$. However, from the relation

$$R^{-1}\partial_t \mathbf{u} = \partial_t \mathbf{r} \quad \text{or} \quad R^{-1}\partial_x \mathbf{u} = \partial_x \mathbf{r}, \tag{2.52}$$

the characteristic variables can be calculated to be

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} u - 2c \\ u + 2c \end{bmatrix}. \tag{2.53}$$

These characteristic variables $\mathbf{r}$ are the so-called Riemann invariants.

In (2.51) a system of two ordinary differential equations is obtained in terms of the Riemann invariants, which are constant along the characteristics $dx/dt = \lambda_i$ provided $\hat{s} = 0$, see

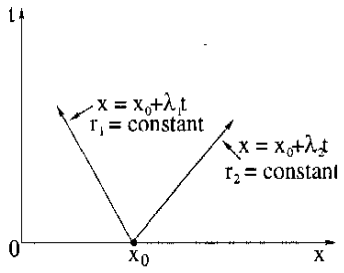


Fig. 2.3: Characteristics through x_0.

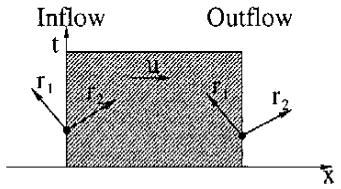


Fig. 2.4: Direction of Riemann invariants for inflow and outflow at a boundary if $Fr < 1$.

Figure 2.3. It is important to know the signs of λ_1 and λ_2 because that determines the direction of the characteristics along which information is transmitted. If the flow is subcritical ($Fr < 1$) then $|u| < c$ and $\lambda_1 < 0$, $\lambda_2 > 0$. The boundary conditions at time t are specified by the Riemann invariants $u \pm 2c$, see Figure 2.4. At both the in- and outflow boundary one of the Riemann invariants must be specified (r_2 for inflow and r_1 for outflow) and the other one must be able to run freely out of the domain. This explains why one boundary condition is needed at both the in- and outflow boundary when the flow is subcritical.

If the flow is supercritical ($Fr > 1$) then $|u| > c$ and both λ_1 and λ_2 have the same sign as u, see Figure 2.5. At the inflow boundary both Riemann invariants must be specified and at

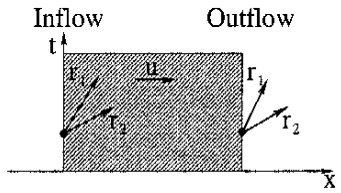


Fig. 2.5: Direction of Riemann invariants for inflow and outflow at a boundary if $Fr > 1$.

the outflow boundary both Riemann invariants must be able to run freely out of the domain, which explains why two boundary conditions are needed at the inflow boundary and zero at the outflow boundary.

Normally there is only limited knowledge of what is happening outside the model region, so in general it is not so easy to find good data to be prescribed on open boundaries. If data is known outside the numerical domain it is most of the times on one of the variables u or H, so the Riemann invariants $u \pm 2c$ are still unknown. In subcritical flows only one Riemann variant has to be specified at the boundaries and the other one is known. The unknown Riemann variant can be derived by manipulating the Riemann variants. For example if H is known (and thus the celerity c) at the domain boundary, $r_2 = u + 2c$ has to be specified at the boundary and $r_1 = u - 2c$ is known, then $r_2 - r_1 = 4c$ which gives $r_2 = r_1 + 4c$. This manipulation is only possible if the solution of the problem does not depend on the boundary conditions, but

unfortunately this is the case in most problems. However, if the boundaries of the domain are chosen far enough away from the region of interest, the possible errors or inaccuracies in the boundary conditions do not reach the region of interest within the time of the total computation.

In shallow areas, parts of the region may dry up at low water levels, which means that there are moving boundaries. In this case $H = 0$ and it is not obvious what will happen, so we will first only look at the case where H remains positive.

Chapter 3

Unstructured Grids

In order to perform numerical calculations of flows, the flow domain has to be divided into small cells. The resulting subdivision is called a grid. Basically a distinction between two types of grids can be made: structured and unstructured grids, see Wendt (1996), Anderson (1995), Wenneker (2002) and Wenneker (2003). The difference between the two grids is explained in Section 3.1 and it also contains the reason why the interest in this thesis is on numerical methods that are applicable to unstructured grids. There are different ways to locate the variables within a grid. The difference between staggered and collocated grids is explained in Section 3.2 and a motivation is given for our choice to use a collocated method on unstructured grids. In Section 3.3 the difference between a cell-centred and a vertex-centred approach is explained, both of which presume a collocated grid and a motivation is given for using the vertex-centred approach.

3.1 Advantages of unstructured grids

Structured grids have the property that at each interior node the same number of cells meet and that (portions of) the grid can be mapped to a rectangle (in 2D) or a block (in 3D). Unstructured grids are obtained by dividing the domain into simple elements with no implied connectivity. In 2D these are often triangles or quadrilaterals and in 3D tetrahedra or prisms. In Figures 3.1 and 3.2 examples of a structured and an unstructured grid are given for the 2D case.

The advantages of unstructured grids compared to structured grids are that the process of grid generation can be automated to a larger extent because of the greater geometric flexibility and that local and adaptive grid refinement is much easier. Unstructured grids also have their disadvantages: the discretization of the flow equations and its implementation becomes more complicated, and execution time is usually larger given a number of unknowns.

In this thesis we only consider problems in one-dimension, but in 1D there is no distinction between structured and unstructured grids, because the only way to divide a line is in intervals. However the methods that are examined have to be in principle applicable to unstructured grids in 2D.

3.2 Staggered versus collocated grids

A distinction can be made between two approaches concerning the location of the variables in the grid for the finite volume and finite difference methods. The variables can be staggered in space, in the sense that the discrete location of the various variables differs, leading to a

Fig. 3.1: Example of a structured grid.

so-called staggered grid. One can argue, see Stelling (1983), that it is beneficial to store the surface elevation in the cell centres, while the velocity components normal to the cell edges are positioned at the midpoints of the cell edges. In a collocated grid however, all variables are located in the same grid location, e.g. in the cell centres or in the vertices. In Figure 3.3 the differences between the two types of grids are depicted for the one-dimensional case and in Figure 3.4 for the two-dimensional case.

We will discuss the advantages and disadvantages of both approaches, because the choice for one of the two has far reaching implications for the discretization and the resulting implementation. The motivation in this thesis is based on the one in Wenneker (2003).

Odd-even decoupling is occurring when a central scheme is used on a collocated grid. The central approximation of the gradient of the surface elevation ($\partial_x \zeta$) decouples the surface elevation in neighbouring points. This may however not be such a big problem on unstructured grids, because on such a grid generally no distinction between 'odd' and 'even' grids can be made.

Fig. 3.2: Example of an unstructured grid.

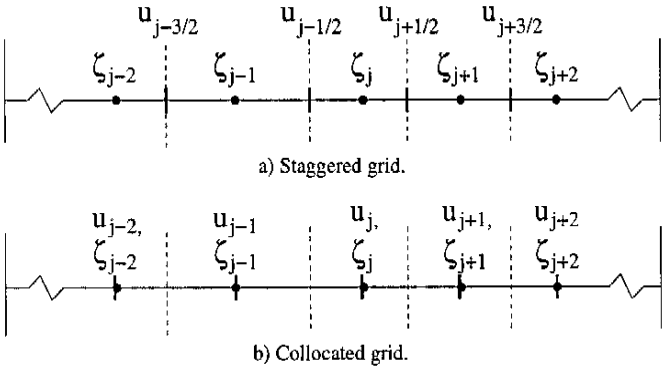


Fig. 3.3: Staggered and collocated grid in 1D.

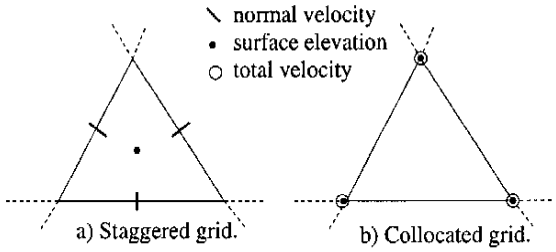


Fig. 3.4: Staggered and collocated grid in 2D.

Higher order approximation of advection According to current standards, second order spatial consistency at least is required for the discretization, including that of the advection term. When using a staggered positioning of the variables, second order accuracy can be achieved on orthogonal structured grids relatively easy, because the relevant velocity components at adjacent

edges are more or less collinear. However, on unstructured staggered grids this is not the case, see Figure 3.4, and second order accuracy is very hard to obtain. Recent work in the group of professor Wesseling at TU Delft confirms this. When a collocated positioning of the variables is used, second order schemes can be constructed on both structured and unstructured grids. An abundance of literature dealing with obtaining second order accuracy on unstructured grids is available, starting with the paper by Barth & Jespersen (1989).

Ease of implementation On unstructured grids, implementation of a staggered scheme is much more complicated than that of a collocated scheme. The difficulty of staggered grids is the fact that there are (at least) two distinct grid locations from which a numerical quadrature has to be set up. This gives the necessity to interpolate back and forth between the various grid locations (especially for the velocity, of which only the components normal to the CV-edges are known), leading to large grid stencils. On a collocated grid, there are no interpolations needed to get the variables at other grid locations. This also makes the implementation of boundary conditions easier. Another advantage of collocated schemes is that for the same operator in different equations the same discretization can be used.

Knowledge present in literature Nowadays only a few publications about a staggered location of the variables on unstructured grids appear and most of the research is on collocated grids.

Conclusion We will use in this thesis a collocated location of the variables, because the implementation is easier and a higher order approximation of advection can be achieved. However we have to take care that no odd-even decoupling occurs. Also since most of the research nowadays is done on collocated grids, using collocated grids avoids time-consuming research that has to be done when using staggered grids.

3.3 Collocated grids: choice between cell-centred and vertex-centred approach

A collocated method can be either of the cell-centred type or the vertex-centred type, depending on whether the flow variables are stored at the centres or at the vertices of the grid cells.

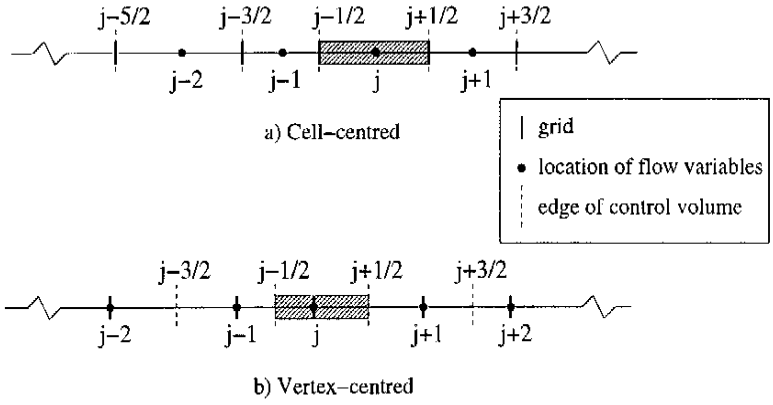
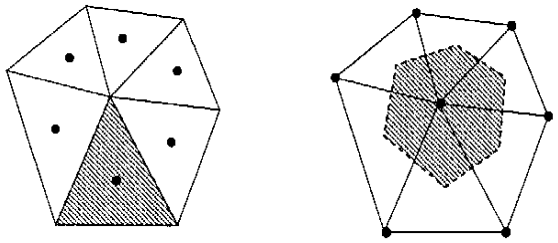


Fig. 3.5: Cell-centred and vertex-centred grid in 1D.



a) Cell–centred b) Vertex–centred

Fig. 3.6: Cell-centred and vertex-centred grid in 2D.

In the cell-centred case, the control volumes coincide with the grid cells and the flow variables are stored in the centre of the grid cells, see Figure 3.5a for 1D and Figure 3.6a for 2D. The values of the variables can be seen as mean values over the cell, at least if the order of accuracy is smaller than three. In this way the cell-centred method is a piecewise constant interpolation in the grid cells in a finite volume sense, see Section 4.2.

In the vertex-centred case, the flow variables are stored in the vertices and the control volumes are chosen in such a way that they fill the whole domain but are non-overlapping. In 1D this can be done by choosing the cell edges exactly in the middle of two neighbouring vertices as is shown in Figure 3.5b. An example in 2D is the the central dual, which is constructed by joining the centroids of the neighbouring cells of a vertex and is depicted in Figure 3.6b. This means that the flow variables do not have to be in the centre of gravity of the control volumes and the control volumes do not coincide with the grid cells, see Figure 3.5b and 3.6b for respectively 1D and 2D. When storing the variables in the vertices a piecewise linear interpolation in the grid cells can be defined in a finite element sense, see Section 4.3.

Although there are various arguments to prefer one over the other, we prefer a vertex-centred positioning of the variables for the following reasons, see also Wenneker (2003), that also hold in 2D and 3D:

- Linear interpolation is defined in a unique manner within a grid cell.
- Evaluation of gradients at control volume edges is easier, because of the uniquely defined linear interpolation.

Chapter 4

Numerical methods

The shallow water equations are nonlinear partial differential equations. These equations can be solved analytically only in a few cases. This means that a numerical method has to be used to solve the SWE in the majority of cases. In the past decades several numerical methods have been developed to solve the SWE. They can roughly be divided in the classes explained in the following subsections. First the finite difference method is discussed in Section 4.1. In Section 4.2 the finite volume method will be explained. After that an outline of the finite element method is given in Section 4.3. Finally there is a brief overview about the spectral method in Section 4.4.

4.1 Finite difference methods

The most common approach to solve the shallow water equations is by the finite difference method (FDM) on a Cartesian grid, see for example Vreugdenhil (1994) and Morton & Mayers (1994). The region of interest is covered by a rectangular grid with grid sizes Δx and Δy, and the spatial derivatives are approximated by finite differences. For example, a central x-difference is computed as

$$\partial_x u \approx \frac{u^n_{j+1} - u^n_{j-1}}{2\Delta x}. \tag{4.1}$$

In reality the region of interest is not a rectangle, so it is difficult using a square grid as the boundaries will not coincide with grid points and more flexible grids are required, for example curvilinear grids, see Figure 3.1. For even more complex geometries unstructured grids are more appropriate, see Section 3, and then discretization is usually done by the finite-volume and finite-element methods described next, because using the FDM becomes very cumbersome.

4.2 Finite volume methods

The finite volume method (FVM) is based on the conservation form of the gouverning equations and is obtained by integrating over finite volumes (or surfaces in 2D and intervals in 1D). These finite volumes are obtained by dividing the domain into non-overlapping control volumes (for example defined by the grid lines). More information about this method can be found in for instance Johnson (1998), Toro (2001), Kulikovskii et al. (2001) and Li et al. (2000). In this section the FVM will be outlined for the 1D case.

Consider the 1D SWE as given in (2.41) with (2.43) in conservation law form

$$\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) = \mathbf{s}(\mathbf{u}). \tag{4.2}$$

By integrating (4.2) over a control volume $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ and using Gauss' theorem the following integral form can be obtained

$$\partial_t \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{u}\, dx + \mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}}, t)) - \mathbf{f}(\mathbf{u}(x_{j-\frac{1}{2}}, t)) = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{s}(\mathbf{u})\, dx. \tag{4.3}$$

For the difference of the places of the control volume boundaries between the cell-centred and the vertex-centred grid, see Section 3.3 and Figure 3.5.

Equation (4.3) is called the integral form of the equation. The solution of the integral form of the problem don't has to be smooth throughout unlike the solution of the differential form (4.2) of the problem, because (4.3) does not contain space derivatives any more. So the integral form must be used if discontinuities are present in the solution.

In this section we will consider a collocated cell-centred grid, where the flow variables are all stored in the centre of the control volumes and the values of the variables can be seen as mean values over the cell. An example of a vertex-centred finite volume method is given in Section 5.1.

Define $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ and

$$U_j = \frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{u}\, dx, \tag{4.4}$$

$$S_j = \frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{s}(\mathbf{u})\, dx. \tag{4.5}$$

Substituting (4.4) and (4.5) in (4.3) the integral form can be written as

$$\partial_t U_j = -\frac{1}{\Delta x_j} \left[\mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}}, t)) - \mathbf{f}(\mathbf{u}(x_{j-\frac{1}{2}}, t)) \right] + S_j. \tag{4.6}$$

It is not evident how to evaluate $\mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}}, t))$, because in the point $x_{j+\frac{1}{2}}$, the vector $\mathbf{u}$ is not uniquely defined, and therefore we replace it by a numerical flux $F_{j+\frac{1}{2}}$. The numerical flux $F_{j+\frac{1}{2}}$ corresponds to the location $x = x_{j+\frac{1}{2}}$, which is the boundary between control volumes $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ and $[x_{j+\frac{1}{2}}, x_{j+1\frac{1}{2}}]$.

Usually the numerical flux is of the form

$$F = F(U_L, U_R), \tag{4.7}$$

where U_L and U_R denote respectively the right and left limits of U at the interfaces where U is discontinuous. In Figure 4.1 a the situation is depicted for the first order case, for example when U_j is a mean, and in Figure 4.1b for the second order case, when U is a piecewise linear function. This numerical flux must have the following properties

i) F is consistent, i.e. $F(U, U) = \mathbf{f}(U)$,

ii) $F(U, V)$ is a non-decreasing function in both of its arguments and

iii) $F(U, V)$ is conservative, i.e. $F(U, V) = -F(V, U)$.

There are a number of numerical fluxes satisfying this properties, such as the Godunov flux, Engquist-Osher flux, Lax-Friedrich flux, HLLC flux and Roe flux. They are all based on solving locally at the control volume boundaries the Riemann problem (see (7.23)), either exact or with an approximation. The HLLC flux will be outlined in 5.2.4 and other numerical fluxes can be found in Toro (1997).

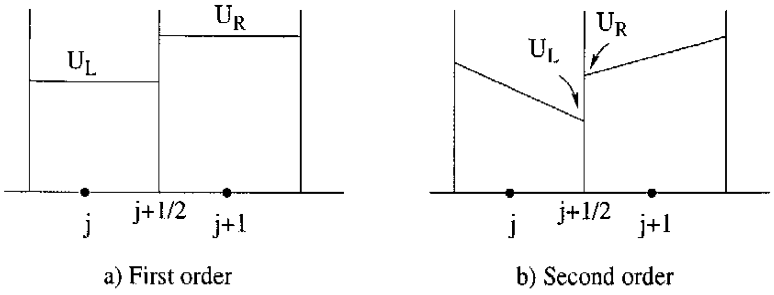


Fig. 4.1: First order and second order approximations.

4.3 Galerkin finite element methods

In the finite element method (FEM) the domain is divided into a number of basic elements (usually) of triangular or quadrilateral form in 2D. In 1D this reduces to dividing the total domain I in subintervals $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ for $j = 1, \ldots, N_{el}$, where N_{el} is the number of elements, see also Figure 4.3. The unknowns are approximated by piecewise smooth functions on each element. In this way the unknowns are defined in the entire region by a sum of piecewise continuous functions. More information about this subject can be found in van der Vegt & Bokhove (2003), Brenner & Scott (1994), Lucquin & Pironneau (1998) and Johnson (1998).

The approximation U of $\mathbf{u}$ is defined as

$$U(x,t) = \sum_{j=0}^{N_{el}} u_{j+\frac{1}{2}}(t)\phi_{j+\frac{1}{2}}(x), \tag{4.8}$$

where the coefficients $u_{j+\frac{1}{2}}$ with $j = 0, \ldots, N_{el}$ are the $(N_{el}+1)$ unknowns and $\phi_{j+\frac{1}{2}}(x)$ are called the basis functions and are determined by the following requirements:

- $\phi_{j+\frac{1}{2}}$ is defined on the whole domain,
- $\phi_{j+\frac{1}{2}}$ is a polynomial of degree k in each element,
- $\phi_{j+\frac{1}{2}} = 1$ in node $x_{j+\frac{1}{2}}$ and zero in all other nodes.

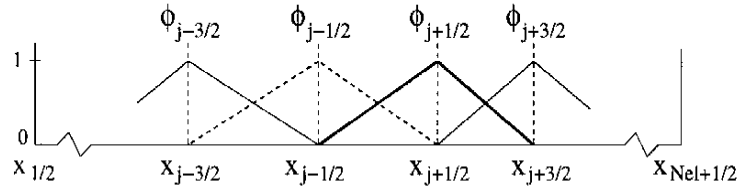


Fig. 4.2: Linear basis functions.

In Figure 4.2 the configuration of linear basis functions ($k = 1$) is depicted and in Figure 4.3 an approximation of $\mathbf{u}$ by linear basis functions can be seen. When using (4.8) U is a piecewise smooth function belonging to the space V_h

$$V_h = \left\{ v \in C^1(I) \quad | \quad v|_{I_j} \in P^k(I_j), \quad j = 1, ..., N_{el} \right\}, \tag{4.9}$$

where $C^1(I)$ is the space of continuous functions on the domain I and $P^k(I_j)$ denotes the space of polynomials in I_j of degree k.

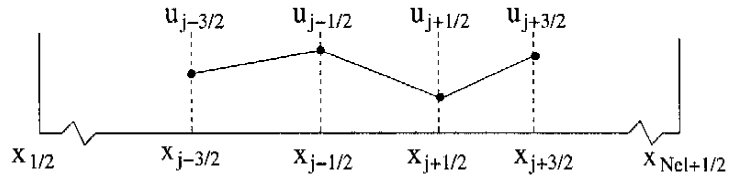


Fig. 4.3: Approximation of $\mathbf{u}$ by linear basis functions.

Next, the equations to be solved, (2.41), are multiplied by a test function ω and integrated over the domain I. This gives the weighted residual formulation

$$\int_I (\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) - \mathbf{s}(\mathbf{u}))\, \omega \, dx = 0. \tag{4.10}$$

The finite element discretization is most easily obtained by splitting the integral in (4.10) into integrals over each element

$$\sum_{j=1}^{N_{el}} \left\{ \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} (\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) - \mathbf{s}(\mathbf{u}))\, \omega \, dx \right\} = 0. \tag{4.11}$$

By integrating by parts the so-called weak formulation is obtained

$$\sum_{j=1}^{N_{el}} \left\{ \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \omega \partial_t \mathbf{u}\, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{f}(\mathbf{u}) \partial_x \omega \, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{s}(\mathbf{u}) \omega \, dx \right.$$
$$\left. + \mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}}))\omega(x_{j+\frac{1}{2}}) - \mathbf{f}(\mathbf{u}(x_{j-\frac{1}{2}}))\omega(x_{j-\frac{1}{2}}) \right\} = 0. \tag{4.12}$$

The important benefit of the weak formulation is that $\mathbf{f}$ now only has to be integrable instead of differentiable. If the solution $\mathbf{u}$ is replaced by its approximation U and we choose $\omega \in V_h$, the fluxes between the cells cancel, because U is continuous at the inter cell boundaries, to yield

$$\sum_{j=1}^{N_{el}} \left\{ \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \omega \partial_t U \, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{f}(U) \partial_x \omega \, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{s}(U) \omega \, dx \right\}$$
$$+ \mathbf{f}(U(x_{N_{el}+\frac{1}{2}}))\omega(x_{N_{el}+\frac{1}{2}}) - \mathbf{f}(U(x_{\frac{1}{2}}))\omega(x_{\frac{1}{2}}) = 0, \tag{4.13}$$

where the last two terms of the left side are called the boundary integrals and they contain the boundary conditions.

The particular choice in the Galerkin method is to choose the test function the same as the basis function, so $\omega = \phi_{j+\frac{1}{2}}$ for $j = 0, \ldots, N_{el}$. In this way only non-zero contributions of the integrals in the elements I_j and I_{j+1} are obtained, because in the other elements $\phi_{j+\frac{1}{2}}$ is zero. This will give for $j = 2, \ldots, N_{el} - 1$, by using (4.8)

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \phi_{j+\frac{1}{2}} \partial_t (u_{j-\frac{1}{2}} \phi_{j-\frac{1}{2}} + u_{j+\frac{1}{2}} \phi_{j+\frac{1}{2}})\, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{f}(U) \partial_x \phi_{j+\frac{1}{2}} \, dx$$
$$- \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{s}(U) \phi_{j+\frac{1}{2}} \, dx + \int_{x_{j+\frac{1}{2}}}^{x_{j+1\frac{1}{2}}} \phi_{j+\frac{1}{2}} \partial_t (u_{j+\frac{1}{2}} \phi_{j+\frac{1}{2}} + u_{j+1\frac{1}{2}} \phi_{j+1\frac{1}{2}})\, dx$$
$$- \int_{x_{j+\frac{1}{2}}}^{x_{j+1\frac{1}{2}}} \mathbf{f}(U) \partial_x \phi_{j+\frac{1}{2}} \, dx - \int_{x_{j+\frac{1}{2}}}^{x_{j+1\frac{1}{2}}} \mathbf{s}(U) \phi_{j+\frac{1}{2}} \, dx = 0. \tag{4.14}$$

Define

$$A^j_{n,m} = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \phi_n \phi_m \, dx, \tag{4.15}$$

$$B^j_m = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathrm{f}(U) \partial_x \phi_m \, dx, \tag{4.16}$$

$$C^j_m = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathrm{s}(U) \phi_m \, dx, \tag{4.17}$$

where the integrals B^j_m and C^j_m can be approximated by a quadrature formula. Substituting (4.15), (4.16) and (4.17) in (4.14) yields

$$A^j_{j+\frac{1}{2},j-\frac{1}{2}} \partial_t u_{j-\frac{1}{2}} + (A^j_{j+\frac{1}{2},j+\frac{1}{2}} + A^{j+1}_{j+\frac{1}{2},j+\frac{1}{2}}) \partial_t u_{j+\frac{1}{2}} + A^{j+1}_{j+\frac{1}{2},j+1\frac{1}{2}} \partial_t u_{j+1\frac{1}{2}} = B^j_{j+\frac{1}{2}} + B^{j+1}_{j+\frac{1}{2}} + C^j_{j+\frac{1}{2}} + C^{j+1}_{j+\frac{1}{2}}. \tag{4.18}$$

For $j = 1$ and $j = N_{el}$ the boundary integrals also have to be taken into account and extra terms on the righthand side of the equations are obtained, $\mathrm{f}(u_{\frac{1}{2}})$ for $j = 1$ and $\mathrm{f}(u_{N_{el}+\frac{1}{2}})$ for $j = N_{el}$. In this way a system of $(N_{el}+1)$ ordinary differential equations is obtained, which can be written into a tri-diagonal matrix system of the form

$$M(\mathbf{u}_h) \partial_t \mathbf{u}_h = R_h(\mathbf{u}_h), \tag{4.19}$$

where $\mathbf{u}_h$ is the vector containing the $(N_{el}+1)$ coefficients $u_{j+\frac{1}{2}}$ with $j = 0, \ldots, N_{el}$, M is a $(N_{el}+1) \times (N_{el}+1)$ tri-diagonal matrix, called the mass-matrix, consisting of the coefficients $A^j_{n,m}$ and $R_h(\mathbf{u}_h)$ a vector of length $(N_{el}+1)$ containing the right hand side of equation (4.18). The time-discretization is usually performed by an implicit method, because this allows for bigger time steps. An implicit method also implies that a matrix system has to be solved, which means that normally more calculations have to be carried out. But (4.19) is already a matrix system, so implementing an implicit method does not increase the computational cost very much.

4.4 Spectral methods

The starting point for the spatial discretization in the spectral methods is again the weak formulation, (4.12). Instead of local basis-functions, global basis-functions are defined, such as Fourier series and several types of orthogonal polynomials (for example Legendre and Chebyshev polynomials). With a proper choice of basis functions spectral methods are very accurate when the solution is sufficiently smooth. They are however difficult to apply in domains with a complicated shape and for general boundary conditions. If for example Fourier series are used, then by definition periodic functions are produced. This works well for periodic boundary conditions, but for general boundary conditions artificial tricks have to be carried out. There exist also hybrid methods which combine the advantages of spectral and finite element methods, the so-called spectral element methods. In these methods the computational domain is divided into elements as in FEM to be able to deal with more complex boundaries and on each element global basis functions are defined as in the spectral method. More general information on spectral methods can be found in Johnson (1998) and Vreugdenhil (1994).

Chapter 5

Examined Numerical methods

In this chapter we describe in detail the two numerical methods that are compared. We choose a finite volume method and a finite element method, because of their ease of application on unstructured grids. The first method is the Collocated Coupled Solution Method (CCSM) and is described in Section 5.1. This is a finite volume method using a collocated cell-centred grid. The second method is described in Section 5.2 and is the Runge-Kutta Discontinuous Galerkin method, which is a finite element method using a discontinuous approximation of the variables.

5.1 Collocated Coupled Solution Method

The Collocated Coupled Solution Method (CCSM) is a collocated finite volume method in which a linear approximation of the primary variables is used, while the solution procedure is based on solving the coupled system of equations at once. The advantages of this scheme lie in its (relative) simplicity and its (expected) accuracy and efficiency. It is a semi-implicit method which allows for relative large time-steps. The disadvantage of the scheme is that, as far as we know, such a scheme has never been developed yet for free surface flows. The method is based on Wenneker (2003), where we will only consider the 1D case in this report. In Section 5.1.1 we will start with the discretization of the equations using a finite volume method and a linear approximation of the variables. After that we will deal in Section 5.1.2 with the conditions at the boundaries. Assembly of the discretized equations, which results in a formulation in terms of matrix algebra, is presented in Section 5.1.3. Finally, in Section 5.1.4, a semi-implicit time discretization using a coupled matrix system will be introduced.

5.1.1 Discretization of the 1D SWE

The finite volume method examined in this masters thesis will be based on the 1D SWE in the form of equations (2.41) with (2.42) and $q = uH$:

$$\partial_t \zeta + \partial_x q = 0, \tag{5.1a}$$

$$\partial_t q + \partial_x (uq) = -g(\zeta + h)\partial_x \zeta, \tag{5.1b}$$

where we used the fact that $H = \zeta + h$ (see equation (2.10)) and $\partial_t H = \partial_t \zeta$, because the bottom h is assumed not to change in time. The domain will be divided in N_{el} control volumes (CV's) $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ with $j = 1, ..., N_{el}$ and the variables are stored in the grid points x_j. The grid is collocated and vertex-centred, see sections 3.2 and 3.3, and is depicted in Figure 5.1. The location of the cell-edges $x_{j+\frac{1}{2}}$ will be exactly in the middle of the two neighbouring vertices as shown in Figure 3.5b. In spite of the fact that we use a vertex-centred grid, the approach at the

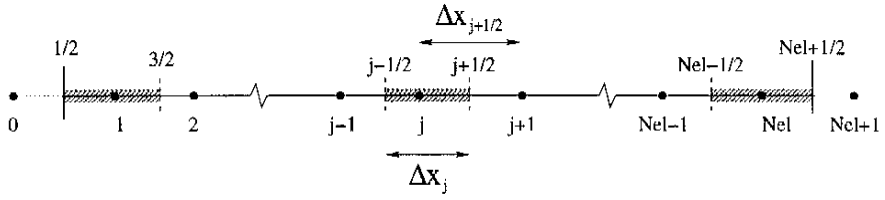


Fig. 5.1: The grid for the Collocated Coupled Solution Method. CV j has size Δx_j and $\Delta x_{j+\frac{1}{2}}$ denotes the size of cell $j+\frac{1}{2}$.

boundaries resembles the one of a cell-centred grid. The boundaries of the domain coincide with the points $x_{\frac{1}{2}}$ and $x_{N_{el}+\frac{1}{2}}$ and for this reason we introduce the fictitious points x_0 and $x_{N_{el}+1}$. The treatment of conditions at the boundaries will be given in Section 5.1.2.

The variables ζ and q are approximated in each cell by a continuous piecewise-linear function of the form

$$\psi(x) = \frac{(x - x_j)\psi_{j+1} + (x_{j+1} - x)\psi_j}{\Delta x_{j+\frac{1}{2}}}, \qquad \psi = \{\zeta, q\}, \tag{5.2}$$

for $x \in [x_j, x_{j+1}]$ and $\Delta x_{j+\frac{1}{2}} = x_{j+1} - x_j$. So $\psi(x_j) = \psi_j$ and $\psi(x_{j+1}) = \psi_{j+1}$ are the values in the vertices. The bottom h is approximated in the same way and this causes the total water depth $H = \zeta + h$ to be linear as well.

Integrating equations (5.1a) and (5.1b) over each control volume $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ with $j = 1, \ldots, N_{el}$ yields

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_t \zeta \, dx + \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x q \, dx = 0, \tag{5.3}$$

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_t q \, dx + \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x \left(uq\right) \, dx = - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} g(\zeta + h) \partial_x \zeta \, dx. \tag{5.4}$$

Subsequently we replace in (5.3) and (5.4) the variables ζ and q by their linear approximations.

The integrated time derivative of $\psi = \{\zeta, q\}$, appearing in (5.3) and (5.4), can be computed by using (5.2) and gives

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_t \psi(x) \, dx = \frac{d}{dt} \left[\tfrac{1}{8} \Delta x_{j-\frac{1}{2}} \psi_{j-1} + \tfrac{6}{8} \Delta x_j \psi_j + \tfrac{1}{8} \Delta x_{j+\frac{1}{2}} \psi_{j+1} \right], \tag{5.5}$$

where $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$, see Figure 5.1.

For uniform grids this becomes

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_t \psi(x) \, dx = \Delta x \frac{d}{dt} \left[\tfrac{1}{8} \psi_{j-1} + \tfrac{6}{8} \psi_j + \tfrac{1}{8} \psi_{j+1} \right]. \tag{5.6}$$

The second integral of equation (5.3) can be calculated in two different ways.

1. The central approach in the continuity equation is using a central scheme for q, which gives

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x q \, dx = q_{j+\frac{1}{2}} - q_{j-\frac{1}{2}}, \tag{5.7}$$

where we use linear interpolation to find the values of q at the control volume edges:

$$q_{j+\frac{1}{2}} = \tfrac{1}{2}\left(q_j + q_{j+1}\right). \tag{5.8}$$

With this (5.7) becomes

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x q \, dx = \tfrac{1}{2}\left(q_{j+1} - q_{j-1}\right). \tag{5.9}$$

2. The upwind approach in the continuity equation is obtained by using the fact that $q = uH = u(\zeta + h)$. This can be substituted in (5.7) to give

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x q \, dx = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x \left[u(\zeta + h)\right] dx = \left[u(\zeta + h)\right]_{j+\frac{1}{2}} - \left[u(\zeta + h)\right]_{j-\frac{1}{2}}. \tag{5.10}$$

We use first order upwind to compute the values of $(\zeta + h)$ at the CV-edges:

$$(\zeta + h)_{j+\frac{1}{2}} = \begin{cases} (\zeta + h)_j & \text{if } u_{j+\frac{1}{2}} > 0, \\ (\zeta + h)_{j+1} & \text{if } u_{j+\frac{1}{2}} < 0, \end{cases} \tag{5.11}$$

and u is interpolated as follows

$$u_{j+\frac{1}{2}} = \frac{q_{j+\frac{1}{2}}}{(\zeta + h)_{j+\frac{1}{2}}} = \frac{q_{j+1} + q_j}{(\zeta + h)_{j+1} + (\zeta + h)_j}. \tag{5.12}$$

In this way $(\zeta + h)$ is dependent of u in (5.11), but u itself is depending on $(\zeta + h)$ too, because of (5.12). In practice this is overcome by using Picard linearization, which means that u is evaluated at the old time level (so this value is known) and $(\zeta + h)$ at the new time level or using an iterative process, see Section 5.1.4. Because of the use of a first order upwind scheme the total scheme will only be first order accurate in space.

Note that we do not use a linear interpolation of u of the form

$$u_{j+\frac{1}{2}} = \tfrac{1}{2}(u_j + u_{j+1}) = \tfrac{1}{2}\left(\frac{q_j}{(\zeta + h)_j} + \frac{q_{j+1}}{(\zeta + h)_j}\right), \tag{5.13}$$

because this introduces an extra linear approximation for u as opposed to (5.12) and we expect that the discretization becomes more accurate if we use a minimum of approximations.

Using (5.11) in (5.10) gives

$$\begin{aligned} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x \left[u(\zeta + h)\right] dx = \tfrac{1}{2}\Big[&-\left(|u_{j-\frac{1}{2}}| + u_{j-\frac{1}{2}}\right)(\zeta + h)_{j-1} \\ &+ \left(|u_{j-\frac{1}{2}}| - u_{j-\frac{1}{2}} + |u_{j+\frac{1}{2}}| + u_{j+\frac{1}{2}}\right)(\zeta + h)_j - \left(|u_{j+\frac{1}{2}}| - u_{j+\frac{1}{2}}\right)(\zeta + h)_{j+1}\Big]. \end{aligned} \tag{5.14}$$

Working out the momentum advection term, i.e. the second integral of equation (5.4), yields

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x(uq)\,dx = (uq)_{j+\frac{1}{2}} - (uq)_{j-\frac{1}{2}}. \tag{5.15}$$

We use again first order upwind to compute the values of q at the CV-edges:

$$q_{j+\frac{1}{2}} = \begin{cases} q_j & \text{if } u_{j+\frac{1}{2}} > 0, \\ q_{j+1} & \text{if } u_{j+\frac{1}{2}} < 0, \end{cases} \tag{5.16}$$

and use the interpolation for u as given in (5.12). Again the nonlinearity is overcome in practice by using Picard linearization or an iterative process. Substituting (5.16) in (5.15) gives

$$\begin{aligned}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x(uq)\,dx = \tfrac{1}{2}\Big[&-\left(|u_{j-\frac{1}{2}}| + u_{j-\frac{1}{2}}\right) q_{j-1} \\ &+ \left(|u_{j-\frac{1}{2}}| - u_{j-\frac{1}{2}} + |u_{j+\frac{1}{2}}| + u_{j+\frac{1}{2}}\right) q_j - \left(|u_{j+\frac{1}{2}}| - u_{j+\frac{1}{2}}\right) q_{j+1}\Big]. \end{aligned} \tag{5.17}$$

The only integral left is the one on the right hand side of (5.4), which can be computed as follows

$$\begin{aligned}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} g(\zeta+h)\partial_x\zeta\,dx &= g\left[\partial_x\zeta\right]_{x_{j-\frac{1}{2}}}^{x_j} \int_{x_{j-\frac{1}{2}}}^{x_j} (\zeta+h)\,dx + g\left[\partial_x\zeta\right]_{x_j}^{x_{j+\frac{1}{2}}} \int_{x_j}^{x_{j+\frac{1}{2}}} (\zeta+h)\,dx \\ &= g\frac{\zeta_j - \zeta_{j-1}}{\Delta x_{j-\frac{1}{2}}} \int_{x_{j-\frac{1}{2}}}^{x_j} \frac{(x-x_{j-1})(\zeta+h)_j + (x_j - x)(\zeta+h)_{j-1}}{\Delta x_{j-\frac{1}{2}}}\,dx \\ &\quad + g\frac{\zeta_{j+1} - \zeta_j}{\Delta x_{j+\frac{1}{2}}} \int_{x_j}^{x_{j+\frac{1}{2}}} \frac{(x-x_j)(\zeta+h)_{j+1} + (x_{j+1} - x)(\zeta+h)_j}{\Delta x_{j+\frac{1}{2}}}\,dx \\ &= \frac{1}{8}g\left[-\left\{(\zeta+h)_{j-1} + 3(\zeta+h)_j\right\}\zeta_{j-1} + \left\{(\zeta+h)_{j-1} - (\zeta+h)_{j+1}\right\}\zeta_j\right. \\ &\quad \left. + \left\{3(\zeta+h)_j + (\zeta+h)_{j+1}\right\}\zeta_{j+1}\right], \end{aligned} \tag{5.18}$$

where we utilized in the first step the fact that $\partial_x\zeta$ is constant in each cell (a consequence of the presumed linearity of ζ).
By using the discretization given in (5.18) it is possible that a so-called odd-even decoupling occurs. If we look at the special situation where $(\zeta+h)_j = H$ and $(\zeta+h)_{j\pm1} = H+\delta$, with $H > 0$ and δ constant, as depicted in Figure 5.2 for the case with a constant bottom level h, then (5.18) becomes

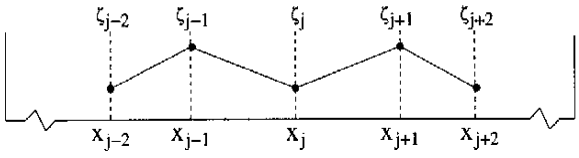


Fig. 5.2: Odd-even decoupling.

$$\frac{1}{8}g\left[-(4H+\delta)\zeta_{j-1} + (4H+\delta)\zeta_{j+1}\right]. \tag{5.19}$$

This equation is equal to zero when $\zeta_{j-1} = \zeta_{j+1}$, for example as in the situation depicted in Figure 5.2, and does not involve ζ_j. But if we look at the left-hand side of (5.18), then the

only time this exact integral can become zero is when $\partial_x \zeta = 0$, that is when the water level is constant and this is definitely not the case with the situation as shown in Figure 5.2. This means that the numerical calculation of the integral yields a value equal to zero even though the water level shows so-called $2\Delta x$-waves.

5.1.2 Boundary conditions

In the previous section we derived two discrete equations for each control volume. This means that we have $2N_{el}$ equations for the $2(N_{el}+2)$ unknowns in the grid points. The other 4 equations will be obtained by deriving equations for the conditions at the boundaries, which are located at the points $x_{\frac{1}{2}}$ and $x_{N_{el}+\frac{1}{2}}$. Due to the linear approximation of the variables we use the following interpolation

$$\left.\begin{aligned} \psi_{\frac{1}{2}} &= \tfrac{1}{2}\left(\psi_0 + \psi_1\right), \\ \psi_{N_{el}+\frac{1}{2}} &= \tfrac{1}{2}\left(\psi_{N_{el}} + \psi_{N_{el}+1}\right), \end{aligned}\right\} \qquad \psi = \{\zeta, q\}. \tag{5.20}$$

We will only look at transmissive and Dirichlet boundary conditions.

1. Transmissive boundaries are obtained by the following relation

$$\psi_0 = \psi_1, \tag{5.21a}$$
$$\psi_{N_{el}+1} = \psi_{N_{el}}, \tag{5.21b}$$

   with $\psi = \{\zeta, q\}$.

2. Dirichlet boundary conditions are of the form

$$\psi_{\frac{1}{2}} = \psi_{\text{giv}}^{\frac{1}{2}}(t), \tag{5.22a}$$
$$\psi_{N_{el}+\frac{1}{2}} = \psi_{\text{giv}}^{N_{el}+\frac{1}{2}}(t), \tag{5.22b}$$

   with $\psi = \{\zeta, q\}$, and $\psi_{\text{giv}}^{\frac{1}{2}}$ and $\psi_{\text{giv}}^{N_{el}+\frac{1}{2}}$ given functions of t.

By using (5.20) we can write (5.21) and (5.22) in generic form as follows

$$\psi_0 + (1 - 2a_\psi^1)\psi_1 = 2\psi_{\text{giv}}^{\frac{1}{2}}(t), \tag{5.23a}$$
$$\psi_{N_{el}+1} + (1 - 2a_\psi^{N_{el}})\psi_{N_{el}} = 2\psi_{\text{giv}}^{N_{el}+\frac{1}{2}}(t), \tag{5.23b}$$

with $\psi = \{\zeta, q\}$. A transmissive boundary is obtained by putting $a_\psi = 1$ and $\psi_{\text{giv}}(t) = 0$ and a Dirichlet boundary is obtained by putting $a_\psi = 0$ and $\psi_{\text{giv}}(t)$ the Dirichlet boundary condition. To be able to write all the equations in matrix form, as will be done in the following section, (5.23) are differentiated in time to yield

$$\tfrac{d}{dt}\psi_0 + (1 - 2a_\psi^1)\tfrac{d}{dt}\psi_1 = 2\tfrac{d}{dt}\psi_{\text{giv}}^{\frac{1}{2}}(t), \tag{5.24a}$$
$$\tfrac{d}{dt}\psi_{N_{el}+1} + (1 - 2a_\psi^{N_{el}})\tfrac{d}{dt}\psi_{N_{el}} = 2\tfrac{d}{dt}\psi_{\text{giv}}^{N_{el}+\frac{1}{2}}(t), \tag{5.24b}$$

Now we have as many equations as unknowns and the system can be solved.

5.1.3 Matrix formulation

In the previous sections we derived a set ordinary differential equations (ODE's) for each control volume. We will write this system of coupled ODE's in matrix form, because a matrix formulation is more succinct. The matrix system is given by

$$M_\zeta \tfrac{d}{dt}\boldsymbol{\zeta} + D\boldsymbol{\Phi} = R_\zeta, \tag{5.25a}$$

$$M_q \tfrac{d}{dt}\mathbf{q} + C\mathbf{q} = -G\boldsymbol{\zeta} + R_q, \tag{5.25b}$$

where $\boldsymbol{\zeta}$ and $\mathbf{q}$ are vectors of length $(N_{el}+2)$ containing the values in the vertices $0, .., N_{el}+1$ and Φ depends on the approach chosen to discretize the advection term in the continuity equation. When using the central approach, given in (5.9), $\boldsymbol{\Phi}$ is equal to $\mathbf{q}$ and in the upwind approach, given in (5.14), $\boldsymbol{\Phi}$ is equal to $\boldsymbol{\zeta}+\mathbf{h}$, with $\mathbf{h}$ the vector of length $(N_{el}+2)$ containing the values for the bottom level in the vertices. The equations representing the particular boundary conditions are processed in the matrices M_ζ and M_q and the vectors R_ζ and R_q.

The tri-diagonal $(N_{el}+2)\times(N_{el}+2)$ mass matrices M_ζ and M_q are of the form

$$M_\psi = \begin{bmatrix} 1 & 1-2a_\psi^1 & & & & \\ \frac{1}{8}\Delta x_{\frac{1}{2}} & \frac{6}{8}\Delta x_1 & \frac{1}{8}\Delta x_{1\frac{1}{2}} & & \emptyset & \\ & \frac{1}{8}\Delta x_{1\frac{1}{2}} & \frac{6}{8}\Delta x_2 & \frac{1}{8}\Delta x_{2\frac{1}{2}} & & \\ & & \ddots & \ddots & \ddots & \\ & \emptyset & & \frac{1}{8}\Delta x_{N_{el}-\frac{1}{2}} & \frac{6}{8}\Delta x_{N_{el}} & \frac{1}{8}\Delta x_{N_{el}+\frac{1}{2}} \\ & & & & 1-2a_\psi^{N_{el}} & 1 \end{bmatrix}, \tag{5.26}$$

for $\psi = \{\zeta, q\}$, and can be deduced from (5.5) and the boundary conditions in (5.24).

The vectors R_ζ and R_q of length $N_{el}+2$ are of the form

$$R_\psi = 2\begin{bmatrix} \frac{d}{dt}\psi_{\text{giv}}^{\frac{1}{2}} \\ 0 \\ \vdots \\ 0 \\ \frac{d}{dt}\psi_{\text{giv}}^{N_{el}+\frac{1}{2}} \end{bmatrix}, \qquad \psi = \{\zeta, q\}, \tag{5.27}$$

as derived from the boundary conditions in (5.24).

From (5.17) the tri-diagonal $(N_{el}+2)\times(N_{el}+2)$ advection matrix C can be found to yield

$$C = \frac{1}{2}\begin{bmatrix} 0 & & & & & \\ c_1^1 & c_1^2 & c_1^3 & & \emptyset & \\ & c_2^1 & c_2^2 & c_2^3 & & \\ & & \ddots & \ddots & \ddots & \\ & \emptyset & & c_{N_{el}}^1 & c_{N_{el}}^2 & c_{N_{el}}^3 \\ & & & & & 0 \end{bmatrix}, \tag{5.28}$$

with

$$\begin{aligned} c_j^1 &= -|u_{j-\frac{1}{2}}| - u_{j-\frac{1}{2}}, \\ c_j^2 &= |u_{j-\frac{1}{2}}| - u_{j-\frac{1}{2}} + |u_{j+\frac{1}{2}}| + u_{j+\frac{1}{2}}, \\ c_j^3 &= -|u_{j+\frac{1}{2}}| + u_{j+\frac{1}{2}}. \end{aligned}$$

The sum of row j of the matrix C is equal to $u_{j+\frac{1}{2}} - u_{j-\frac{1}{2}}$, except for the first and last row. The boundary conditions have been taken care of in (5.26) and (5.27). This means that if $\mathbf{q}$ is a vector with constant coefficients then $C\mathbf{q}$ gives the vector with coefficients $q(u_{j+\frac{1}{2}} - u_{j-\frac{1}{2}})$ for $j = 1, ..., N_{el}$. The reason for this is that $C\mathbf{q}$ calculates the integral of $\partial_x(uq)$ over each control volume. If q is constant then it can be taken out of the integral and this gives

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x(uq)\, dx = q \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_x u\, dx = q(u_{j+\frac{1}{2}} - u_{j-\frac{1}{2}}). \tag{5.29}$$

If u is also constant then the sum of each row of C is equal to zero and thus $C\mathbf{q}$ becomes equal to zero. This is what we expect, because if we calculate the integral of $\partial_x(uq)$ when u and q are constant, this will have zero as a result.

The tri-diagonal $(N_{el}+2) \times (N_{el}+2)$ matrix D depends on the approach chosen to discretize the advection term in the continuity equation. For the central approach, the divergence operator D is of the form

$$D = \frac{1}{2} \begin{bmatrix} 0 & 0 & & & & \\ -1 & 0 & 1 & & \emptyset & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & \emptyset & & -1 & 0 & 1 \\ & & & & 0 & 0 \end{bmatrix}, \tag{5.30}$$

which can be seen from equation (5.9). Note that the sum of each row of the matrix D is equal to zero. This means that if $\mathbf{q}$ is a vector with constant coefficients (remember that $\mathbf{\Phi} = \mathbf{q}$ in the first approach) then $D\mathbf{q}$ is equal to zero. This is caused by the fact that $D\mathbf{q}$ calculates the integral of $\partial_x q$. If q is constant in space, then the spatial derivative must be zero and hence its integral. For the upwind approach, given in (5.14), the matrix D is equal to the advection matrix C.

The tri-diagonal $(N_{el}+2) \times (N_{el}+2)$ gradient operator G is defined as follows

$$G = \frac{1}{8} g \begin{bmatrix} 0 & & & & & \\ G_1^1 & G_1^2 & G_1^3 & & \emptyset & \\ & G_2^1 & G_2^2 & G_2^3 & & \\ & & \ddots & \ddots & \ddots & \\ & \emptyset & & G_{N_{el}}^1 & G_{N_{el}}^2 & G_{N_{el}}^3 \\ & & & & & 0 \end{bmatrix}, \tag{5.31}$$

with

$$\begin{aligned} G_j^1 &= -(\zeta + h)_{j-1} - 3(\zeta + h)_j, \\ G_j^2 &= (\zeta + h)_{j-1} - (\zeta + h)_{j+1}, \\ G_j^3 &= 3(\zeta + h)_j + (\zeta + h)_{j+1}. \end{aligned}$$

as derived from (5.18). For a reason similar to that for matrix D and C the sum of each row of G is equal to zero.

5.1.4 Time integration

Equations (5.25a) and (5.25b) are integrated in time, using the θ-method. If $\{t^n\}_{n=0}^N$ is a partition of $[0, T]$, with T the total computation time, and $\Delta t^n = t^{n+1} - t^n$ for $n = 0, ..., N-1$, the

time discretization is as follows:

$$M_\zeta \frac{\boldsymbol{\zeta}^{n+1} - \boldsymbol{\zeta}^n}{\Delta t^n} + \theta_d D^{n+1}\boldsymbol{\Phi}^{n+1} + (1-\theta_d) D^n \boldsymbol{\Phi}^n = R_\zeta^n, \tag{5.32a}$$

$$M_q \frac{\mathbf{q}^{n+1} - \mathbf{q}^n}{\Delta t^n} + \theta_c C^{n+1}\mathbf{q}^{n+1} + (1-\theta_c) C^n \mathbf{q}^n = -\theta_g G^{n+1}\boldsymbol{\zeta}^{n+1} - (1-\theta_g) G^n \boldsymbol{\zeta}^n + R_q^n. \tag{5.32b}$$

with θ_c, θ_d and θ_g given constants between zero and one. If $\theta_i = 0$ for $i = \{d, c, g\}$, the time integration of the corresponding term is explicit and if $\theta_i = 1$, it is fully implicit. Note that we have the freedom to choose different values for the various θ, but in this thesis only equal values for the θ's are used. The superscripts n and $n+1$ indicate at what time level a quantity is evaluated.

The vectors R_ζ^n and R_q^n are obtained by integrating (5.27) in time and are given by

$$R_\psi^n = \frac{2}{\Delta t^n} \begin{bmatrix} \psi_{\text{giv}}^{\frac{1}{2}}(t^{n+1}) - \psi_{\text{giv}}^{\frac{1}{2}}(t^n) \\ 0 \\ \vdots \\ 0 \\ \psi_{\text{giv}}^{N_{el}+\frac{1}{2}}(t^{n+1}) - \psi_{\text{giv}}^{N_{el}+\frac{1}{2}}(t^n) \end{bmatrix}, \qquad \psi = \{\zeta, q\}. \tag{5.33}$$

If we put all the unknown variables in (5.32) at time level $n+1$ on the left-hand side and the known variables at time level n on the right-hand side we get the following system

$$\frac{1}{\Delta t^n} M_\zeta \boldsymbol{\zeta}^{n+1} + \theta_d D^{n+1}\boldsymbol{\Phi}^{n+1} = \frac{1}{\Delta t^n} M_\zeta \boldsymbol{\zeta}^n - (1-\theta_d) D^n \boldsymbol{\Phi}^n + R_\zeta^n, \tag{5.34a}$$

$$\begin{aligned} \frac{1}{\Delta t^n} M_q \mathbf{q}^{n+1} + \theta_c C^{n+1}\mathbf{q}^{n+1} + \theta_g G^{n+1}\boldsymbol{\zeta}^{n+1} = \frac{1}{\Delta t^n} M_q \mathbf{q}^n - (1-\theta_c) C^n \mathbf{q}^n \\ - (1-\theta_g) G^n \boldsymbol{\zeta}^n + R_q^n. \end{aligned} \tag{5.34b}$$

This system can also be written as a coupled matrix system in the following way

$$M_{\text{imp}} \mathbf{U}^{n+1} = M_{\text{exp}} \mathbf{U}^n + R^n \tag{5.35}$$

with

$$M_{\text{imp}} = \frac{1}{\Delta t^n} \begin{bmatrix} M_\zeta & \theta_d \Delta t^n D \\ \theta_g \Delta t^n G^{n+1} & M_q + \theta_c \Delta t^n C^{n+1} \end{bmatrix}, \qquad \mathbf{U} = \begin{bmatrix} \boldsymbol{\zeta} \\ \mathbf{q} \end{bmatrix},$$

$$M_{\text{exp}} = \frac{1}{\Delta t^n} \begin{bmatrix} M_\zeta & (\theta_d - 1)\Delta t^n D \\ (\theta_g - 1)\Delta t^n G^n & M_q + (\theta_c - 1)\Delta t^n C^n \end{bmatrix}, \quad R^n = \begin{bmatrix} R_\zeta^n \\ R_q^n \end{bmatrix},$$

for the central approach in the continuity equation. Except for the value of Δt^n, the only things that change in matrices M_{imp} and M_{exp} in each time step are the matrices G and C.

For the upwind approach in the continuity equation the matrices M_{imp}, M_{exp} and the vector

R^n are given by

$$M_{\text{imp}} = \frac{1}{\Delta t^n}\begin{bmatrix} M_\zeta + \theta_d \Delta t^n D^{n+1} & 0 \\ \theta_g \Delta t^n G^{n+1} & M_q + \theta_c \Delta t^n C^{n+1} \end{bmatrix}, \tag{5.36}$$

$$M_{\text{exp}} = \frac{1}{\Delta t^n}\begin{bmatrix} M_\zeta + (\theta_d - 1)\,\Delta t^n D^n & 0 \\ (\theta_g - 1)\,\Delta t^n G^n & M_q + (\theta_c - 1)\,\Delta t^n C^n \end{bmatrix}, \tag{5.37}$$

$$R^n = \begin{bmatrix} R_\zeta^n - \left(\theta_d D^{n+1} + (1-\theta_d) D^n\right) h \\ R_q^n \end{bmatrix},$$

and this time also the matrix D changes in each time step. The vector R^n does in this case also include the matrix D at time t^{n+1}, but this is solved by using Picard linearization or an iterative process as is explained later.

We can rearrange system (5.35) and write it in block-tridiagonal form. This can be obtained by rearranging the vectors $\mathbf{U}$ and R^n in the following way

$$\widetilde{\mathbf{U}} = \begin{bmatrix} \zeta_0 \\ q_0 \\ \zeta_1 \\ q_1 \\ \vdots \\ \zeta_{N_{el}} \\ q_{N_{el}} \\ \zeta_{N_{el}+1} \\ q_{N_{el}+1} \end{bmatrix}, \quad \widetilde{R}^n = \begin{bmatrix} \widetilde{R}^n_{\zeta,0} \\ \widetilde{R}^n_{q,0} \\ \widetilde{R}^n_{\zeta,1} \\ \widetilde{R}^n_{q,1} \\ \vdots \\ \widetilde{R}^n_{\zeta,N_{el}} \\ \widetilde{R}^n_{q,N_{el}} \\ \widetilde{R}^n_{\zeta,N_{el}+1} \\ \widetilde{R}^n_{q,N_{el}+1} \end{bmatrix},$$

where $\widetilde{R}^n_\zeta$ is equal to the first half of R^n and $\widetilde{R}^n_q$ to the second half. Then system (5.34) can be written as a block-tridiagonal system

$$M_B \widetilde{\mathbf{U}}^{n+1} = M_b \widetilde{\mathbf{U}}^n + \widetilde{R}^n, \tag{5.38}$$

with

$$M_B = M^{n+1}(\theta_d, \theta_g, \theta_c), \qquad M_b = M^n((\theta_d - 1), (\theta_g - 1), (\theta_c - 1)). \tag{5.39}$$

For notational purposes we introduce the block-tridiagonal matrix $M^m(\alpha, \beta, \gamma)$, where we have in (5.39) $m = n$ or $m = n+1$, $\alpha = \theta_d$ or $\alpha = \theta_d - 1$ and similar relations for β and γ. The matrix $M^m(\alpha, \beta, \gamma)$ is defined as follows:

$$M^m(\alpha, \beta, \gamma) = \begin{bmatrix} A_0 & B_0 & & & & \\ C_1 & A_1 & B_1 & & \emptyset & \\ & C_2 & A_2 & B_2 & & \\ & & \ddots & \ddots & \ddots & \\ & \emptyset & & C_{N_{el}} & A_{N_{el}} & B_{N_{el}} \\ & & & & C_{N_{el}+1} & A_{N_{el}+1} \end{bmatrix} \tag{5.40}$$

with the values of A_j, B_j and C_j depending on the approach used to calculate the advection term in the mass equation. The central approach will give

$$A_0 = \frac{1}{\Delta t^n}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_0 = \frac{1}{\Delta t^n}\begin{bmatrix} 1-2a_\zeta^1 & 0 \\ 0 & 1-2a_q^1 \end{bmatrix},$$

$$C_j = \frac{1}{\Delta t^n}\begin{bmatrix} \frac{1}{8}\Delta x_{j-\frac{1}{2}} & -\frac{1}{2}\Delta t^n\alpha \\ -\frac{1}{8}\Delta t^n\beta g\left((\zeta+h)_{j-1}+3(\zeta+h)_j\right)^m & \frac{1}{8}\Delta x_{j-\frac{1}{2}}+\frac{1}{2}\Delta t^n\gamma\left(c_j^1\right)^m \end{bmatrix} \quad \text{for } j=1,\ldots,N_{el},$$

$$A_j = \frac{1}{\Delta t^n}\begin{bmatrix} \frac{6}{8}\Delta x_j & 0 \\ \frac{1}{8}\Delta t^n\beta g\left((\zeta+h)_{j-1}-(\zeta+h)_{j+1}\right)^m & \frac{6}{8}\Delta x_j+\frac{1}{2}\Delta t^n\gamma\left(c_j^2\right)^m \end{bmatrix} \quad \text{for } j=1,\ldots,N_{el},$$

$$B_j = \frac{1}{\Delta t^n}\begin{bmatrix} \frac{1}{8}\Delta x_{j+\frac{1}{2}} & \frac{1}{2}\Delta t^n\alpha \\ \frac{1}{8}\Delta t^n\beta g\left(3(\zeta+h)_j+(\zeta+h)_{j+1}\right)^m & \frac{1}{8}\Delta x_{j+\frac{1}{2}}+\frac{1}{2}\Delta t^n\gamma\left(c_j^3\right)^m \end{bmatrix} \quad \text{for } j=1,\ldots,N_{el},$$

$$C_{N_{el}+1} = \frac{1}{\Delta t^n}\begin{bmatrix} 1-2a_\zeta^{N_{el}} & 0 \\ 0 & 1-2a_q^{N_{el}} \end{bmatrix}, \quad A_{N_{el}+1} = \frac{1}{\Delta t^n}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

For the upwind approach the matrices for $j=1,\ldots,N_{el}$ are given by

$$C_j = \frac{1}{\Delta t^n}\begin{bmatrix} \frac{1}{8}\Delta x_{j-\frac{1}{2}}+\frac{1}{2}\Delta t^n\alpha\left(c_j^1\right)^m & 0 \\ -\frac{1}{8}\Delta t^n\beta g\left((\zeta+h)_{j-1}+3(\zeta+h)_j\right)^m & \frac{1}{8}\Delta x_{j-\frac{1}{2}}+\frac{1}{2}\Delta t^n\gamma\left(c_j^1\right)^m \end{bmatrix},$$

$$A_j = \frac{1}{\Delta t^n}\begin{bmatrix} \frac{6}{8}\Delta x_j+\frac{1}{2}\Delta t^n\alpha\left(c_j^2\right)^m & 0 \\ \frac{1}{8}\Delta t^n\beta g\left((\zeta+h)_{j-1}-(\zeta+h)_{j+1}\right)^m & \frac{6}{8}\Delta x_j+\frac{1}{2}\Delta t^n\gamma\left(c_j^2\right)^m \end{bmatrix},$$

$$B_j = \frac{1}{\Delta t^n}\begin{bmatrix} \frac{1}{8}\Delta x_{j+\frac{1}{2}}+\frac{1}{2}\Delta t^n\alpha\left(c_j^3\right)^m & 0 \\ \frac{1}{8}\Delta t^n\beta g\left(3(\zeta+h)_j+(\zeta+h)_{j+1}\right)^m & \frac{1}{8}\Delta x_{j+\frac{1}{2}}+\frac{1}{2}\Delta t^n\gamma\left(c_j^3\right)^m \end{bmatrix}.$$

This means that in the implementation the same algorithm can be used to create the matrices M_B and M_b and the only difference is the value for α, β and γ and the time of evaluation of certain terms. It also turns out to be very simple to make changes in the discretization, see for example the difference between the central or the upwind approach for the advection term in the mass equation. The only things that change in the implementation is the setup of matrix $M^m(\alpha,\beta,\gamma)$ and that of vector R^n.

A problem in solving (5.38) is that matrix M_B is dependent of values at the unknown time level $n+1$ and for the upwind approach the vector $\widetilde{R}^n$ also contains values at that time level. The easiest way to solve this is by using Picard linearization. This means that instead of the unknown values at time level $n+1$ known values at time level n are used to compute M_B and $\widetilde{R}^n$, and for this reason the method becomes semi-implicit. This can however in certain cases lead to errors in the numerical solution as can be seen in the numerical results in sections 7.3 and 7.4. Another way of dealing with the unknown values at time level $n+1$ is by using an iterative process of the following form:

1. Take $M_B = M^n(\theta_d, \theta_g, \theta_c)$ and $D^{n+1} = D^n$ in R^n (Picard linearization) and solve

$$M_B \widetilde{\mathbf{U}}^{(0)} = M_b \widetilde{\mathbf{U}}^n + \widetilde{R}^n. \tag{5.41}$$

2. For $i = 1, \ldots, N_{it}$, with N_{it} the number of iteration steps, take $M_B = M^{(i-1)}(\theta_d, \theta_g, \theta_c)$ and $D^{n+1} = D^{(i-1)}$ in R^n, where the superscript $(i-1)$ indicates that the values used in M_B and D are the ones obtained from $\widetilde{\mathbf{U}}^{(i-1)}$.
   Solve

$$M_B \widetilde{\mathbf{U}}^{(i)} = M_b \widetilde{\mathbf{U}}^n + \widetilde{R}^n. \tag{5.42}$$

3. Set $\mathbf{U}^{n+1} = \mathbf{U}^{(N_{it})}$.

Notice that when $N_{it} = 0$ the iterative process is identical to using Picard linearization. Another, more accurate, form of linearization is Newton-Raphson linearization, but that is not considered in this thesis.

The block-diagonal system is solved by an algorithm that first performs the block LU decomposition of the block-tridiagonal matrix and then solves the matrix system by means of a double sweep mechanism.

The order of accuracy of the time-integration can be derived by looking at the truncation error of the time-integration. The truncation error is obtained by substituting the exact solution into the numerical scheme and using Taylor expansion to compare it with the original differential equation. We will only look at the truncation error for the time-integration and simplify the discretization by neglecting the time-dependence of the matrices D, C and G and the vectors R_ζ and R_q. The truncation errors for the mass- and momentum equation then become

$$T_\zeta = M_\zeta \frac{\zeta_e(t+\Delta t) - \zeta_e(t)}{\Delta t} + D\left(\theta_d \mathbf{\Phi}_e(t+\Delta t) + (1-\theta_d)\mathbf{\Phi}_e(t)\right) - R_\zeta, \tag{5.43}$$

$$\begin{aligned} T_q = M_q \frac{\mathbf{q}_e(t+\Delta t) - \mathbf{q}_e(t)}{\Delta t} &+ C\left(\theta_c \mathbf{q}_e(t+\Delta t) + (1-\theta_c)\mathbf{q}_e(t)\right) \\ &+ G\left(\theta_g \zeta_e(t+\Delta t) + (1-\theta_g)\zeta_e(t)\right) - R_q, \end{aligned} \tag{5.44}$$

where $\boldsymbol{\zeta}_e$, $\mathbf{q}_e$ and $\mathbf{\Phi}_e$ are the vectors containing the exact solution in the vertices. We can express the variables $\phi = \{\boldsymbol{\zeta}_e, \mathbf{q}_e, \mathbf{\Phi}_e\}$ with a Taylor series around $t + \frac{1}{2}\Delta t$ as follows

$$\phi(t+\Delta t) = \phi + \tfrac{1}{2}\Delta t \partial_t \phi + \tfrac{1}{2}\left(\tfrac{1}{2}\Delta t\right)^2 \partial_t^2 \phi + \tfrac{1}{6}\left(\tfrac{1}{2}\Delta t\right)^3 \partial_t^3 \phi + \mathcal{O}((\Delta t)^4), \tag{5.45}$$

$$\phi(t) = \phi - \tfrac{1}{2}\Delta t \partial_t \phi + \tfrac{1}{2}\left(\tfrac{1}{2}\Delta t\right)^2 \partial_t^2 \phi - \tfrac{1}{6}\left(\tfrac{1}{2}\Delta t\right)^3 \partial_t^3 \phi + \mathcal{O}((\Delta t)^4), \tag{5.46}$$

where ϕ, $\partial_t \phi$ etc. are evaluated in $t + \frac{1}{2}\Delta t$. We find

$$\phi(t+\Delta t) - \phi(t) = \Delta t \partial_t \phi + \tfrac{1}{24}(\Delta t)^3 \partial_t^3 \phi + \mathcal{O}((\Delta t)^5), \tag{5.47}$$

$$\begin{aligned} \theta\phi(t+\Delta t) + (1-\theta)\phi(t) = \phi &+ \left(\theta - \tfrac{1}{2}\right)\Delta t \partial_t \phi + \tfrac{1}{8}(\Delta t)^2 \partial_t^2 \phi \\ &+ \tfrac{1}{24}\left(\theta - \tfrac{1}{2}\right)(\Delta t)^3 \partial_t^3 \phi + \mathcal{O}((\Delta t)^4). \end{aligned} \tag{5.48}$$

The truncation error can be found by substituting (5.47) and (5.48) for the appropriate variables and θ's in (5.43) and (5.44) and yields for the continuity equation

$$\begin{aligned} T_\zeta = \underbrace{\left[M_\zeta \partial_t \boldsymbol{\zeta}_c + D\mathbf{\Phi}_e - R_\zeta\right]}_{=0} &+ \left[D\left(\theta_d - \tfrac{1}{2}\right)\partial_t \mathbf{\Phi}_e\right]\Delta t \\ &+ \left[\tfrac{1}{24} M_\zeta \partial_t^3 \boldsymbol{\zeta}_e + \tfrac{1}{8} D \partial_t^2 \mathbf{\Phi}_e\right](\Delta t)^2 + \mathcal{O}((\Delta t)^4). \end{aligned} \tag{5.49}$$

For the momentum equation the truncation error becomes

$$T_q = \underbrace{\left[M_q \partial_t \mathbf{q}_e + C\mathbf{q}_e + G\zeta_e - R_q\right]}_{=0} + \left[C\left(\theta_c - \tfrac{1}{2}\right)\partial_t \mathbf{q}_e + G\left(\theta_g - \tfrac{1}{2}\right)\partial_t \zeta_e\right]\Delta t + \left[\tfrac{1}{24} M_q \partial_t^3 \mathbf{q}_e + \tfrac{1}{8} C \partial_t^2 \mathbf{q}_e + \tfrac{1}{8} G \partial_t^2 \zeta_e\right](\Delta t)^2 + \mathcal{O}((\Delta t)^4). \quad (5.50)$$

The first term of the truncation errors are equal to zero because the exact solution satisfies the differential equations (5.25a) and (5.25b). If all the θ's are equal to one half, a so-called Crank-Nicolson scheme, then the linearized scheme (due to the neglecting of the time dependence of D, C, G, R_ζ and R_q) is second order in time, because all the terms with Δt cancel in both truncation errors. If however one of the θ's is unequal to one half the linearized scheme is first order in time. The 1D SWE are however non-linear equations so we expect the scheme not to be perfectly second order in time when all the theta's are equal to one half when solving the 1D SWE.

When the advection terms are taken explicit, usually time step restrictions of the form

$$\Delta t \leq \frac{\Delta x}{|u|} \quad (5.51)$$

need to be satisfied. In our case the advection term is taken semi-implicit and there are no time-step restrictions for the linearized scheme. Only for accuracy reasons we want to limit the time-step.

5.2 Discontinuous Galerkin finite element method

The discontinuous Galerkin finite element methods (DG) are a class of finite element methods using a discontinuous piecewise polynomial space for the basis functions and the test functions. This has several advantages. First, like in finite element methods, it is straightforward to use unstructured grids. Second, the structure of the scheme allows to build in arbitrary order of accuracy per element. Third, the scheme is extremely local as the data communication occurs entirely through the faces between elements and this allows for efficient parallellizations. Also the implementation of boundary conditions is efficient and accurate due to the local nature of the scheme. Disadvantages are that the scheme is more complex, and that more degrees of freedom are involved relative to finite-volume schemes. The method used in this thesis is based on Cockburn & Shu (1989), Cockburn et al. (1989), Cockburn (1998), Schwanenberg & Harms (2003) and Schwanenberg (2003). We will only consider the 1D case of the shallow water equations. In Section 5.2.1 we start with the spatial discretization of the equations by discontinuous piecewise polynomials. This is followed in Section 5.2.2 by the time discretization which is done by a Runge-Kutta method. The principles of total variation are explained in 5.2.3 and a modification of the Runge-Kutta method is introduced by the implementation of a slope limiter. Finally we will in Section 5.2.4 introduce the HLLC numerical flux that will be used in the spatial discretization.

5.2.1 Spatial discretization

Division of the spatial domain in intervals is the same as in FEM, see Section 4.3. Again the unknowns are approximated by piecewise smooth functions, but these functions are not continuous any more. Figure 5.3 shows a discontinuous approximation of $\mathbf{u}$ by linear basis functions. In Bokhove (2003) basis functions, ϕ, corresponding to the mean and slope of the approxima-

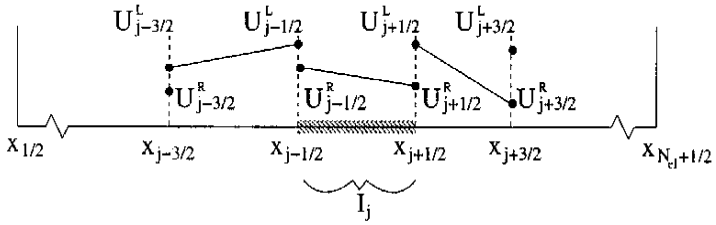


Fig. 5.3: Discontinuous Galerkin approximation U of $\mathbf{u}$ by linear basis functions.

tion are used, while in Cockburn (1998) and Schwanenberg (2003) Legendre polynomials P_l are chosen as local basis functions, which we also will use in this thesis. The L^2-orthogonality of the Legendre polynomials

$$\int_{-1}^{1} P_l(s)P_m(s)\,ds = \frac{2}{2l+1}\delta_{lm} \tag{5.52}$$

with δ_{lm} the Kronecker delta, can be exploited, as we will see, to obtain a diagonal mass-matrix. This freedom in the choice of basis functions is one of the advantages of DG compared to FEM.

The Legendre polynomials up to degree three are given by

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_2(x) &= \frac{1}{2}(3x^2 - 1), \\ P_3(x) &= \frac{1}{2}(5x^3 - 3x) \end{aligned}$$

The approximation U of $\mathbf{u}$ in each element $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ by the first $k+1$ Legendre polynomials is as follows

$$U(x,t) = \sum_{l=0}^{k} u_j^l(t)\phi_j^l(x) \qquad \text{for } x \in [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \tag{5.53}$$

with basis functions

$$\phi_j^l(x) = P_l\left(\frac{2(x-x_j)}{\Delta x_j}\right), \tag{5.54}$$

and $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ and $x_j = \frac{1}{2}(x_{j+\frac{1}{2}} + x_{j-\frac{1}{2}})$. So in each element $\mathbf{u}$ is approximated by a polynomial of degree k and there are $(k+1)$ unknown coefficients u_j^l, $l = 0, ..., k$ defined on each element I_j. This means that there are in total $N_{el} \times (k+1)$ unknowns in contrast with the N_{el} unknowns in the finite volume method. The approximation U belongs to the space

$$V_{DG} \equiv \left\{ v \in L^1(I) | v \in P^k(I_j) \, \forall I_j \in I \right\}, \tag{5.55}$$

where $L^1(I)$ is the space of Lebesque integrable functions on the domain I and $P^k(I_j)$ again the space of polynomials in I_j of degree k (Brenner & Scott (1994)).

We consider the weak formulation (4.12) obtained in deriving the finite element method, given by

$$\begin{aligned} \sum_{j=1}^{N_{el}} \Bigg\{ \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \omega \partial_t \mathbf{u} \, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{f}(\mathbf{u}) \partial_x \omega \, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{s}(\mathbf{u}) \omega \, dx \\ + \mathbf{f}(\mathbf{u}(x_{j+\frac{1}{2}}))\omega(x_{j+\frac{1}{2}}) - \mathbf{f}(\mathbf{u}(x_{j-\frac{1}{2}}))\omega(x_{j-\frac{1}{2}}) \Bigg\} = 0. \end{aligned} \tag{5.56}$$

We can not use (4.13) as we did in the finite element method, because due to the discontinuity of U the fluxes between the cells do not vanish any more. If the approximation U of the exact solution $\mathbf{u}$ is substituted in (4.12), the flux $\mathbf{f}(U)$ must be replaced by a numerical flux, F

$$F(U(x_{j+\frac{1}{2}}, t)) = F(U^L(x_{j+\frac{1}{2}}, t), U^R(x_{j+\frac{1}{2}}, t)), \tag{5.57}$$

where U^L and U^R denote the right and left limits of U respectively at the interfaces where U is discontinuous, see Figure 5.3. This numerical flux must have the same properties as the numerical flux encountered in the FVM in equation (4.7).

We take the test function ω out of the same space, V_{DG}, as U and again choose the test function the same as the basis functions, so $\omega = \phi_j^m(x)$ with $m = 0, ..., l$. Substituting $\omega = \phi_j^m(x)$, the approximation U and the numerical flux F in the weak formulation (4.12), will give in each element

$$\begin{aligned} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \left\{ \partial_l \sum_{l=0}^{k} u_j^l \phi_j^l \right\} \phi_j^m \, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{f}(U) \frac{d}{dx} \phi_j^m \, dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{s}(U) \phi_j^m \, dx \\ + F(U(x_{j+\frac{1}{2}}, t)) - (-1)^m F(U(x_{j-\frac{1}{2}}, t)) = 0, \end{aligned} \tag{5.58}$$

where we used $\phi_l(x_{j+1/2}) = P_l(1) = 1$ and $\phi_l(x_{j-1/2}) = P_l(-1) = (-1)^l$.

We can rewrite the first term of (5.58) as

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \partial_t \left\{ \sum_{l=0}^{k} u_j^l \phi_j^l \right\} \phi_j^m \, dx = \partial_t \left(\sum_{l=0}^{k} \left[u_j^l \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \phi_j^l \phi_j^m \, dx \right] \right)$$
$$= \frac{\Delta x_j}{2m+1} \frac{d}{dt} u_j^m, \tag{5.59}$$

where we used

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \phi_j^l(x) \phi_j^m(x) \, dx = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} P_l(s) P_m(s) \, dx$$
$$= \frac{\Delta x_j}{2} \int_{-1}^{1} P_l(s) P_m(s) \, ds = \frac{\Delta x_j}{2l+1} \delta_{lm}, \tag{5.60}$$

with $s = 2(x - x_j)/\Delta x_j$. By choosing this specific values for ω a diagonal mass-matrix can be obtained and as a result no matrix system has to be solved and the scheme becomes extremely local, allowing an efficient parallelization.

Substituting (5.59) into (5.58) we find for $j = 1, \ldots, N_{el}$ and $l = 0, .., k$:

$$\frac{d}{dt} u_j^l = \frac{2l+1}{\Delta x_j} \left[\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathrm{f}(U) \frac{d}{dx} \phi_j^l \, dx + \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathrm{s}(U) \phi_j^l \, dx \right]$$
$$- \frac{2l+1}{\Delta x_j} \left[F(U(x_{j+\frac{1}{2}}, t)) - (-1)^l F(U(x_{j-\frac{1}{2}}, t)) \right]. \tag{5.61}$$

Here we obtained a system of ODE's for the coefficients u_j^l of the form

$$\frac{d}{dt} U_h = L_h(U_h), \tag{5.62}$$

where U_h is a $N_{el} \times (k+1)$ matrix consisting of the coefficients u_j^l

$$U_h = \begin{bmatrix} u_1^0 & u_1^1 & \cdots & u_1^k \\ u_2^0 & u_2^1 & \cdots & u_2^k \\ \vdots & \vdots & & \vdots \\ u_{N_{el}}^0 & u_{N_{el}}^1 & \cdots & u_{N_{el}}^k \end{bmatrix}, \tag{5.63}$$

and L_h contains the right hand side of (5.61). The integrals in L_h can be easily approximated by a quadrature formula of the form

$$\int_a^b f(x) \, dx \cong \sum_{i=1}^{d} w_i f(\xi_i). \tag{5.64}$$

For the the three point Gauss approximation ($d = 3$), the weights w_i and the Gauss points ξ_i are given in Table 5.1. This three point approximation is exact for polynomials up to and including degree five.

Table 5.1: Values for the weights and the Gauss points for the three point Gauss approximation.

i	w_i	ξ_i
1	5/9	$a + \left(\frac{1}{2} - \frac{1}{2}\sqrt{3/5}\right)(b-a)$
2	8/9	$(b-a)/2$
3	5/9	$b - \left(\frac{1}{2} - \frac{1}{2}\sqrt{3/5}\right)(b-a)$

For constant basis functions ($k = 0$) equation (5.61) takes the form

$$\frac{d}{dt}u_j^0 = -\frac{1}{\Delta x_j}\left[F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}\right] + S_j, \tag{5.65}$$

where $F_{j+\frac{1}{2}} = F(u_j^0, u_{j+1}^0)$ is the flux between element j and $j+1$ and $S_j = \frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \boldsymbol{s}(U)\,dx$. This is exactly the same form as (4.6) for the finite volume method, so the first order RKDG-method is identical to the finite volume method described in Section 4.2. This means that the DG-method is a generic form of the finite volume method.

5.2.2 Runge-Kutta time-discretization

To discretize in time the TVD Runge-Kutta time discretization of Shu & Osher (1988) will be used as was done in Cockburn (1998). The starting point is a given initial condition $\mathbf{u}(x,0)$ satisfying the boundary conditions and its polynomial approximation U_{h0}. If $\{t^n\}_{n=0}^N$ is a partition of $[0,T]$, with T the total computation time, and $\Delta t^n = t^{n+1} - t^n$ for $n = 0, ..., N-1$, the algorithm is as follows:

1. Set $U_h^0 = U_{h0}$.
2. For $n = 0, ..., (N-1)$ compute U_h^{n+1} from U_h^n as follows:
   (a) set $U_h^{(0)} = U_h^n$,
   (b) for $i = 1, ..., (m+1)$, with $m+1$ the order of the Runge-Kutta method, compute the intermediate functions

$$U_h^{(i)} = \left\{\sum_{l=0}^{i-1} \alpha_{il} U_h^{(l)} + \beta_{il}\Delta t^n L_h(U_h^{(l)})\right\}, \tag{5.66}$$

   (c) set $U_h^{n+1} = U_h^{(m+1)}$.

The values for the coefficients α and β can be found in Tables 5.2 and 5.3. For example the first order Runge-Kutta scheme is given by

$$U_h^{n+1} = \alpha_{1,0}U_h^n + \beta_{1,0}\Delta t^n L_h(U_h^n) = U_h^n + \Delta t^n L_h(U_h^n), \tag{5.67}$$

which is actually the Explicit Euler Forward scheme.

For a stable numerical scheme, the order of the Runge-Kutta time discretization, $m+1$, must be at least as large as that of $k+1$, with k the highest degree of the used basis functions. So when using linear basis functions, as we do in this thesis, at least a second order Runge-Kutta scheme must be applied.

Explicit numerical methods, such as this Runge-Kutta method, are limited in the choice of the time step Δt because of stability reasons. A reference number related to the grid is the

Table 5.2: Values for α.

Order	i	$l=0$	$l=1$	$l=2$
$m=0$	1	1	-	-
$m=1$	1	1	-	-
	2	1/2	1/2	-
$m=2$	1	1	-	-
	2	3/4	1/4	-
	3	1/3	0	2/3

Table 5.3: Values for β.

Order	i	$l=0$	$l=1$	$l=2$
$m=0$	1	1	-	-
$m=1$	1	1	-	-
	2	0	1/2	-
$m=2$	1	1	-	-
	2	0	1/4	-
	3	0	0	2/3

CFL-condition. This number gives the relation between two speeds, namely the maximum wave propagation speed and the grid speed $\Delta x/\Delta t$. For one-dimensional elements this CFL-number can be defined as

$$\sigma = \frac{\Delta t}{\Delta x}(|u| + c) \quad \Rightarrow \quad \Delta t = \sigma \frac{\Delta x}{|u| + c}. \tag{5.68}$$

This condition actually says that the scheme will allow time steps Δt such that the fastest waves do not traverse more than a single cell of width Δx in time Δt. The value of σ lies between zero and one and depends on the actual order of the RKDG method. The maximal values for σ are given in Table 5.4 and also the values used in this thesis. The estimation of the maximal value

Table 5.4: Maximal and used values for σ for the RKDG-method.

Order	maximum value for σ	used value for σ
$k=0$	1.00	0.90
$k=1$	0.33	0.30
$k=2$	0.20	0.18
$k=3$	0.16	0.15

of σ is obtained from Schwanenberg (2003), where the estimation is carried out numerically because analytical enquiries failed. The used values of σ are also the same as in Schwanenberg (2003) and they correspond in 90% of the numerical experiments to the maximal possible time step.

5.2.3 Total variation properties

Total variation The basic idea of the Total Variational Diminishing (TVD) methods is to restrict the total variation to avoid unphysical oscillations in the numerical solution. Given a function $u = u(x)$, the total variation of u is defined as

$$TV(u) = \lim_{\delta \to 0} \sup \frac{1}{\delta} \int_{-\infty}^{\infty} |u(x+\delta) - u(x)|\, dx. \tag{5.69}$$

Moreover, if $u^n = \{u_j^n\}$, then the total variation of u^n is defined as

$$TV(u^n) = \sum_{j=-\infty}^{\infty} |u_{j+1}^n - u_j^n|. \tag{5.70}$$

A fundamental property of the exact solution of the non-linear scalar conservation law

$$\partial_t u + \partial_x f(u) = 0, \tag{5.71}$$

when the initial data $u(x,0)$ has bounded total variation, is

- No new local extrema in x may be created.
- The value of a local minimum does not decrease and the value of a local maximum does not increase.

From this it follows that the Total Variation $TV(u(t))$ is a decreasing function of time

$$TV(u(t_2)) \leq TV(u(t_1)) \qquad \forall t_1 \leq t_2. \tag{5.72}$$

This property of the exact solution is the one that we want to mimic when designing numerical methods. A scheme is now said to be a Total Variation Diminishing (TVD) scheme if

$$TV(u^{n+1}) \leq TV(u^n), \qquad \forall n. \tag{5.73}$$

This statement can be extended to a system of conservation laws, such as the 1D shallow water equations, by applying it separately to all the characteristic variables.

Numerical methods with property (5.73) are called TVD methods. First order upwind schemes satisfy this property in principle, but they lead to dispersed shock-waves and a poor quality of the approximation. It can be proven, see Cockburn (1998), that the first order RKDG method ($k = 0$) satisfies the TVD property. Higher order methods give substantial better solutions, but close to steep gradients and discontinuities unphysical oscillations may occur, hence the TVD property is violated. TVD methods are therefore equipped with a suitable mechanism to maintain the TVD property. This mechanism makes sure that the higher order method is used when the solution is smooth, but in the vicinity of steep gradients or discontinuities it uses a first order method to keep the TVD property satisfied.

Slope limiter One of the techniques to construct a TVD method is to use a slope limiter. Slope limiter methods are based on the reconstruction of the variables inside an element using linear or quadratic functions. We will use the slope limiter $\Lambda\Pi_h^k$ as given in Cockburn (1998) to modify the RKDG method in such a way that it will also satisfy the TVD property when the order is greater than one.

Recall that in (5.53) the approximation of $\mathbf{u}$ was done by Legendre polynomials and was called U

$$U(x,t) = \sum_{l=0}^{k} u_j^l(t)\phi_j^l(x). \tag{5.74}$$

Define what could be called the linear part of U

$$V(x,t) = \sum_{l=0}^{1} u_j^l(t)\phi_j^l(x), \tag{5.75}$$

with the matrix containing all the coefficients given by

$$V_h = \begin{bmatrix} u_1^0 & u_1^1 \\ u_2^0 & u_2^1 \\ \vdots & \vdots \\ u_{N_{el}}^0 & u_{N_{el}}^1 \end{bmatrix}. \tag{5.76}$$

We will apply the slope limiter $\Lambda\Pi_h^k$ on V_h. The slope limiting is done with respect to the characteristic values, because we are dealing with a system of partial differential equations. We define $V_{\text{lim}} = \Lambda\Pi_h^k(V_h)$, in the following way:

For $j = 1, ..., N_{cl}$ compute $u^0_{\text{lim},j}$ and $u^1_{\text{lim},j}$ as follows:

1. Compute the following transformed variables

$$v^1_j = B^{-1} u^1_j, \tag{5.77}$$
$$v^L_j = B^{-1} \left(u^0_j - u^0_{j-1} \right), \tag{5.78}$$
$$v^R_j = B^{-1} \left(u^0_{j+1} - u^0_j \right), \tag{5.79}$$

   where the Jacobian B is as in (2.45) given by

$$B = \begin{bmatrix} 0 & 1 \\ \sqrt{gH^0_j} - \left(q^0_j / H^0_j \right)^2 & 2q^0_j / H^0_j \end{bmatrix}. \tag{5.80}$$

2. Perform the limiting of v^1_j as follows

$$v^1_{\text{lim},j} = m \left(v^1_j, v^L_j, v^R_j \right), \tag{5.81}$$

   where the minmod function m is defined as

$$m(a_1, a_2, a_3) = \begin{cases} sign(a_1) \min\limits_{1 \le n \le 3} |a_n| & \text{if} \quad sign(a_1) = sign(a_2) = sign(a_3), \\ 0 & \text{otherwise.} \end{cases} \tag{5.82}$$

3. Transform the variable $v^1_{\text{lim},j}$ back, to obtain the limited value of u^1_j. No modification of u^0_j is needed, because only the slope is limited.

$$u^0_{\text{lim},j} = u^0_j, \tag{5.83}$$
$$u^1_{\text{lim},j} = B v^1_{\text{lim},j}. \tag{5.84}$$

Then V_{lim} is given by (5.76), only with the coefficients u^n_j replaced by $u^n_{\text{lim},j}$. See Figure 5.4 for a graphical representation of the limiting process. It can be seen that the slope in element j is limited between the mean values in the neighbouring elements.

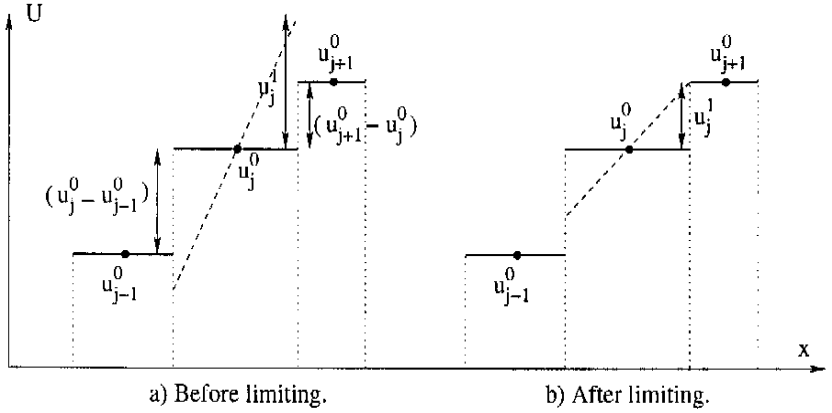


Fig. 5.4: Functioning of the slope limiter.

The slope limiter is implemented by modifying the Runge-Kutta time-discretization by replacing equation (5.66) with

$$U_h^{(i)} = \Lambda\Pi_h^k \left\{ \sum_{l=0}^{i-1} \alpha_{il} U_h^{(l)} + \beta_{il} \Delta t^n L_h(U_h^{(l)}) \right\}, \tag{5.85}$$

This means that the limiting is performed after each step of the RKDG method. In Cockburn (1998) it is proven that the thus obtained RKDG method satisfies the TVD property for the mean of U for the scalar case. It is expected that the favourable properties also apply to the system case.

It is possible to modify the slope limiter in such a way that the degradation of the accuracy at local extrema is avoided. In Figure 5.5 an example of the approximation at a local extremum is given. The slope limiter, as it introduced till now, limits the slope in cells j and $j+1$ to zero.

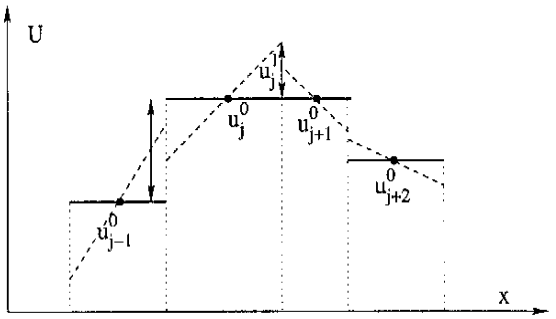


Fig. 5.5: Limiting at local extrema.

This reduces the accuracy and this can be avoided by not applying the slope limiter as long as the slope is less than a certain threshold. The method becomes then total variation bounded (TVB) instead of total variation diminishing (TVD).

To achieve this, we modify the definition of the slope limiter by simply replacing the minmod function m by the TVB corrected minmod function $\widetilde{m}$ as follows

$$\widetilde{m} = \begin{cases} a_1 & \text{if} \quad |a_1| \leq M(\Delta x)^2, \\ m(a_1, a_2, a_3) & \text{otherwise.} \end{cases} \tag{5.86}$$

where M is a given constant. This means that as long as $|v_j^1| \leq M(\Delta x)^2$ no limiting is performed. The TVB correction constant M is an upper bound of the absolute value of the second-order derivative of the solution at local extrema. When $M = 0$, the TVB corrected minmod function reduces to the original minmod function. In this thesis the TVB corrected minmod function is used.

According to Schwanenberg (2003), the above described slope limiter can together with great changes in the bottom level lead to uncorrect results. After limiting, the water level can become constant leading to great changes in the water level whenever the bottom level is not constant. This can be overcome by limiting the water level ζ instead of the total water depth H. For constant bottom level this reduces to the above described slope limiter. In this work only the above described slope limiter is used and not the one which takes the bottom level into account.

5.2.4 The HLLC Solver

There are many possible choices for calculating the numerical flux in equation (5.61). In this thesis we will use the so-called HLLC solver. The HLLC solver is an approximate Riemann solver

and a modification of the basic HLL scheme. Full details on the HLL and HLLC approach are given in Toro (1997).

Denote by $U_L = [H_L, q_L]$ and $U_R = [H_R, q_R]$ the state on the left and the right side, respectively, of an interface where U is discontinuous. Then the velocity on both sides is given by $u_K = q_K/H_K$ and the celerity by $c_K = \sqrt{gH_K}$ for $K = L, R$. Figure 5.6 illustrates the assumed wave structure in the HLLC Riemann solver. The HLLC approach assumes estimates

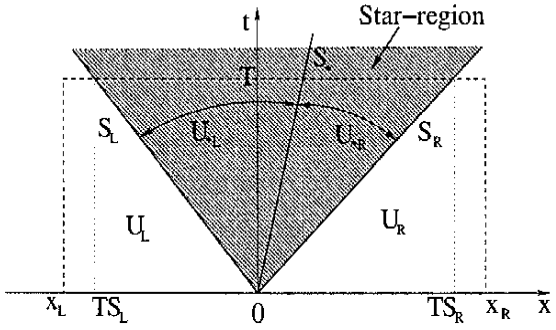


Fig. 5.6: HLLC Riemann solver.

S_L, S_R and S_* for the smallest and largest signal velocities and the speed of the middle wave, corresponding to the different eigenvalues of the system. There are several possible choices available. According to Toro (2001) we will use

$$S_L = u_L - c_L r_L, \qquad S_R = u_R + c_R r_R, \tag{5.87}$$

where r_K for $K = L, R$ is given by

$$r_K = \begin{cases} \sqrt{\frac{1}{2}\left[\frac{(H_*+H_K)H_*}{H_K^2}\right]} & \text{if} \quad H_* > H_K, \\ 1 & \text{if} \quad H_* \le H_K. \end{cases} \tag{5.88}$$

Here H_* is an estimate for the exact solution for H in the star region between S_L and S_R. We will use the estimate

$$H_* = \frac{1}{2}(H_L + H_R) - \frac{1}{4}(u_R - u_L)(H_L + H_R)/(c_L + c_R). \tag{5.89}$$

The wave speed of the middle wave will be estimated as follows

$$S_* = \frac{S_L H_R(u_R - S_R) - S_R H_L(u_L - S_L)}{H_R(u_R - S_R) - H_L(u_L - S_L)}. \tag{5.90}$$

According to Toro (1997) the HLLC numerical flux can be derived by integrating over the space-time volume $[x_L, x_R] \times [0, T]$ as depicted in Figure 5.6 to yield

$$F_{j+\frac{1}{2}} = \begin{cases} F_L & \text{if} \quad 0 \le S_L, \\ F_{*L} & \text{if} \quad S_L \le 0 \le S_*, \\ F_{*R} & \text{if} \quad S_* \le 0 \le S_R, \\ F_R & \text{if} \quad 0 \ge S_R, \end{cases} \tag{5.91}$$

where

$$F_K = \mathbf{f}(U_K), \tag{5.92}$$

$$F_{*K} = F_K + S_K(U_{*K} - U_K), \tag{5.93}$$

and the states U_{*L} and U_{*R} are given by

$$U_{*K} = H_K \left(\frac{S_K - u_K}{S_K - S_*} \right) \begin{bmatrix} 1 \\ S_* \end{bmatrix} \tag{5.94}$$

For the one dimensional case the HLLC numerical flux reduces to the HLL flux, because there are only two eigenvalues and hence there is no middle wave and then $U_{*L} = U_{*R}$.

Chapter 6

Flooding and drying

The shallow water equations are typically used to model surface irrigation, overland flow, river and lake hydrodynamics, and long wave runup, as well as estuarine and coastal circulation. Many of these applications involve moving boundaries in which flooding and drying occurs, and simulating these processes is becoming increasingly important. Predictions of flooding due to a storm surge, breached dam, or overtopped dike are crucial for disaster planning and wave runup estimates are needed for beach and coastal structure design.

The definition of moving boundaries would appear to be rather straightforward: the water depth $H = 0$. However, there are some difficulties involved. First, with this condition, the hyperbolic character of the equations gets lost, because all the propagation speeds coincide, and it is not so obvious that you are left with a mathematically well-posed problem. Second, some terms in the flow equations may have a singular character, for example friction terms.

Many numerical models have been developed for these problems and have used different approaches to accommodate the flooding and drying process. Early models neglected flooding and drying and instead placed fixed wall boundaries near the shoreline. Other models were initialized with a thin layer of water everywhere in the domain. However, with stationary boundaries, the storage, conveyance and energy dissipation properties of intermittently wetted areas are completely ignored, while assuming a thin layer of water everywhere results in incorrect propagation of waves. This means that the flooding and drying fronts should be tracked in some way.

According to Shyy et al. (1996) the techniques for tracking moving boundaries can be classified in two main categories:

(a) surface tracking or Lagrangian methods and

(b) volume tracking or Eulerian methods.

The most physically realistic solution would be to use a numerical grid that adapts at each timestep to follow the continuously deforming fluid boundary, a so called Lagrangian method. These methods require recalculation of nodal positions and automatic reconfiguration of the grid near the shoreline. The use of such schemes other than as research tools is not straightforward. This is the reason why most methods use an Eulerian approach.

The Eulerian methods usually employ a fixed grid formulation and the interface is not explicitly tracked, but is reconstructed from the variables. This approach supposes that the modeler is able to make an 'educated guess' about the region that will be affected by the flow field.

This chapter contains a literature survey of flooding and drying mechanisms as they are used nowadays. All the methods mentioned here use an Eulerian approach and they are subdivided into the following techniques:

- masking out or including whole elements,
- introduction of artificial porosity,
- modification of the shallow water equations,
- tracking the boundary points,
- extrapolation of variables,
- modification of the Riemann solver,

In the next paragraphs, the different approaches are explained and outlined by an abstract of relevant literature. For details of the procedures the literature in question can be consulted. At the end of this chapter some recommendations are given for implementing flooding and drying algorithms in the CCSM-method and the RKDG-method.

Masking out or including whole elements
In this approach the drying and flooding is constrained to follow the sides of the grid cells. The process of drying and flooding is represented by removing grid points (in finite difference approach) or cells (in finite volume or element approach) from the flow domain that become 'dry' and by adding grid points or cells that become 'wet' and no special tracking procedure for the shoreline is used

In Delft3D-FLOW, see User Manual Delft3D-Flow (2001) and de Goede (1995), a staggered finite difference method is used. If the total water level drops below a specified threshold, the velocity point is set to zero. If the total water level at a water level point becomes negative, the velocity points at the cell sides are set dry. The boundary of the wet area can only move one grid cell per time step, otherwise oscillations after flooding are generated. Ten models that are all based on a staggered grid, and are similar to the one used in Delft3D-FLOW, are reviewed and evaluated in Balzano (1998). With all the methods considered it is possible to advance one grid at a time.

In Hu et al. (2000) a finite volume model using an approximate Riemann solver is used. A computational cell is assumed to be dry when its water depth is below a 'minimum wet depth'. The bottom friction may then become very large and a 'minimum friction depth' is introduced. When the water is shallower than this depth, the 'minimum friction depth' is used to calculate an equivalent friction loss.

A similar approach is used in Hubbard & Dodd (2002) and Sleigh et al. (1998). A fully adapted mesh approach is undertaken in Hubbard & Dodd (2002) in which high grid resolution is used only where necessary. The shoreline is covered with the finest possible mesh and the start of the flooding and drying procedure consists of searching for dry cells which are in imminent danger of flooding. Each cell with this property is then wetted by setting the water dept $H = H_{tol}$ and keeping the velocity in this cell at zero. After the update (performing the rest of the algorithm), the cells in which the water depth has dropped below zero are considered dry and their depth is reset to zero. A second depth balance $H_{TOL} > H_{tol}$ is introduced for cells that are almost dry. When the water level drops below H_{TOL} the depth is not altered, but the x- and y-momentum are set to zero.

In Sleigh et al. (1998) the problem is reformulated when the water depths are small but grid cells are only removed from the calculation when the water depth is very small. Depending on the combination of wet/dry/partial-dry cells and cell-faces, a choice of flux calculation for each cell face is made. A cell-face is a land boundary if the left and the right water depth are smaller than H_{tol}. A cell is dry if the water depth is less than H_{tol} and all cell-faces are land boundaries.

A cell is partially dry if the water depth of the fluid is greater than H_{tol} but less than H_{TOL}, or when the water depth is less than H_{tol} but one of the cell-faces is not a land boundary. For a partially wet cell the momentum fluxes are set to zero. The cell is wet if the water depth in the cell is greater than H_{TOL}.

Introduction of artificial porosity

When using artificial porosity the water level is allowed to drop below the bottom level, due to a permeable bottom. In this way the flow equations can also be applied in the 'dry' region. Some extra conditions have to be imposed in the 'dry' region to avoid mass balance errors.

In TRITON the concept of the artificial porosity is introduced by considering a permeable bottom in the near shore region, see Bonekamp & Borsboom (2002). A porosity layer depth δ_{dry} is introduced, which is small compared to the water depth. The total water depth can be considered as an integral over the porosity variable α,

$$H = \int_{-\infty}^{\zeta} \alpha \, dz.$$

The porosity α is modelled as

$$\alpha = \begin{cases} 1, & \zeta \geq -h + \delta_{dry}, \\ \exp\left(\frac{\zeta + h - \delta_{dry}}{\delta_{dry}}\right), & \zeta \geq -h + \delta_{dry}. \end{cases}$$

The porosity model adjust the reference bottom depth h and replaces it by an effective reference depth h_{eff} which contains the effect of the artificial porosity. A proper functioning of the drying and flooding procedure requires the introduction of bottom friction, because in this way large discharges at very low water depths are avoided. This is done by adding an artificial bottom friction term to the momentum equation that slows down the flow, but only at low water depths. The model is implicit and this means that greater time steps can be chosen. A disadvantage is that if the porosity layer depth is too small a 'film' of water is created if the water retreats.

In Ertürk et al. (2002) also a porous layer beneath the bottom level is introduced. The flow in the porous layer is described by Darcian flow, see for example Faber (1995). The kinematic momentum balance is substituted into the continuity equation and also the Darcian description of the porous layer is incorporated. This results in a scalar nonlinear diffusion equation, with a storage coefficient and a nonlinear diffusion coefficient as functions of the total water depth. Both coefficients have a different expression for the porous layer and the water layer.

Nothing is presumed or tested on the water level position in Heniche (2000). By letting the water level free to plunge under the bed level, positive and negative water depth values may be encountered. The model is able to reproduce the same problem with either negative or positive water depths, but this also means that the model cannot distinguish between the wet and the dry area. The porosity of the field is taken into account to make the difference between wet and dry area. In the dry area the porosity is equal to zero and in the wet area it is equal to one. The mass and momentum equation are modified, by using in the dry area only the steady state conditions. The flow in the dry area is also frozen in order to have zero discharge conditions. This is done by increasing the friction coefficient. Problems with the singularity at the shoreline, where $H = 0$ are overcome by correcting the water depth as $\widetilde{H} = \max(H_{min}, |H|)$.

Modification of the shallow water equations

Standard models are usually based on numerical integration of the shallow water equations. The main drawback of these equations lies in that they strictly apply solely to the wet domain. The approaches presented in this paragraph attempt to reformulate the flow equations over partially

wet elements by introducing a scaling coefficient, representing the true volume of water residing on each element.

In Bates & Anderson (1993) a domain coefficient, θ, is defined that represents the proportion of the fluid domain available for flow. This varies between 1 (fully wet) and 0 (fully dry) as the elements de-waters. To approximate the flow boundary, partly wet elements are maintained within the computation and the domain coefficient is used to scale the simulated elemental water volume to the true water volume residing on the partial wet element. A small positive water depth is maintained at each node while the element of which it is part remains within the computation. This is achieved by defining some minimum value of θ, θ_{min}, which is small and positive and represents nearly dry conditions. This avoids the reduction of stability through the incorporating nodes with zero depth. A discontinuity in the value of θ, when the total water depth becomes zero, is overcome by allowing θ to vary between $\theta = 1$ and $\theta = \theta_{min}$ over some small range.

In Bates & Hervourt (1999) the same procedure as in Bates & Anderson (1993) is used, but this time there is also made the distinction between dam break and flooding types within the partly dry elements. It consists of the following three components: (1) Identification of partly dry elements on the basis of a simple two stage analysis of water surface slopes. A distinction is made between dam break and flooding types. (2) Cancellation of spurious water surface slope terms in the momentum equation for the partial dry elements. (3) Rescaling of the continuity equation to represent the true volume of water on partial dry elements on the basis of the sub-grid topography.

In Defina (2000) the equations are obtained by an average process consisting of multiplying with a phase function, which is one if there is water and zero if there is no water in an element, integrating over an element and dividing by the area of the element. For the continuity equation this results in multiplying the time derivative of H with the wet fraction of the element. If the water depth is large enough, the equations reduce to the normal shallow water equations. A very thin layer of water is initially ponded in the dry domain to avoid a mathematical singularity at the beginning of the computation. There are two choices for dealing with partial dry elements, to either include or exclude them from the computational domain. In both cases, errors in the mass balance are introduced and lead to unrealistic flow fields near the wet/dry boundaries. The approach proposed in Defina (2000) allows gradual flooding and drying of each computational element. Because the equations apply both to wet and partially dry elements, correct identification of partially wet elements is not required.

In Horritt (2002) three methods for the treatment of partially wet elements in finite element models are evaluated. The technique of Defina (2000) is followed, but the continuity equation is not modified. In the element masking (EM) approach the partially wet elements are masked out in the calculation. The mass balance error for EM is getting worse for rapid inundation problems and coarse grids. The mass balance errors can be reduced by adding a correction to the continuity equations, for the elements next to the partial wet elements, resulting in the continuity correction (CC) scheme. First, masked elements next to unmasked ones are identified and then the water free surface elevation is projected across the masked element. This is implemented as a source term. Rather than masking out elements, the model can also include all elements in the calculation in order to conserve both mass and momentum in partially wet elements. Setting the free surface to zero in the momentum equation (free surface correction (FSC) scheme) avoids accelerations away from the shore line in partial wet elements. A small positive water depth is used to calculate the friction terms over dry nodes to avoid instabilities due to the friction term when the water depth tends to zero.

Tracking the boundary points

The method used in Bokhove (2003) is till now only implemented for the 1D problem in a finite element method and uses a more Lagrangian approach. The fluid is divided into one or more distinct patches of fluid. In one dimension, each patch has a left and a right boundary. A dry-wet boundary is modelled with a finite element where the node at the boundary moves with the flow. In this moving node the water depth is equal to zero. The finite element formulation is adjusted to deal with the moving nodes. This is done by taking into account elements of varying length and the flux is modified for the speed of the moving nodes. For elements which are not next to a free boundary, the formulation reduces to the normal finite element discretization. In order to maintain the mostly Eulerian nature of the numerical scheme (1) an edge element is split when it becomes too large, (2) an edge element is merged with its neighbour if it becomes too small. Therefore the number of elements may change over time. In addition fluid patches can also merge and split.

Extrapolation of variables

Another way of dealing with flooding and drying fronts is to extrapolate the values for the flow variables in the partial wet or dry cells from the values in the neighbouring wet cells.

In the finite volume method of Bradford & Sanders (2002) new velocities are computed in a fully wet cell only if $H > H_{tol}$. The surface slope in a wet cell bounded by a partial wet of dry cell is linearly extrapolated from the wet neighbour. In addition the value of the surface level at the partial wet side of a cell face is extrapolated from the fully wet side. The momentum equations are not solved in partially filled cells and instead the velocities are extrapolated from the neighbouring wet cell with the largest water depth. Neumann extrapolation (the velocity is taken the same as in the wettest neighbour) is found to work better than Characteristic extrapolation (the characteristic in the wet cell is taken the same as that in the wettest neighbour).

Lynch & Gray (1978) uses a fixed grid finite difference model. In this paper a linear extrapolation is used near the wet/dry boundary, thereby allowing the real boundary location to exist in between nodal points. It would be advantageous if the moving boundary scheme did not require any sort of special treatment of the derivatives near the wet/dry boundary. With this in mind, the moving boundary scheme will employ a linear extrapolation of free surface level and velocity components, through the wet/dry boundary, and into the dry region. With extrapolated values of ζ and u in the dry region, solving the model equations at wet nodes can proceed. Although no derivatives are calculated at dry points, the physical values of free surface and velocity at these points are used to evaluate derivatives at neighbouring wet points. If $H > \delta$, the node is assumed to be wet and the model equations will be applied at the node; otherwise, the physical variables at the node will be extrapolated from a neighbouring node. In 1D this is done by using the two wet points nearest to the boundary to perform a linear extrapolation into the dry region. The 2D extrapolation is performed by checking the surrounding wet points, and for each a 1D linear interpolation is carried out. The free surface value at the dry node is taken to be the average of the 1D extrapolations. This approach avoids the appearance of $2\Delta x$ waves which occurs if the velocity is simply set to zero in the dry nodes.

In Quecedo & Pastor (2002) a Taylor-Galerkin finite element method is used. The authors followed the simple method of interpolation within the elements using the nodal variables, considering a null value for the variables corresponding to dry nodes. In this way, calculations to accurately determine the position of the boundary within the partially dried-flooded elements are not done.

Modification of the Riemann solver

A finite volume method based on an approximate Riemann solver is applied in Brufau et al. (2002). The moving boundaries are considered as wetting fronts, and hence are included in the ordinary cell procedure in a through calculation that assumes zero water depth for the dry cells. As far as we can judge from the paper, the moving boundaries are restricted to follow the cell faces. This approach provides satisfactory results when dealing with wetting fronts over flat or downward sloping surfaces but can lead to difficulties in advances over adverse slopes. Wetting fronts over dry surfaces can be reduced to Riemann problems in which one of the initial depths is zero. The solution for a sloping bed, when dealing with adverse slopes, identifies a subset of conditions incompatible with fluid motion (stopping flow). Two situations can be found for a wetting and drying front over an adverse slope: (1) $H_L < H_R$, this corresponds to the stopping conditions and the basic procedure has to be modified, (2) $H_L \geq H_R$, nothing has to be done and the basic Riemann solver can be used. When (1) occurs a modification is made in the bed slope. This is done to satisfy the equilibrium condition

$$(\Delta H)_{LR} = 0 \Rightarrow (\Delta \zeta)_{LR} = -(\Delta h)_{LR},$$

where $\Delta x_{LR} = x_L - x_R$. This condition is derived by the discretization of the mass equation to ensure still water steady state at the interface LR. When this condition is not satisfied, numerical velocities with no physical meaning can occur. The technique proposed is to enforce local redefinition of the bottom level difference at the interface to fulfil the equilibrium and therefore mass conservation in the way that $(\Delta \zeta)_{LR}^{mod} = -(h_R - h_L)$. It is also necessary to reduce the velocity components at the interface LR to zero to avoid too rapid propagation of the front.

According to Tchamen & Kahawita (1998), a positive depth scheme (PDS) is a scheme that always ensures a positive depth when applied to the full nonlinear SWE and they are a subset of the TVD-schemes. Although TVD-schemes are an extension of monotone schemes to a coupled system, they do not automatically produce a depth positive scheme. A sufficient condition is that the scheme for the depth relationship may be written in the following form:

$$H_j = b + \sum_k a_k H_k \tag{6.1}$$

with the coefficients b and a_k being non-negative, but no assumption of linearity is made. When the Van Leer's flux is used in the finite volume technique a scheme that belongs to the family of PDS is obtained. Positive depth is a desirable property, but not sufficient to guarantee stable bounded solutions. The authors of Tchamen & Kahawita (1998) found a necessary condition for the existence of a solution

$$u_R - 2c_R \leq u_L + 2c_L. \tag{6.2}$$

Most Riemann solvers display instabilities when this condition is violated. Too strong bottom curvature may be one of the main reasons for violation. To obtain a stable solution near partly wet cells, the imposition of zero velocity is still used. To date, this appears to be the only approach that may be used to stabilize the solution in a specific situation, such as a stable, partially wet cell, without any recourse to some kind of subcell resolution. However, an imposed zero front velocity will result in an artificial slowing of the propagation velocity. It is not clear from the paper how the authors deal with the drying process, because a positive depth scheme avoids the water depth becoming zero or negative.

Recommendations

When including flooding and drying in the CCSM-method and the RKDG-method it is advisable to start with a very simple algorithm. Masking out or including whole elements, such as done in Hu et al. (2000) and Hubbard & Dodd (2002) will be a good start. This algorithm can be easily implemented in both the CCSM-method and the RKDG-method and the way both methods cope with the algorithm can be investigated and compared.

For a more realistic representation a flooding and drying procedure which modifies the shallow water equations can be used. In this way a better handling of partial wet cells is achieved and it is also possible to use implicit time integration. The approach used in Defina (2000) is a good starting point and later issues discussed in Bates & Hervourt (1999) can be included. In both the CCSM-method and the RKDG-method it is possible to do the necessary modifications of the shallow water equations.

It would also be interesting to investigate the more Lagrangian approach of Bokhove (2003). In Bokhove (2003) only the RKDG-method is considered, but the procedure must also be possible for the CCSM-method, because a linear approximation of the variables is used.

Chapter 7

Test cases

In this chapter some numerical test cases are presented for which the exact solution is known. These problems will be used to test the numerical methods by comparing the numerical solution with the exact solution. Also one more realistic test case without exact solution is done in which tidal wave movement is mimicked. We will start in Section 7.1 with some preliminary information in order to be able to compare both numerical methods and to define a way in which the numerical solutions will be compared with the exact solution. In Section 7.2 we will look at the solution of the linearized equations for a small perturbation of an initial constant condition. This test case is used to check the convergence of both methods and examine the time step restrictions. Transition in time from continuous solutions to solutions containing discontinuities is examined in Section 7.3, by looking at the solution when one of the Riemann variables is taken constant. Two different Riemann problems are solved in Section 7.4. The first one is a dam break problem and the second one involves two rarefractions and a nearly dry bed. In Section 7.5 a tidal wave is mimicked by enforcing a sine-shaped boundary condition at the left boundary of the domain. The last two test cases take into account the source term caused by a non-flat bottom. In Section 7.6 the ability of the methods to deal with changes in the bottom level is tested. Flow over an isolated ridge, as is examined analytically in Houghton & Kasahara (1968), is the subject of Section 7.7.

7.1 Preliminary information

In order to compare both methods described in Chapter 5, the number N and the location of the grid points are taken the same in both methods. However, the control volumes in the CCSM-method do not coincide with the elements in the RKDG-method. This means that the value of N_{el} is equal to $N-1$ in the RKDG-method and equal to N in the CCSM-method. In this way the domain of computation of the CCSM-method will be larger than that of the RKDG-method. This is because the boundaries in the RKDG-method coincide with the first and the last grid point, but the boundaries in the CCSM-method coincide with the boundaries of the control volumes around the first and the last grid point, see Figure 5.1. Only for the test case with the tidal wave this has to be dealt with, because in the other test cases we are only interested in the part of the solution far from the boundaries.

In each test case we start with a given analytical initial condition $U_0(x)$, with $U = \{H, q\}$, that has to be approximated on the discrete grid. The graphical representation of the difference in the approximation used in both methods for the first order case is given in Figure 7.1. The initial vector for the first order RKDG-method, $\mathbf{U}_0^{\text{RKDG}}$, contains the mean values of $U_0(x)$ in each element and has length $(N-1)$. For the CCSM-method the initial vector, $\mathbf{U}_0^{\text{CCSM}}$, of length

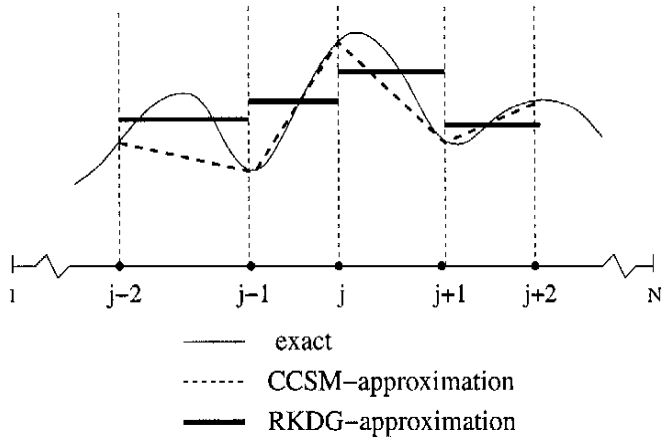


Fig. 7.1: Approximation of a function in the first order RKDG- and CCSM-method.

N contains the values of $U_0(x)$ in the grid points.

For each of the test cases described in the next sections we know the exact solution, $U_{ex}(x)$, (or a good approximation of it) at a given time. The numerical solutions at the end-time, T_{end}, of the computation will be compared with the exact solution at that time, both graphically and numerically. The numerical comparison will be done by looking at the error in the max-norm, L_∞, and in the L_1-norm which are defined as follows

$$E_\infty = \max(|U - U_{ex}|), \quad E_1 = \int_{x_1}^{x_N} |U - U_{ex}|\, dx, \tag{7.1}$$

where U is the numerical solution. The exact solution U_{ex} is evaluated in the grid points x_j and also in $M-1$ points, with M an even integer, equally spaced between the grid points x_j and x_{j+1}. The number of points, N_{ex}, in which U_{ex} is evaluated is then given by the relation $(N_{ex}-1) = M(N-1)$. In this way M is the factor between the number of grid cells of the exact solution and the numerical solution. The value of M has to be chosen big enough to get a reasonable value for the error and is normally chosen larger than 20.

The max-norm of the error will be approximated in the following way

$$E_\infty = \max_{1 \le i \le N_{ex}} (|U_i - U_{ex,i}|). \tag{7.2}$$

The L_1-norm of the error can be written as follows

$$E_1 = \int_{x_1}^{x_N} |U - U_{ex}|\, dx = \sum_{j=1}^{N-1} \int_{x_j}^{x_{j+1}} |U - U_{ex}|\, dx = \sum_{j=1}^{N-1} \sum_{i=0}^{M/2} \int_{x_{j+\frac{2i}{M+1}}}^{x_{j+\frac{2i+1}{M+1}}} |U - U_{ex}|\, dx \tag{7.3}$$

where the integrals in the last term can be approximated by Simpson's formula

$$\int_a^b f(x)\, dx \approx \tfrac{1}{6}(b-a)\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right). \tag{7.4}$$

This is also the reason why we used $M/2$ in the summation in (7.3) because in this way the point $(a+b)/2$ in Simpson's formula coincides with a point in which U_{ex} is evaluated, since this points were equally spaced.

7.2 Linear wave solution

In this test case an almost linear wave is considered and we check the order of convergence of the methods and examine the time step restrictions. We will start with the 1D SWE (2.38a) and (2.38b) without source terms (i.e. flat bottom) and look at their linearized form. By assuming $H = H_0 + \tilde{H}$ and $u = u_0 + \tilde{u}$, with H_0 and u_0 constants and $\tilde{H}$ and $\tilde{u}$ small perturbations we find the linearized equations

$$\partial_t \tilde{H} + u_0 \partial_x \tilde{H} + H_0 \partial_x \tilde{u} = 0, \tag{7.5}$$

$$\partial_t \tilde{u} + u_0 \partial_x \tilde{u} + g \partial_x \tilde{H} = 0, \tag{7.6}$$

where only the first order terms in $\tilde{H}$ and $\tilde{u}$ are taken into account. Taking $u_0 = 0$, and manipulating the equations by differentiating in time or space, multiplying by g or H_0 and adding and subtracting give the following uncoupled equations

$$\partial_{tt} \tilde{H} - c_0^2 \partial_{xx} \tilde{H} = 0, \tag{7.7}$$

$$\partial_{tt} \tilde{u} - c_0^2 \partial_{xx} \tilde{u} = 0, \tag{7.8}$$

with $c_0 = \sqrt{gH_0}$. A general solution of (7.7) and (7.8) is of the form

$$\Phi(x,t) = \frac{1}{2}\Phi_0(x - c_0 t) + \frac{1}{2}\Phi_0(x + c_0 t), \quad \Phi = \{\tilde{H}, \tilde{u}\}, \tag{7.9}$$

with given initial condition $\Phi_0(x)$, an arbitrary function of x. So the solution exists of two waves travelling in opposite direction with speed c_0 and half the amplitude of the initial perturbation. We start with a small and symmetric initial perturbation of the form

$$\tilde{H}_0(x) = \beta e^{-\gamma(x - x_p)^2}, \tag{7.10}$$

with β the maximal amplitude of the perturbation, γ a measure for the width of the perturbation and x_p the location of the centre of the perturbation. The value of β has to satisfy $\beta \ll H_0$, otherwise the linear approximation is not valid.

Convergence We first look at the order of convergence of both methods. The order of convergence can be found by looking at the factor r with which the error in the numerical solution decreases when the grid size is divided by a factor 2. So if E is the error between the numerical and the exact solution in either the L_1 or L_∞ norm, then the order of convergence p of the numerical method can be found through

$$r = \frac{E(2\Delta x)}{E(\Delta x)} = 2^p. \tag{7.11}$$

The following values for the parameters are used: $(N - 1) = \{20, 40, 80, 160, 320, 640\}$, $(N_{cx} - 1) = 12800$, $x_1 = 0\,m$, $x_{N_{el}} = 100\,m$, $x_p = 50\,m$, $H_0 = 10\,m$, $\beta = 0.001\,m$, $\gamma = 0.01\,m^{-2}$, $T_{end} = 1\,s$, $\theta_i = 0.51$ for $i = \{d, c, g\}$, $\sigma_{\text{RKDG}} = 0.3$ for the first order RKDG-method and $\sigma = 1.6 \cdot 10^{-3}$ for the second order RKDG method and we used the same time step for the CCSM-method. The time-steps are chosen small enough to be able to look at the convergence without considering the contribution of convergence in the time-step. The second order RKDG-method is used without slope limiter, because the solution is almost linear and slope limiting is not necessary. In the CCSM-method Picard linearization is sufficient, because the changes of the velocity in time are small, and the central approach to discretize the advection term in the

continuity equation is applied, because the problem is subcritical and does not involve great changes in the velocity. Since the boundaries are chosen far enough from the domain of interest, transmissive boundary conditions are sufficient for all the variables at both boundaries.

The errors in the total water depth when the gridsize is decreased with a factor 2 for the first order RKDG-method are given in Table 7.1, for the second order RKDG-method without slope limiter in Table 7.2 and for the CCSM-method in Table 7.3.

Table 7.1: Convergence results in the total water depth for the first order RKDG-method for the linear wave.

$N-1$	E_1	r	p	E_∞	r	p
20	$3.52 \cdot 10^{-3}$			$1.34 \cdot 10^{-4}$		
40	$2.19 \cdot 10^{-3}$	1.61	0.69	$7.60 \cdot 10^{-5}$	1.76	0.82
80	$1.11 \cdot 10^{-3}$	1.96	0.97	$4.34 \cdot 10^{-5}$	1.75	0.81
160	$6.16 \cdot 10^{-4}$	1.81	0.86	$2.60 \cdot 10^{-5}$	1.67	0.74
320	$3.02 \cdot 10^{-4}$	2.04	1.03	$1.25 \cdot 10^{-5}$	2.08	1.06
640	$1.51 \cdot 10^{-4}$	2.01	1.01	$5.67 \cdot 10^{-6}$	2.20	1.14

Table 7.2: Convergence results in the total water depth for the second order RKDG-method without slope limiter for the linear wave.

$N-1$	E_1	r	p	E_∞	r	p
20	$4.58 \cdot 10^{-4}$			$2.77 \cdot 10^{-5}$		
40	$1.09 \cdot 10^{-4}$	4.22	2.08	$8.59 \cdot 10^{-6}$	3.22	1.67
80	$2.61 \cdot 10^{-5}$	4.16	2.06	$2.37 \cdot 10^{-6}$	3.62	1.86
160	$6.67 \cdot 10^{-6}$	3.92	1.97	$6.19 \cdot 10^{-7}$	3.84	1.94
320	$1.94 \cdot 10^{-6}$	3.43	1.78	$1.57 \cdot 10^{-7}$	3.95	1.98
640	$1.08 \cdot 10^{-7}$	2.41	1.27	$4.53 \cdot 10^{-8}$	3.46	1.79

Table 7.3: Convergence results in the total water depth for the CCSM-method for the linear wave.

$N-1$	E_1	r	p	E_∞	r	p
20	$1.13 \cdot 10^{-3}$			$4.04 \cdot 10^{-5}$		
40	$2.60 \cdot 10^{-4}$	4.36	2.12	$9.67 \cdot 10^{-6}$	4.18	2.06
80	$6.35 \cdot 10^{-5}$	4.10	2.04	$2.44 \cdot 10^{-6}$	3.96	1.99
160	$1.54 \cdot 10^{-5}$	4.11	2.04	$5.98 \cdot 10^{-7}$	4.08	2.03
320	$3.76 \cdot 10^{-6}$	4.10	2.04	$1.43 \cdot 10^{-7}$	4.17	2.06
640	$1.08 \cdot 10^{-6}$	3.48	1.80	$4.92 \cdot 10^{-8}$	2.92	1.55

We can see that the first order RKDG-method is indeed first order, and that the second order RKDG-method converges with order two. The CCSM-method also converges with order two, where we would expect a convergence of order one because it is a first order method. This can be explained by the fact that we look at almost linear solutions. In this case the contribution of the non-linear terms is very small and the only reason why the CCSM-method is a first order method is because of the first order upwind scheme used for the non-linear advection term in the momentum equation. So if the non-linear terms are negligible the method will converge with order two.

For both the second order RKDG-method and the CCSM method the order of convergence becomes smaller when using 320 or more grid-points. This is probably caused by rounding

errors, because the error in the numerical solution is already becoming very small when using such a fine grid. Another reason can be that the exact solution is the solution to the linearized equations and the numerical methods are solving the nonlinear equations.

The graphical results for the first order RKDG-method are given in Figure 7.2, for the second order RKDG-method without slope limiter in 7.3 and for the CCSM method in Figure 7.4. The

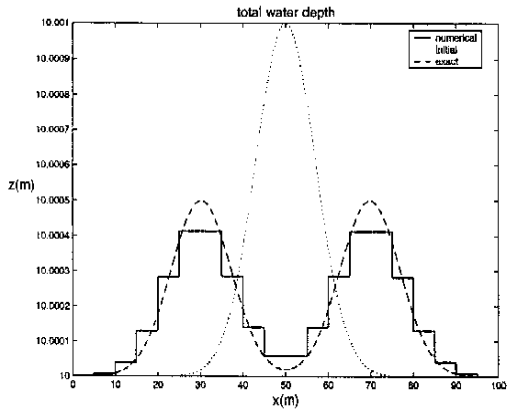


Fig. 7.2: Total water depth for $N-1=20$ and $\sigma=0.9$ for the first order RKDG method.

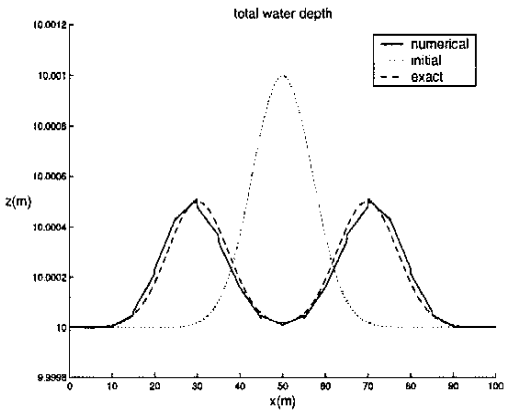


Fig. 7.3: Total water depth for $N-1=20$ and $\sigma=0.3$ for the second order RKDG method without slope limiter.

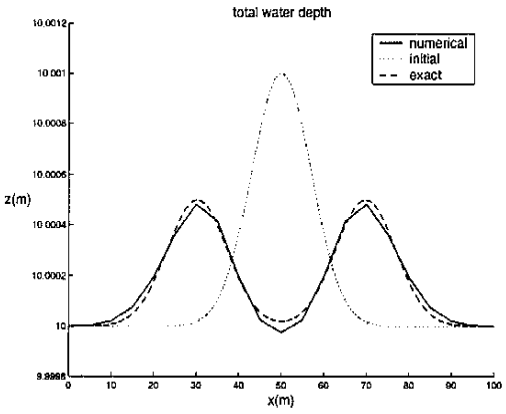


Fig. 7.4: Total water depth for $N-1=20$ for the CCSM method and using the same timestep restriction as for the second order RKDG-method.

initial condition is given by the dotted line, the dashed line represents the exact solution to the linearized equations and the numerical solution is represented by the solid line. The following values for the parameters are used: $(N-1)=20$, $T_{end}=2\,s$ and $\sigma_{\text{RKDG}}=0.9$ for the first order RKDG-method, $\sigma=0.3$ for the second order RKDG-method and we used the same time step restriction for the CCSM-method. The other parameters are kept the same. We chose for both RKDG-methods the largest possible value of σ as given in Table 5.4, because this gives the best results. Using the same value of σ ($\sigma=0.3$) for the first order method as for the second order method causes the numerical solution to become much more diffusive. However reducing the value of σ for the second order RKDG-method does not give this diffusion, so it is only a problem for the first order method.

In the figures the difference between the different approximations of the variables can be clearly seen: the discontinuous approximations for the RKDG-methods and the linear approximation for the CCSM-method. As expected the second order RKDG-method gives the best result, but the results of the CCSM-method are similar and the first order RKDG-method produces the least accurate results.

Time step variations Because the RKDG-method is an explicit method the time step is restricted by the following expression

$$\Delta t \leq \sigma \frac{\Delta x}{|u| + \sqrt{gH}}, \tag{7.12}$$

with the CFL-number $\sigma < 1$ as given in Section 5.2.2 and Table 5.4. In the CCSM-method the time step is only restricted because of accuracy reasons and because of linearization of the nonlinear terms. No theoretically time step restriction as for the RKDG-method is known, and is probably impossible to derive. We will now look what happens if we use the time step as given in (7.12) for both methods, but take a value for σ larger than one, where we will only consider the first order case. The same values for the parameters are used as in the previous section only with $(N-1) = 80$ and for σ we used the value 5. The results are shown in Figure 7.5.

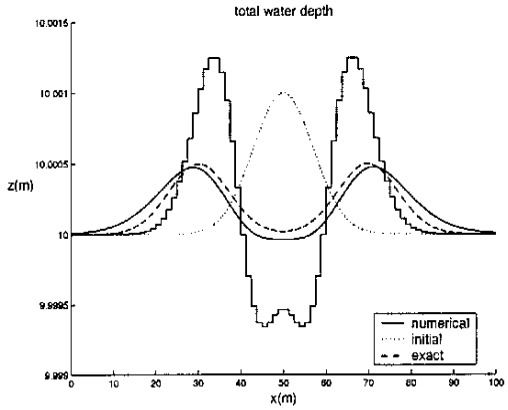


Fig. 7.5: Solutions of the first order RKDG-method (solid staircase line) and CCSM-method (solid smooth line) for $\sigma = 5$ with $(N-1) = 80$.

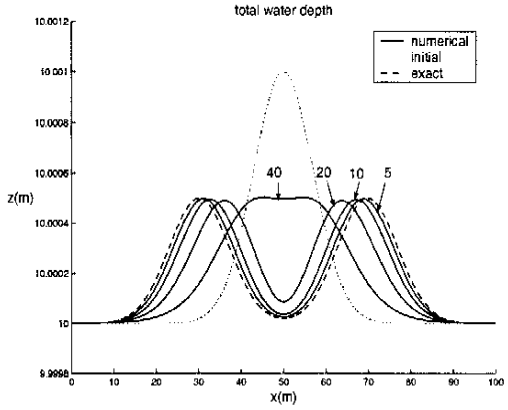


Fig. 7.6: Solutions of the CCSM-method for $\sigma = \{5, 10, 20, 40\}$ with $(N-1) = 400$.

The numerical solution of the RKDG-method is given by the solid staircase line, that of the CCSM-method by the solid smooth line and the initial and exact solution are given by the dotted and dashed line respectively. The numerical solution of the CCSM-method remains much closer to the exact solution for $\sigma = 5$ than the numerical solution of the first order RKDG-method as was expected from theory.

The numerical solutions of the CCSM-method for $(N-1) = 400$, when the value of σ is increased even more, are shown in Figure 7.6, where $\sigma = \{5, 10, 20, 40\}$. For a CFL-number up to 10, the numerical solution stays quite close to the exact solution. For $\sigma = 20$ the shape of the numerical solution is still right, but it is less accurate. For σ equal to 40 the numerical solution is not accurate anymore. So for this linear problem the time step for the CCSM-method can be chosen 10 times as large as the one for the RKDG-method to get still a quite accurate solution.

For the CCSM-method with upwind approach in the continuity equation, the CFL-number cannot be increased that much. Only values up to $\sigma = 1.4$ give good results. This can be caused by the fact that a simple upwind scheme is used. When the velocity changes sign, this can cause

oscillations in the numerical solution. A possible way to overcome this is by using the Engquist-Osher flux for the scalar case, see Engquist & Osher (1981), instead of the simple upwind scheme, but this is not investigated any further in this thesis. There is also no explanation for the fact that no errors occur with the CCSM-method with the central approach in the continuity equation, because there also the upwind approach is used for the advection term in the momentum equation.

Conclusions The first order RKDG-method converges with order one and the second order RKDG-method is indeed second order for this linear test case. The CCSM-method is second order in space because of the linearity of the solution.

The second order RKDG-method and the CCSM-method give similar results and the first order RKDG-method gives the least accurate results.

By increasing the time step, the first order RKDG-method produces wrong results when $\sigma > 1$ as expected from theory. The time step for the CCSM-method can be chosen a factor 10 larger and still produce quite accurate results.

7.3 Transformation to Burgers equation

Solutions of the inviscid Burgers equation may contain discontinuities. In this section we rewrite the 1D SWE to the invicid Burgers equation and examine the transition in time of a solution. We will look at the SWE written in the form of equation (2.51) without source terms (i.e. flat bottom), with the Riemann invariants as the variables given by

$$\partial_t(u - 2c) + (u - c)\partial_x(u - 2c) = 0, \tag{7.13a}$$
$$\partial_t(u + 2c) + (u + c)\partial_x(u + 2c) = 0. \tag{7.13b}$$

The special case where the Riemann invariant $u+2c$ is equal to a given constant K is considered. Then equation (7.13b) is always satisfied and (7.13a) can be written in the following form

$$\partial_t(K - 4c) + (K - 3c)\partial_x(K - 4c) = 0. \tag{7.14}$$

Because K is constant we can derive from (7.14) that

$$\partial_t c + (K - 3c)\partial_x c = 0. \tag{7.15}$$

Subtracting (7.15) from (7.14) gives

$$\partial_t(K - 3c) + (K - 3c)\partial_x(K - 3c) = 0, \tag{7.16}$$

which is equal to the invicid Burgers equation

$$\partial_t v + v\partial_x v = 0 \tag{7.17}$$

with $v = K - 3c$. The solution of the invicid Burgers equation (7.17) can be written implicitly as

$$v = v_0(x - vt), \tag{7.18}$$

with given initial condition $v_0(x)$, an arbitrary function of x. The value for H is then given by

$$H = \left(\frac{K - v}{3\sqrt{g}}\right)^2, \tag{7.19}$$

as can be calculated from the relation $v = K - 3c$, with $c = \sqrt{gH}$, and the discharge q can be derived from $u + 2c = K$ and $q = uH$ to be

$$q = KH - 2H\sqrt{gH}. \tag{7.20}$$

If we start with smooth initial data $v_0(x)$ for which $v_0'(x)$ is somewhere negative, then the wave will break at time

$$T_b = -\frac{1}{\min v_0'}, \tag{7.21}$$

which means that the solution contains a discontinuity at that time. We will start with the initial condition

$$v_0 = \sin\left(\frac{2\pi x}{L}\right), \tag{7.22}$$

with $L = x_N - x_0$ the length of the domain, and then the breaking time will be at $T_b = L/2\pi \approx 0.159L$.

Boundary conditions At the left boundary the flow is directed inwards and for this reason we use a Dirichlet boundary condition at the left boundary. We know the exact solution in this point, which is given by (7.19) and (7.20) with $v = 0$ and these values will be imposed. At the right boundary the flow is directed outwards so we can simply apply transmissive boundary conditions.

Convergence of CCSM-method We look again at the convergence of the CCSM-method, because this time the equations contain a reasonable amount of nonlinearity. The end time of the computation is taken far before the breaking time T_b, so the solution will still be smooth, the following values for the parameters are used: $x_0 = 0\,m$, $x_{N_{el}} = 1\,m$, $T_{end} = 0.5T_b$, $\theta_i = 0.51$ for $i = \{d, c, g\}$, $K = 2\,m/s$ and $\sigma = 0.01$ for the time step restriction given in (7.12). The results are given in Table 7.4 for the total water depth. The convergence of the CCSM-method

Table 7.4: Convergence results in the total water depth for the first order CCSM method for the Burgers equation.

$N-1$	E_1	r	p	E_∞	r	p
20	$9.41 \cdot 10^{-4}$			$4.32 \cdot 10^{-3}$		
40	$4.33 \cdot 10^{-4}$	2.17	1.12	$2.18 \cdot 10^{-3}$	1.98	0.99
80	$2.27 \cdot 10^{-4}$	1.91	0.93	$1.12 \cdot 10^{-3}$	1.94	0.96
160	$1.13 \cdot 10^{-4}$	2.01	1.01	$5.58 \cdot 10^{-4}$	2.01	1.01
320	$5.19 \cdot 10^{-5}$	2.18	1.12	$2.62 \cdot 10^{-4}$	2.13	1.09
640	$2.55 \cdot 10^{-5}$	2.04	1.03	$1.27 \cdot 10^{-4}$	2.06	1.04

is in this case indeed of order one.

Evolution in time In Figures 7.7 till 7.11 the evolution in time is given until the breaking time T_b for the different methods and in Table 7.5 the errors in the total water depth and velocity are given at the breaking time T_b. The following values for the parameters are used for the computation: $(N-1) = 40$, $x_0 = 0\,m$, $x_{N_{el}} = 10\,m$, $T_{end} = \{0.33T_b, 0.66T_b, T_b\}$ where $T_b \approx 1.59\,s$, $\theta_i = 0.51$ for $i = \{d, c, g\}$, $K = 2\,m/s$, $M = 0$ for the TVB correction constant

of the slope limiter and $\sigma = 0.3$ for the time step restriction given in (7.12) for the second order RKDG-method and the CCSM-method and $\sigma = 0.9$ for the first order RKDG-method. Figure 7.7 shows the solution for the first order RKDG-method, Figure 7.8 for the second order RKDG-method without and Figure 7.9 with slope limiter. The solution of the CCSM-method with central approach in the continuity equation and $\theta_i = 0.51$ is shown in Figure 7.10 and with upwind approach and $\theta_i = 1$ in Figure 7.11.

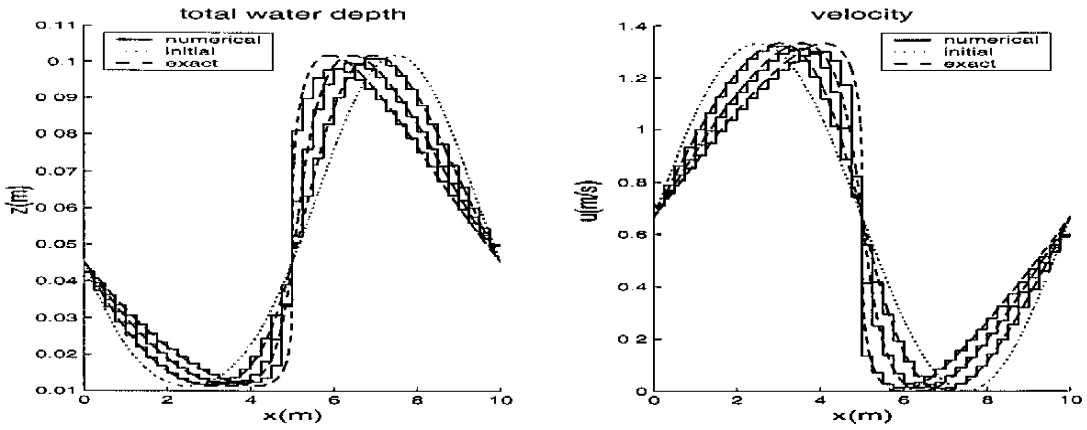


Fig. 7.7: Solutions of the first order RKDG-method with $N - 1 = 40$ and $\sigma = 0.9$ for the Burgers equation.

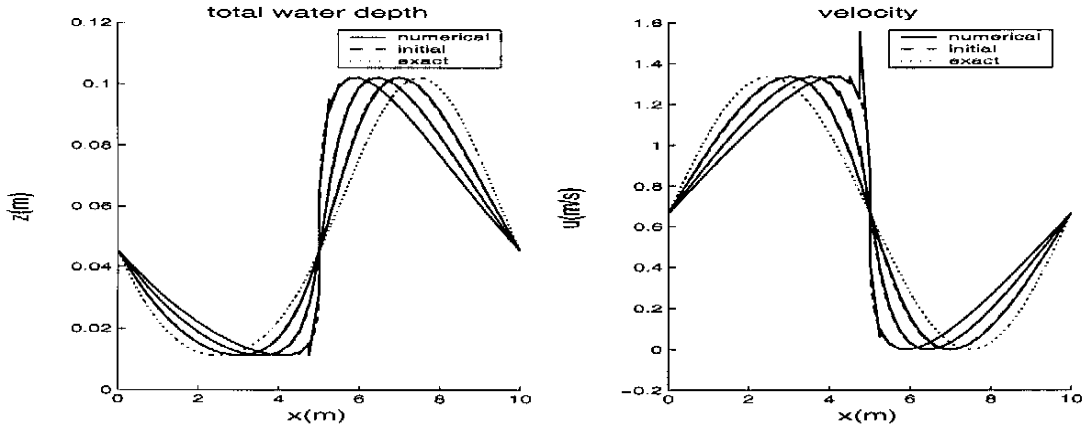


Fig. 7.8: Solutions of the second order RKDG-method without slope limiter with $N-1 = 40$ and $\sigma = 0.3$ for the Burgers equation.

Table 7.5: Errors of the methods at time T_h for the total water depth and the velocity. The addition '+ S' stands for 'with slope limiter'.

Method	E_1 water depth	E_∞ water depth	E_1 velocity	E_∞ velocity
first order RKDG	$1.94 \cdot 10^{-2}$	$3.57 \cdot 10^{-2}$	$3.14 \cdot 10^{-1}$	$5.34 \cdot 10^{-1}$
second order RKDG	$2.18 \cdot 10^{-3}$	$1.98 \cdot 10^{-2}$	$4.29 \cdot 10^{-2}$	$3.27 \cdot 10^{-1}$
second order RKDG + S	$3.03 \cdot 10^{-3}$	$1.89 \cdot 10^{-2}$	$5.40 \cdot 10^{-2}$	$2.55 \cdot 10^{-1}$
central CCSM	$1.89 \cdot 10^{-2}$	$3.40 \cdot 10^{-2}$	$1.00 \cdot 10^{-0}$	$2.35 \cdot 10^{-0}$
upwind CCSM	$2.74 \cdot 10^{-2}$	$2.34 \cdot 10^{-2}$	$5.07 \cdot 10^{-1}$	$4.93 \cdot 10^{-1}$

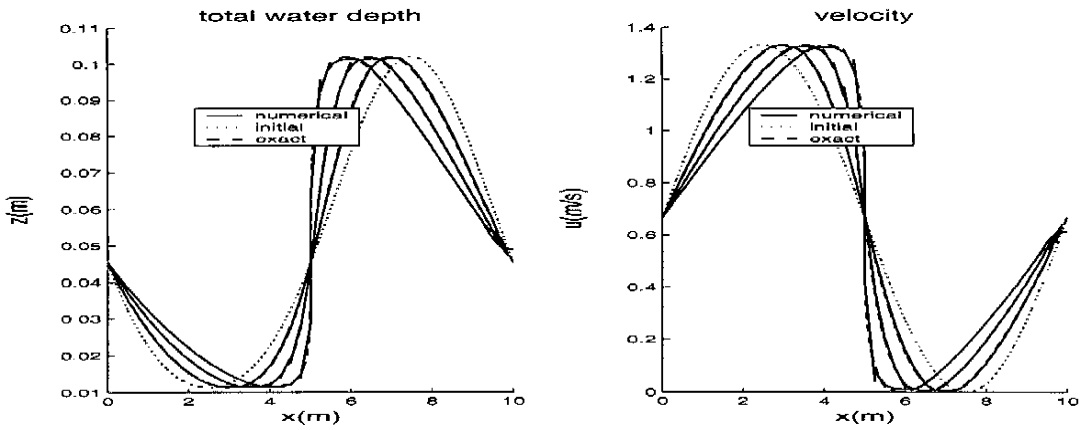


Fig. 7.9: Solutions of the second order RKDG-method with slope limiter with $N-1=40$ and $\sigma=0.3$ for the Burgers equation.

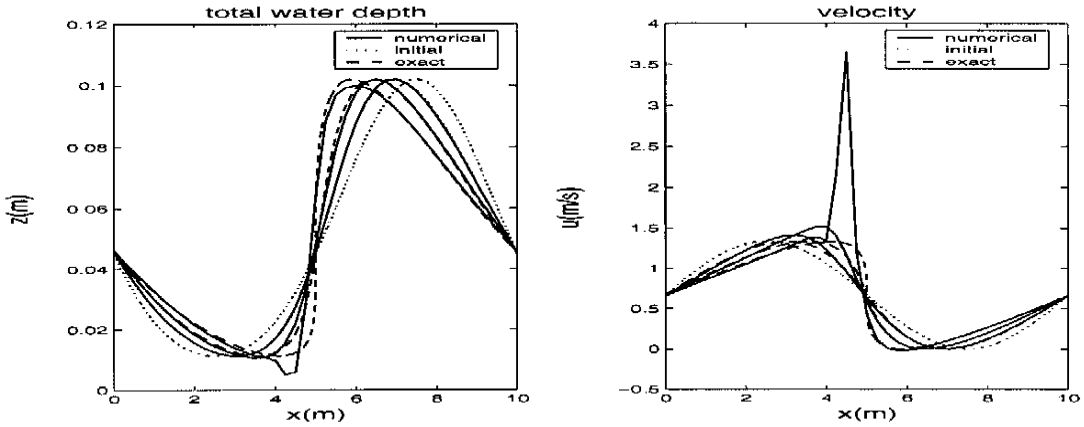


Fig. 7.10: Solutions of the central CCSM-method with $N-1=40$, $\theta_i=0.51$ and $\sigma=0.3$ for the Burgers equation.

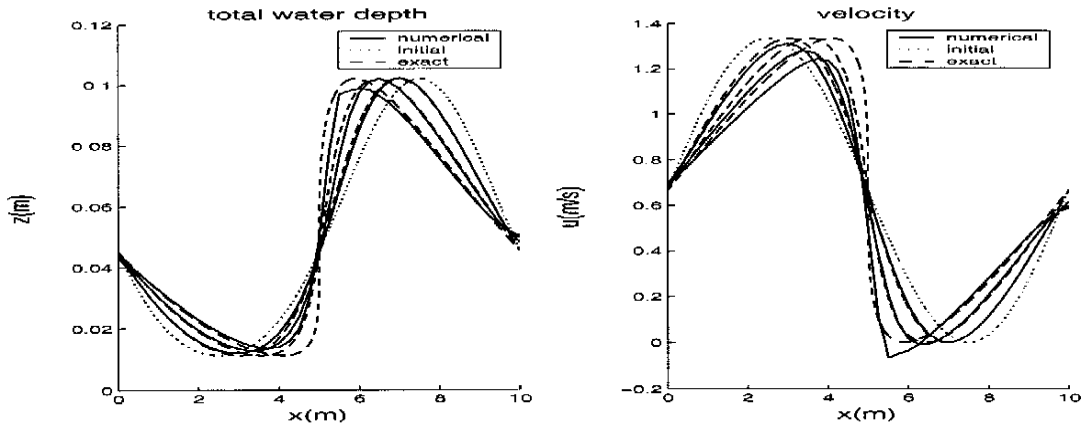


Fig. 7.11: Solutions of the 'fully implicit' upwind CCSM-method with $N-1=40$ and $\sigma=0.3$ for the Burgers equation.

Using a slope limiter in the second order RKDG-method improves mainly the L_∞-error, but the L_1-error becomes slightly bigger, so the slope limiter has a diffusive effect. This can also be seen in Figure 7.9 where the solution is smoother than in Figure 7.8, where no slope limiter is used.

Configuration of the CCSM-method In the present paragraph features of the CCSM-method are changed to see what effect they have on the numerical solution where we look at the solution at time T_b.

When using the upwind approach in the CCSM-method for the advection term in the continuity equation, the solution gets smoother, but also more diffusive. The L_1-norm of the water depth becomes slightly bigger but the other errors are smaller, especially for the velocity.

Using an iterative process instead of Picard linearization in case of the central approach makes the solutions a little bit more accurate, only the L_∞-norm of the velocity becomes larger. When using the upwind approach the numerical solution gets slightly worse when an iteration process with one step is used. Using more then one iteration step does not improve the solution much more.

Applying the 'fully implicit' method ($\theta_i = 1$ for $i = \{d, c, g\}$) the solution of the method with the upwind approach in the continuity equation becomes again slightly better. Making the method 'fully explicit' ($\theta_i = 0$ for $i = \{d, c, g\}$) gives also slightly better results than for $\theta_i = 0.51$, but worse then when the 'fully implicit' method is used.

Conclusions All methods are able to cope with the steepening of the wave, but with the second order RKDG-method a slope limiter has to be used and the upwind approach has to be used in the CCSM-method to avoid overshoots in the neighbourhood of large gradients.

As expected, the second order RKDG-method with slope limiter gives the best results. The first order RKDG-method and the CCSM-method with upwind approach give both quite accurate results with an error of the same order of magnitude, but they are both more diffusive than the second order RKDG-method.

When the end time of the computation is far before T_b then the CCSM-method converges with order one.

7.4 Riemann problems

In this section two test problems are presented that can be solved exactly. The initial data for the Riemann problem is given by

$$\mathbf{u}_0 = \begin{cases} \mathbf{u}_L & \text{if } x < x_p, \\ \mathbf{u}_R & \text{if } x > x_p, \end{cases} \tag{7.23}$$

with x_p the location of the initial discontinuity. The solution to this problem is calculated by using the exact Riemann solver given in Toro (2001). In Table 7.6 the data are given for two different test problems as given in Schwanenberg (2003) and Toro (2001). The first test case is

Table 7.6: Data for the two test problems.

Test	$H_L(m)$	$u_L(m/s)$	$H_R(m)$	$u_R(m/s)$	$L(m)$	$x_p(m)$	$T_{end}(s)$
1	6.0	0.0	2.0	0.0	2500	1500	70
2	1.0	-5.0	1.0	5.0	80	40	2

a so called dam break problem and the second one contains two rarefractions and a nearly dry

bed. The main goal of these test cases is to investigate how well the numerical schemes deal with the initial discontinuity.

7.4.1 Test 1: Dam break problem

This test case mimics a dam break problem in 1D. The initial condition consits of two areas with a different water depth and no flow, separated by a so called dam located at x_p. The water level at the left side of the dam will be higher than that on the right side. At $t = 0$ the dam is removed and the water starts to flow. The solution consists of a shock wave propagating to the right, a rarefraction wave to the left and a constant water level in between.

The number of elements we use is $(N - 1) = 40$, transmissive boundary conditions are implemented, the TVB correction constant is taken to be $M = 0$ and $\sigma = 0.3$ for the second order RKDG-method and the CCSM-method and $\sigma = 0.9$ for the first order RKDG-method. The values of the other parameters are listed in Table 7.6.

Figures 7.12 till 7.16 show the numerical solutions at $T_{end} = 70\,s$ for the different numerical methods. Figure 7.12 shows the solution for the first order RKDG-method, Figure 7.13 for the second order RKDG-method without and Figure 7.14 with slope limiter. The solution of the CCSM-method with central approach in the continuity equation and $\theta_i = 0.51$ is shown in Figure 7.15 and with upwind approach, one iteration step and $\theta_i = 1$ in Figure 7.16. All the

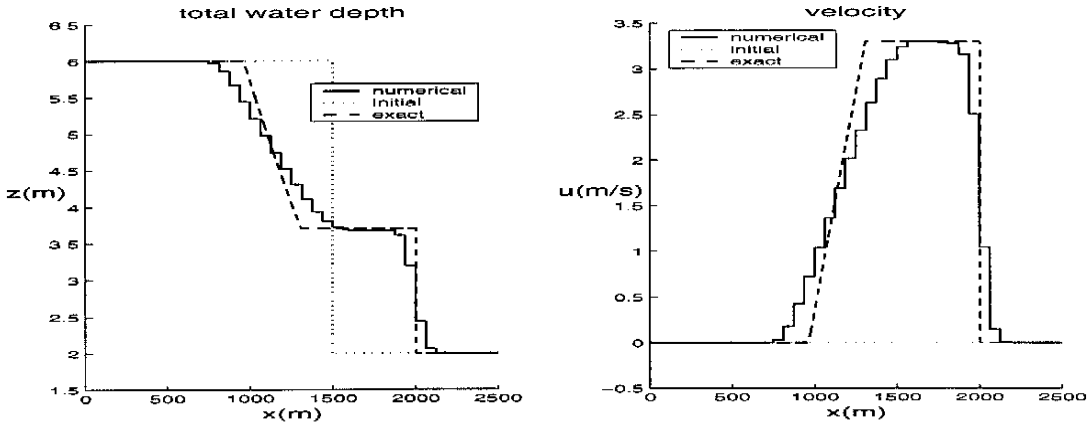


Fig. 7.12: Test 1: Total water depth and velocity for the first order RKDG method for $N - 1 = 40$ and $\sigma = 0.9$.

methods can cope with the discontinuous problem. Again the smoothening effect of the slope limiter can be seen in 7.14 for the second order RKDG-method.

Configuration of the CCSM-method In this paragraph we change features of the CCSM-method to see what effect they have on the numerical solution.

When using the upwind approach in the CCSM-method, the solution gets more accurate, but also more diffusive.

Using an iterative process instead of Picard linearization in case of the central approach makes the solutions more accurate, especially for the L_1-norms. When using the upwind approach the numerical solution also gets better when an iteration process with one step is used. Using more then one iteration step does not improve the solution very much more.

Applying the 'fully implicit' method ($\theta_i = 1$ for $i = \{d, c, g\}$) the solution of the method with the upwind approach becomes smoother but also more diffusive. Making the method fully explicit ($\theta_i = 0$ for $i = \{d, c, g\}$) deteriorates the solution.

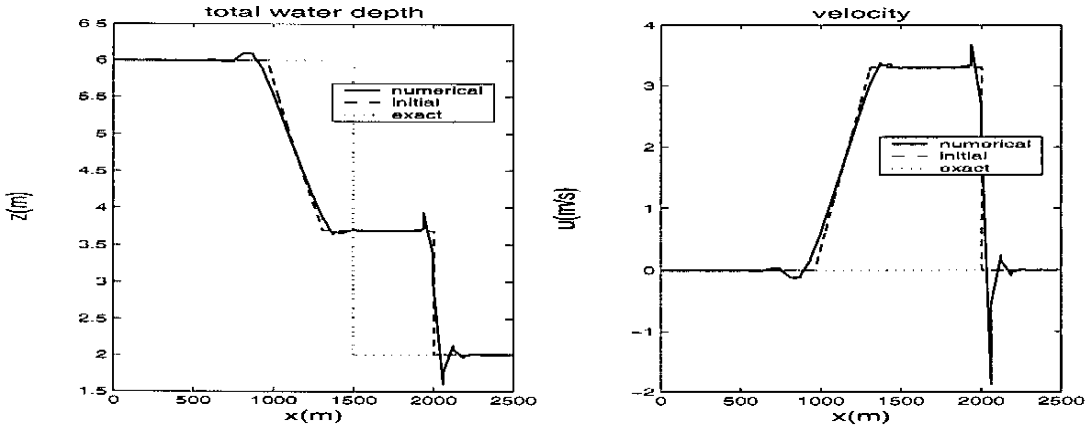


Fig. 7.13: Test 1: Total water depth and velocity for the second order RKDG method without slope limiter for $N-1=40$ and $\sigma=0.3$.

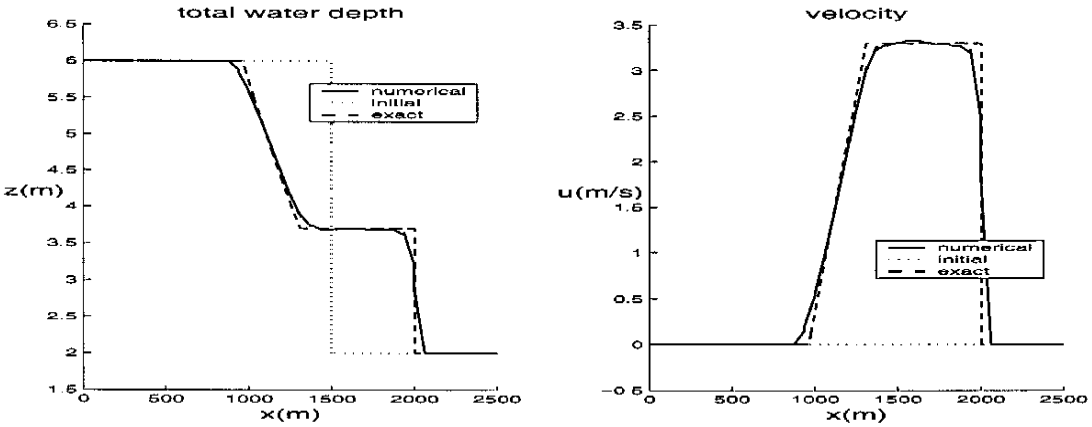


Fig. 7.14: Test 1: Total water depth and velocity for the second order RKDG method with slope limiter for $N-1=40$ and $\sigma=0.3$.

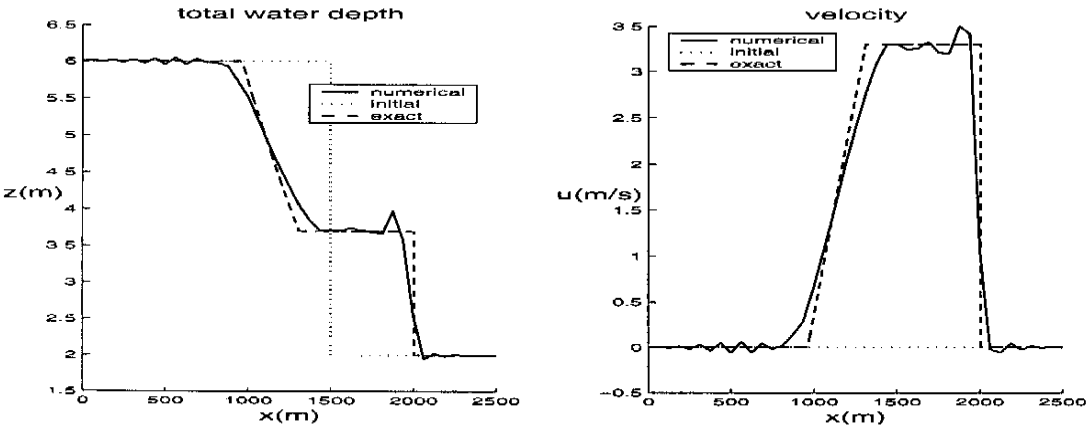


Fig. 7.15: Test 1: Total water depth and velocity for the CCSM method with central approach and Picard linearization for $N-1=40$ and $\sigma=0.3$.

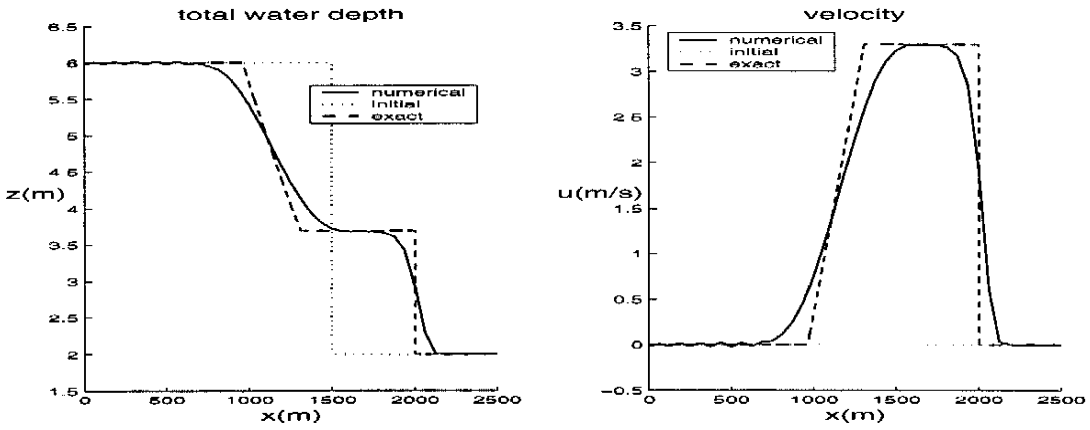


Fig. 7.16: Test 1: Total water depth and velocity for the 'fully implicit' CCSM method with upwind approach and one iteration step for $N-1=40$ and $\sigma=0.3$.

In Figure 7.16 the solution of the 'fully implicit' CCSM-method with upwind approach is given. Comparing it with the solution of the CCSM-method with central approach, Picard linearization an $\theta_i = 0.51$ shows indeed the diffusive effect of the upwind approach and the 'fully implicit' procedure.

The cause of the wiggles most visible around $x = 500$ in Figure 7.16 is not known yet. The wiggles propagate from the initial discontinuity to the left and right. Lumping the mass matrix removes these wiggles but gives a less accurate solution in the rest of the domain. When the initial condition for the CCSM-method is taken to be the exact solution of the problem at $T = 9\,s$ or later, then there are no wiggles anymore. So the wiggles are probably caused by the initial discontinuity.

Conclusions The second order RKDG-method with slope limiter gives again the best solution, but needs a slope limiter to avoid over- and undershoots. The 'fully implicit' CCMS-method with upwind approach and one iteration step gives practically the same result as the first order RKDG-method. Only some small wiggles which are probably caused by the initial discontinuity are generated by the CCSM-method.

7.4.2 Test 2: Two rarefractions and nearly dry bed

This test case is performed because it tests the ability of a method to deal with very shallow water depths. The solution consists of two rarefractions waves travelling in opposite direction leaving a very shallow water level in between. It turns out that this test does not crash with the first order RKDG-method, the second order RKDG-method with slope-limiter and the CCSM-method with upwind approach. If no slope-limiter is used in the second order RKDG-method or the central approach is used in the CCSM-method, the water depth becomes negative and this causes a run-time error in the computation.

Again $(N-1) = 40$ elements and transmissive boundaries are used. For the TVB correction constant we take the value $M = 0.01$ and for the CFL-numbers we use $\sigma = 0.3$ for the second order RKDG-method and the CCSM-method and $\sigma = 0.9$ for the first order RKDG-method. The values of the other parameters are listed in Table 7.6.

Figure 7.17 shows the solution for the first order RKDG-method and Figure 7.18 for the second order RKDG-method with slope limiter. The solution of the CCSM-method with upwind approach in the continuity equation and $\theta_i = 0.51$ is shown in Figure 7.19.

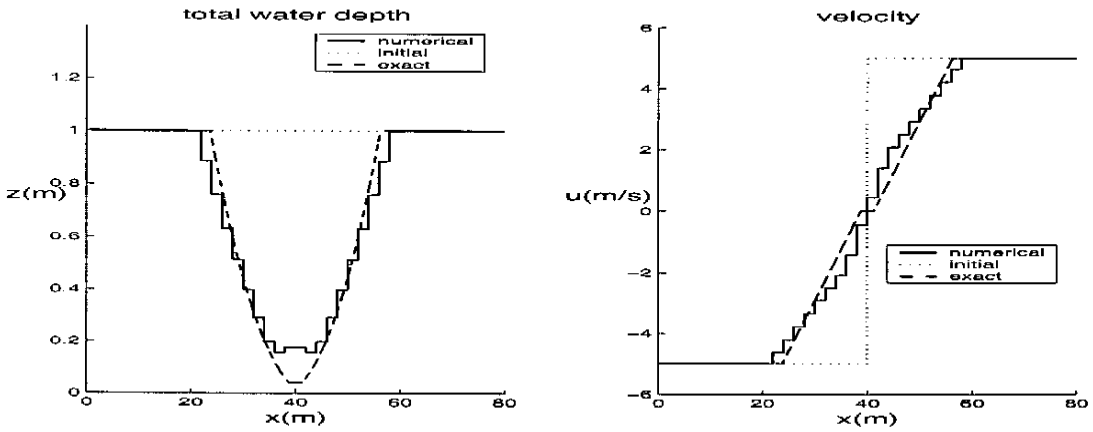


Fig. 7.17: Test 2: Total water depth and velocity for the first order RKDG method for $N-1=40$ and $\sigma=0.9$.

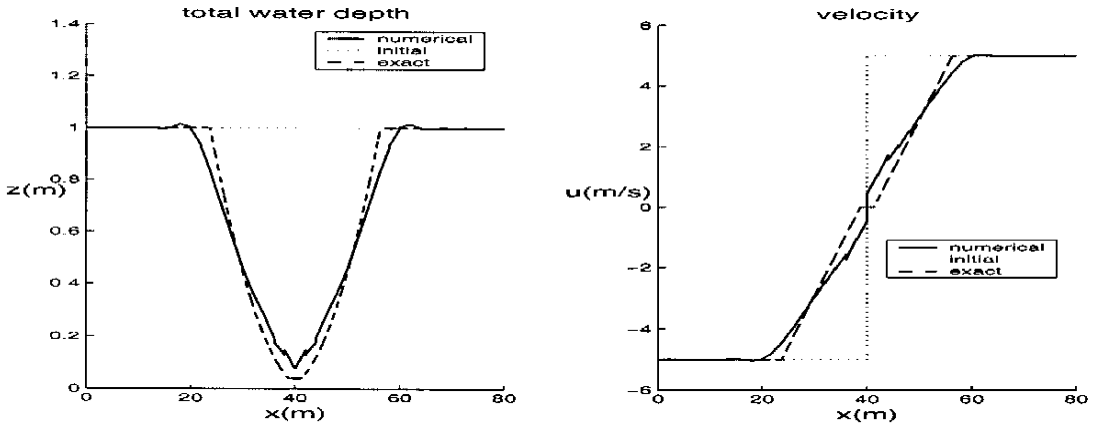


Fig. 7.18: Test 2: Total water depth and velocity for the second order RKDG method with slope limiter for $N-1=40$ and $\sigma=0.3$.

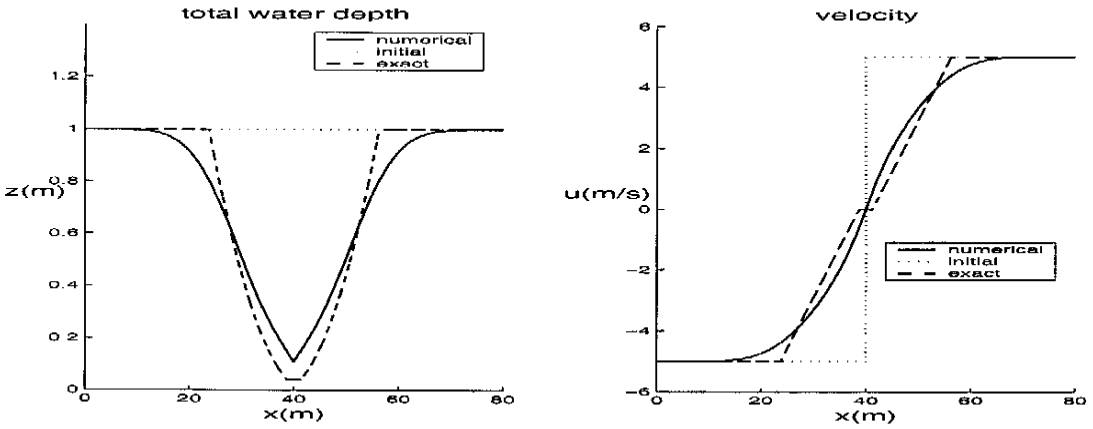


Fig. 7.19: Test 2: Total water depth and velocity for the CCSM method with upwind approach for $N-1=40$ and $\sigma=0.3$.

For the upwind CCSM-method the configuration with one iteration step and $\theta_i = 0.51$ for $i = \{d, c, g\}$ gives the best results. Making the method fully implicit does not improve the solution.

Conclusions A slope limiter has to be used for the second order RKDG-method to avoid negative water depths and this method gives as usual the best results. The first order RKDG-method does not stay behind much and the upwind CCSM-method with one iteration step gives also good results but is a bit more diffusive.

7.5 Tidal movement

Tidal movement typically involves low Froude numbers. To mimic tidal movement we consider a domain with a flat bottom and an initial flat water level with no flow. At $t = 0$ a sine-shaped boundary condition is imposed at the left boundary for both the total water depth, H_L, and the discharge, q_L. They are given by the following relations

$$H_L = H_0 + A_H \sin\left(\frac{2\pi t}{T}\right), \tag{7.24}$$

$$q_L = A_q \sin\left(\frac{2\pi t}{T}\right), \tag{7.25}$$

with H_0 the initial total water depth, A_H and A_q the amplitudes of the imposed sines and T the period of the tidal movement. Given the amplitude of the total water depth, the amplitude for the discharge must fulfil the following relation

$$A_q = A_H c_0, \tag{7.26}$$

with $c_0 = \sqrt{gH_0}$ the speed of the tidal wave, as can be derived from the linearized equations.

The typical period for tidal movement is $T = 12\,h$ and the initial total water depth is taken to be $H_0 = 50\,m$. The wavelength is given by $\lambda = c_0 T \approx 957\,km$. The length of the domain is chosen to be $L = 3500\,km$, so that two tidal periods ($T_{end} = 24\,h$) can be examined. For the amplitude of the total water depth we take $A_H = 5\,m$ and this gives $A_q \approx 110.7\,m^2/s$.

Using the imposed boundary conditions for the CCSM-method as described in Section 5.1 causes the generation of $2\Delta x$-waves, also called odd-even decoupling, as shown in Figure 5.2. To overcome this problem we impose the boundary condition on the first grid-point in the domain and also a boundary condition, modified for the shift in place, at the real boundary. As a consequence, the place of the imposed boundary condition in the CCSM-method is the same as for the RKDG-method, namely in the first grid point.

There exists no exact solution to this problem, but according to the Burgers test case of Section 7.3 we expect the sine to move into the domain and then to steepen, because the problem is nonlinear. The solution calculated by the second order RKDG-method with slope limiter on a fine grid ($(N-1) = 1600$) is used as the approximation to the exact solution.

In Figure 7.20 the numerical solution of the first order RKDG-method is shown, in Figure 7.21 the second order RKDG-method with slope limiter, in Figure 7.22 the central CCSM-method and in Figure 7.23 the upwind CCSM-method with one iteration and $\theta_i = 0.55$. For the first order methods a timestep with $\sigma = 0.9$ is used and for the second order RKDG-method $\sigma = 0.3$. The TVB correction constant is taken to be $M = 0$.

For the upwind CCSM-method at least one iteration step is needed, otherwise the solution gets unstable. Taking the upwind CCSM-method 'fully implicit' makes the method very diffusive, but some diffusion is necessary to get rid of unwanted oscillations. For this reason the

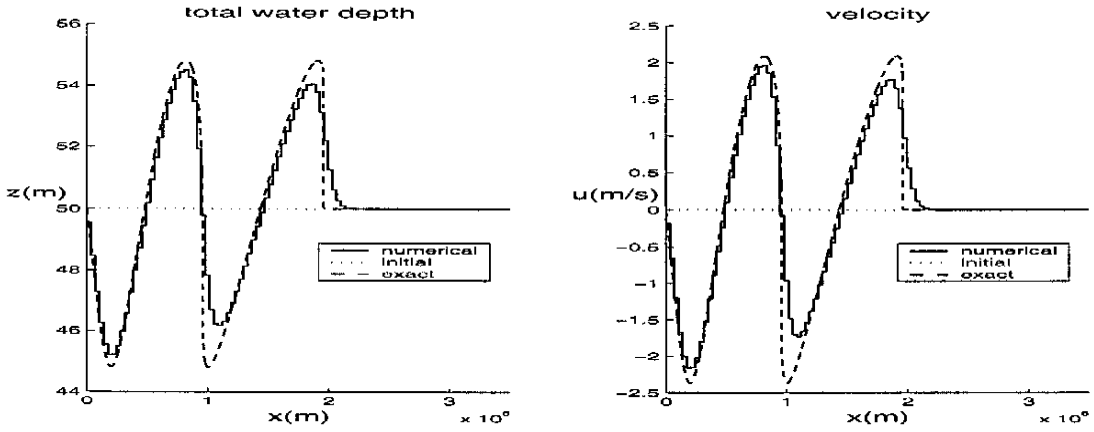


Fig. 7.20: Solution of the first order RKDG-method for $N-1=100$ at $T_{end}=24\,h$ and $\sigma=0.9$ for the tidal wave problem.

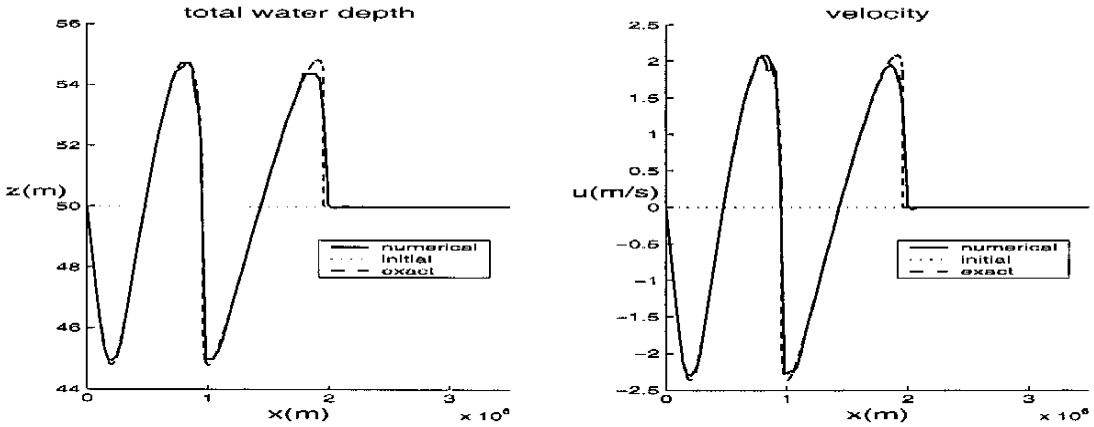


Fig. 7.21: Solution of the second order RKDG-method with slope limiter for $N-1=100$ at $T_{end}=24\,h$ and $\sigma=0.3$ for the tidal wave problem.

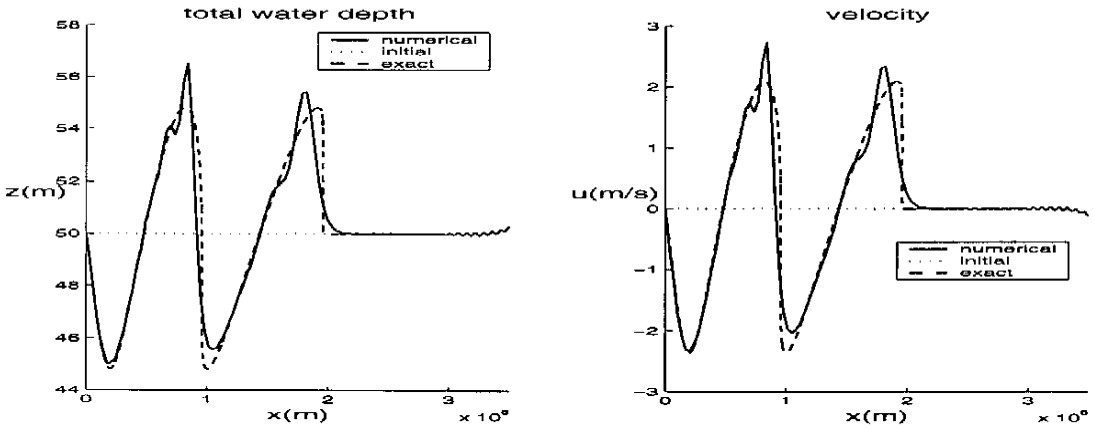


Fig. 7.22: Solution of the central CCSM-method for $N-1=100$ at $T_{end}=24\,h$ and $\sigma=0.9$ for the tidal wave problem.

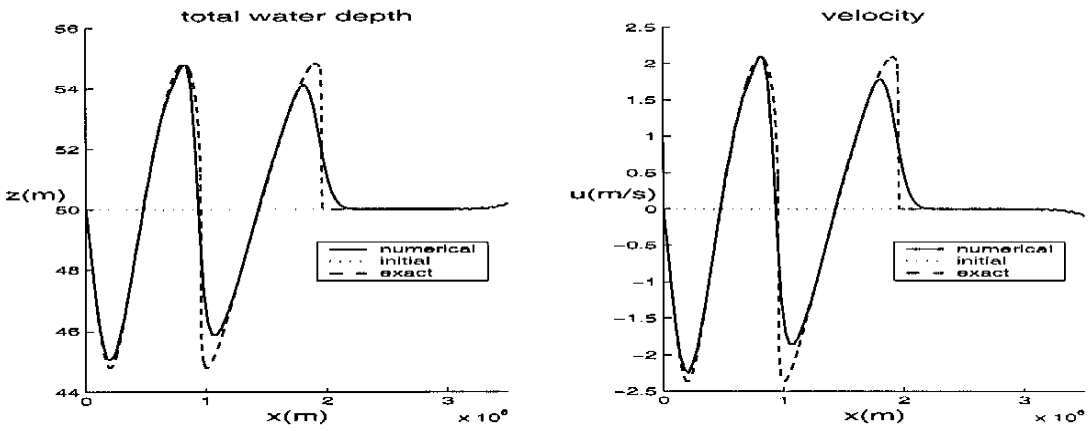


Fig. 7.23: Solution of the upwind CCSM-method with one iteration step and $\theta_i = 0.55$ for $N - 1 = 100$ at $T_{end} = 24\,h$ and $\sigma = 0.9$ for the tidal wave problem.

values of the θ_i's are slightly increased to $\theta_i = 0.55$ for $i = \{d, c, g\}$. In both Figure 7.22 and 7.23 wiggles appear at the right boundary. The cause of these wiggles is not known yet, but they are probably caused by the boundary condition. The solution of thus obtained CCSM-method, shown in Figure 7.23, is practically the same as the solution of the first order RKDG-method, depicted in Figure 7.20, but the solution of the second order RKDG-method with slope limiter gives better results.

Increasing the time step substantially is still not possible for the upwind CCSM-method, see also the paragraph 'Time step variation' in Section 7.2. So we look at the central CCSM-method and use one iteration step and $\theta_i = 0.55$ for $i = \{d, c, g\}$ to make the solution more accurate. The results for $\sigma = \{1, 2, 4\}$ when $A_H = 5\,m$ are shown in Figure 7.24 and for a smaller amplitude, $A_H = 0.5\,m$, in Figure 7.25.

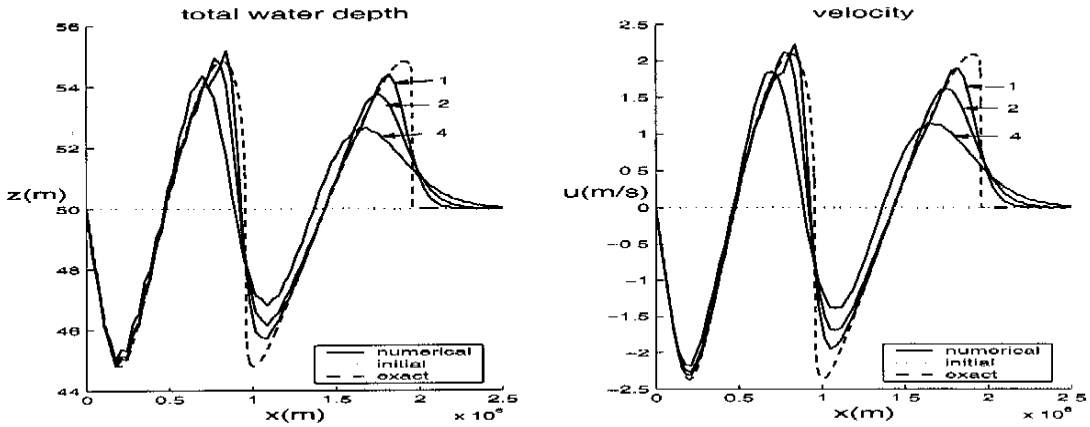


Fig. 7.24: Solution to the tidal wave problem for different values for the time step, $\sigma = \{1, 2, 4\}$ for the central CCSM-method with one iteration step and $\theta_i = 0.55$ for $N - 1 = 100$ at $T_{end} = 24\,h$ and $A_H = 5\,m$.

As can be seen the solution gets more diffusive if the time step is increased also with a small amplitude, but the shape of the solution remains right. So if only a global idea of the wave structure is necessary a larger time step can be used.

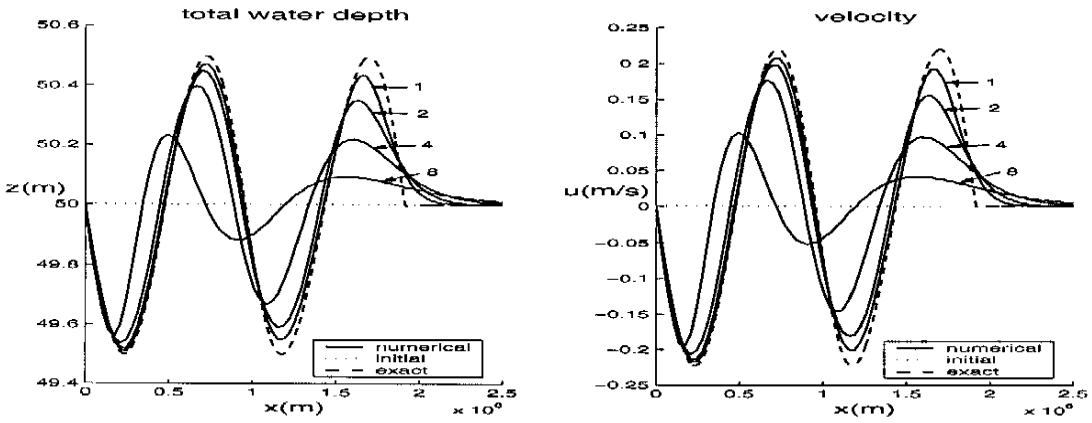


Fig. 7.25: Solution to the tidal wave problem for different values for the time step, $\sigma = \{1, 2, 4, 8\}$ for the central CCSM-method with one iteration step and $\theta_i = 0.55$ for $N - 1 = 100$ at $T_{end} = 24\,h$ and $A_H = 0.5\,m$.

Conclusions An extra boundary condition has to be imposed for the CCSM-method to avoid odd-even decoupling.

The first order RKDG-method and the upwind CCSM-method using one iteration step and $\theta_i = 0.55$ give practically the same results. As usual the second order RKDG-method gives the best solution, but a slope limiter is needed to avoid overshoots.

Increasing the time step is not possible for the upwind CCSM-method. Doing this for the central CCSM-method makes the method much more diffusive, although the shape of the solution remains still right when the time step is increased by a factor four.

7.6 Stationary solution with bottom elevation

In this and the next test case we take into account the source term as the result of a non-zero bottom. If the system starts in rest, with a constant water level and no flow, and there is no disturbance coming from the outside world the system should remain in rest, even if there are changes in the bottom level. In Figure 7.26 the solution is shown at $T_{end} = 1\,s$ for the first order RKDG-method. The bold line represents the bottom and it can be seen that in the numerical solution the total water depth in each element is constant, resulting in a non-constant water level. This causes the water to flow and results in a non-zero velocity. Figure 7.27 shows the solution of the second order RKDG-method at $T_{end} = 10\,s$. It can be seen that the water level remains constant and thus the velocity remains zero, as it should be. The solution of the CCSM-method is the same of that of the second order RKDG method: nothing happens. This means that the first order RKDG-method can not be used when there are changes in the bottom level.

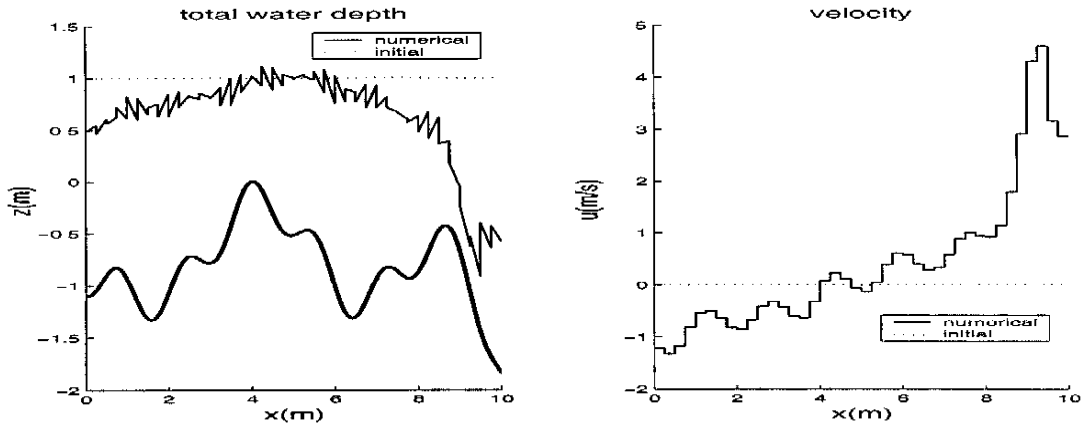


Fig. 7.26: Solution of the first order RKDG-method for $N-1=40$ at $T_{end}=1\,s$.

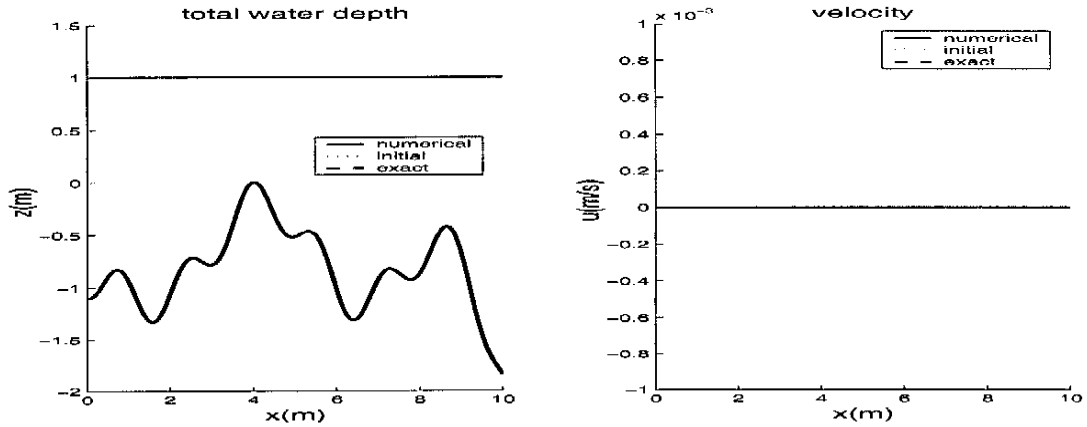


Fig. 7.27: Solution of the second order RKDG-method and CCSM-method for $N-1=40$ at $T_{end}=10\,s$.

7.7 Flow over an isolated ridge

In this test case we look at flow over an isolated ridge as is examined in Houghton & Kasahara (1968). We consider a smooth convex obstacle which is symmetric with respect to its crest and the bottom is defined by the following relation

$$h=\begin{cases}\delta(x-x_p)^2 & \text{if} \quad -\sqrt{h_c/\delta}<(x-x_p)<\sqrt{h_c/\delta},\\ h_c & \text{otherwise,}\end{cases} \tag{7.27}$$

with x_p the location of the crest of the obstacle and h_c the height of the crest and the reference level $\zeta=0$ is at the top of the ridge. To represent the height of the obstacle, we use the parameter M_c indicating the ratio of the height of the crest, h_c, over the depth of the approaching fluid, H_0,

$$M_c=\frac{h_c}{H_0}, \tag{7.28}$$

so if $M_c\geq 1$ the flow is totally blocked.

The initial condition consists of a constant water level and a constant velocity. There are four possible solution domains of the flow, depending on the initial Froude number ($Fr_0=u_0/\sqrt{gH_0}$)

and the value of M_c, see Houghton & Kasahara (1968). For increasing values of Fr_0 by constant M_c subsequently the following solution domains will be gone through:

I The flow is everywhere subcritical and the free surface of the steady state dips symmetrically over the symmetrical obstacle, see Figure 7.28.

II The flow is discontinuous and is critical at the crest of the obstacle. At the lee side the flow is supercritical and a stationary lee jump on the downstream side of the obstacle crest occurs together with a rarefraction wave. At the upstream side a bore propagates away from the obstacle, see Figure 7.29.

III The flow is almost the same as in Domain II only the lee jump moves away from the obstacle, see Figure 7.30.

IV The flow is everywhere supercritical and the free surface of the steady state rises symmetrically over the obstacle, see Figure 7.31.

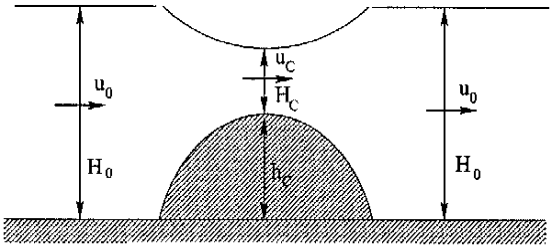


Fig. 7.28: Solution in Domain I.

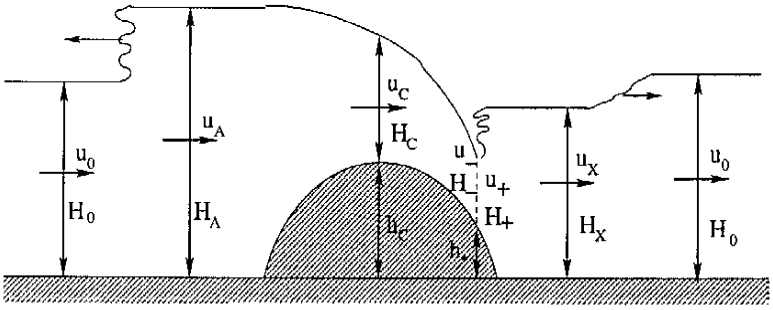


Fig. 7.29: Solution in Domain II.

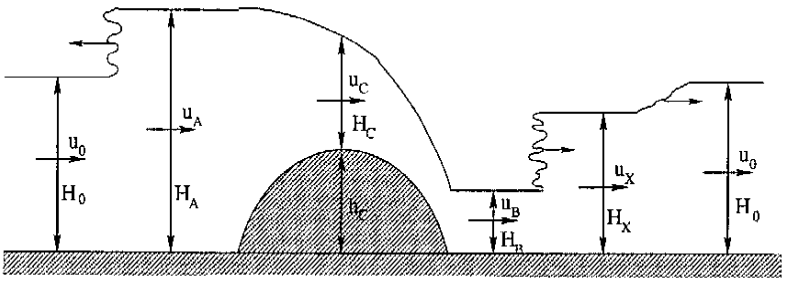


Fig. 7.30: Solution in Domain III.

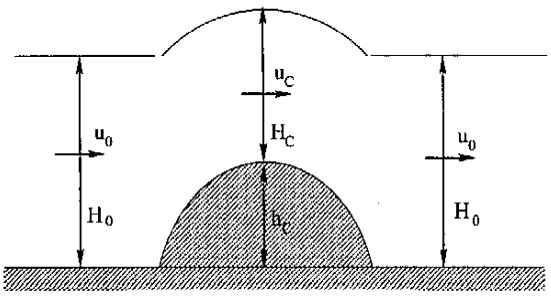


Fig. 7.31: Solution in Domain IV.

The values of the parameters, depicted in the figures, can be calculated analytically as can be found in Houghton & Kasahara (1968). We will compare these analytical values with the values calculated by the numerical models and also look at the graphical representation of the numerical solutions.

Flow over a flat plate When $h_c = 0$ in (7.27) we have flow over a flat plate. In this case the CCSM-method with the central approach in the continuity equation produces uncorrect solutions when $Fr > 1.1$, but there is no restriction on the Froude number when the upwind approach in the continuity equation is used. So when the Froude number is expected to exceed 1 it is better to use the upwind approach in the continuity equation. And for this reason we use the upwind approach in this test case throughout.

The following values for the parameters are used in all the domains: $N - 1 = 100$, $x_0 = 0\,m$, $x_N = 100\,m$, $H_0 = 0.2\,m$, $h_c = 0.1\,m$ (this gives $M_c = 0.5$), $\delta = 0.05\,m^{-1}$, $x_p = 40\,m$, $T_{end} = 20\,s$ and in the different domains we use:

I $u_0 = 0.28\,m/s$ $(Fr_0 = 0.2)$,

II $u_0 = 0.42\,m/s$ $(Fr_0 = 0.3)$,

III $u_0 = 0.98\,m/s$ $(Fr_0 = 0.7)$,

IV $u_0 = 2.66\,m/s$ $(Fr_0 = 1.9)$.

At the boundaries transmissive boundary conditions are used, because they are far enough away from the domain of interest to have no influence on the solution.

The Figures 7.32 till 7.35 show the solutions for the four cases for the both methods. The bold line represents the solution of the CCSM-method and the thin line the solution of the second order RKDG-method. A slope limiter with TVB correction constant $M = 1 \cdot 10^{-2}$ is used in Domain II and III for the second order RKDG-method and no slope limiter is applied in Domain I and IV. The CCSM-method uses the upwind approach in the continuity equation, one iteration step and $\theta_i = 0.51$ for $i = \{d, c, g\}$. The same time step with $\sigma = 0.3$ is used for both methods in all domains. In the figures we zoomed in on the area of interest.

The slope limiter, as it is given in Section 5.2.3, does not account for changes in the bottom level. This can be seen in Figures 7.33 and 7.34 where the slope limiter is used, because there are still overshoots and undershoots when there are large changes in the bottom. This is explained by the fact that the slope limiter limits the slope in the total water depth, not in the water level.

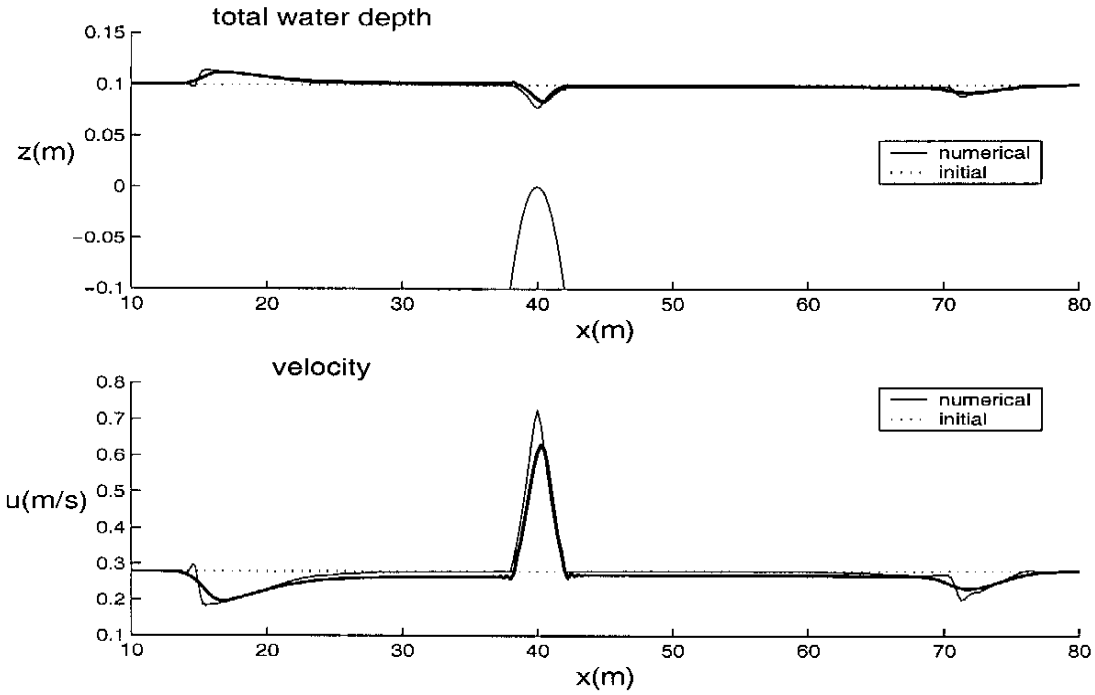


Fig. 7.32: Solution of the second order RKDG-method with slope limiter (thin line) and the CCSM-method with upwind approach (bold line) for case I with $N-1=400$.

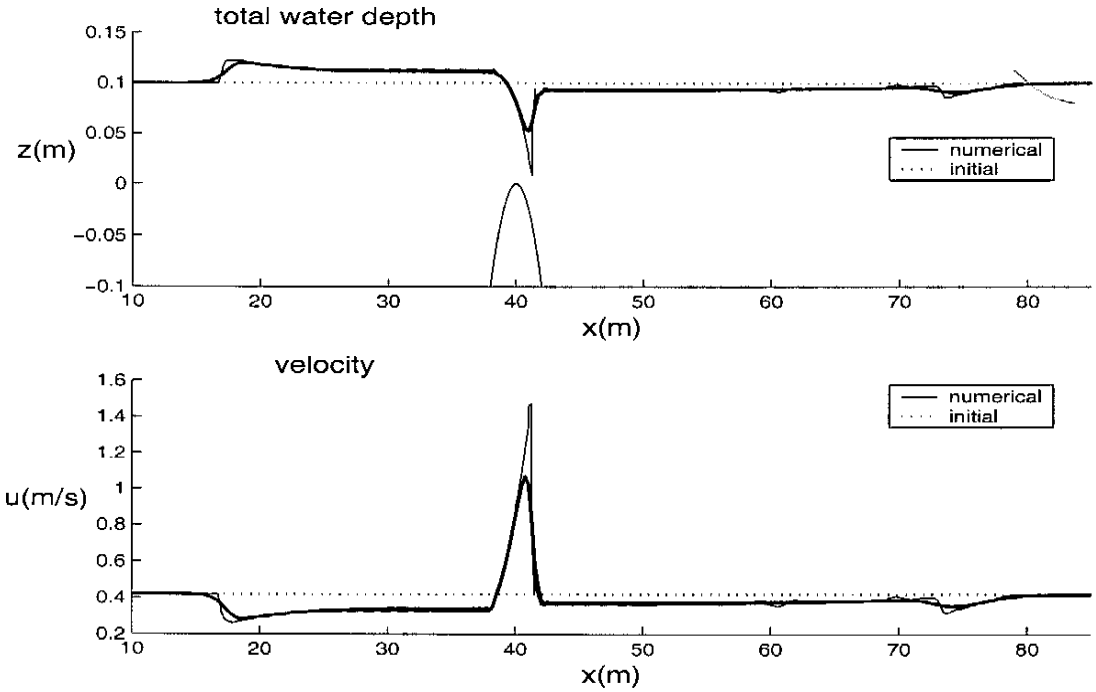


Fig. 7.33: Solution of the second order RKDG-method with slope limiter (thin line) and the CCSM-method with upwind approach (bold line) for case II with $N-1=400$.

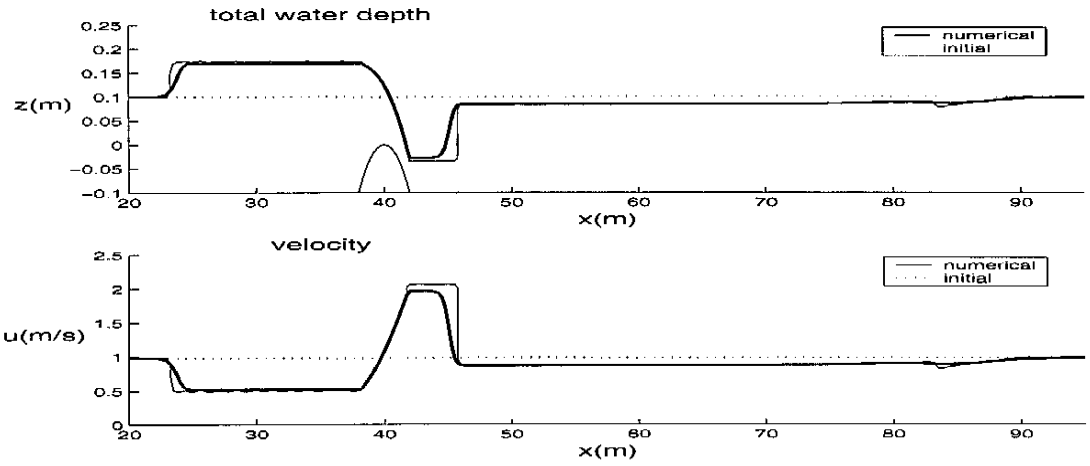


Fig. 7.34: Solution of the second order RKDG-method with slope limiter (thin line) and the CCSM-method with upwind approach (bold line) for case III with $N-1=400$.

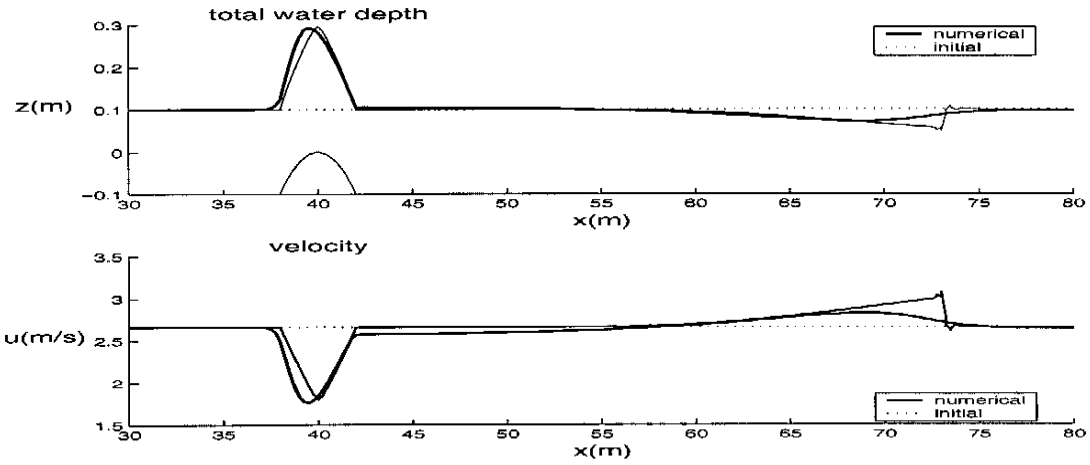


Fig. 7.35: Solution of the second order RKDG-method with slope limiter (thin line) and the CCSM-method with upwind approach (bold line) for case IV with $N-1=400$.

By comparing the solutions in the Domains II and III it can be seen that the CCSM-method is more diffusive than the RKDG-method. Especially the lee jump in Figure 7.33 for the CCSM-method than for the RKDG-method. In Figure 7.35 the solution of the CCSM-method is not symmetrical as it should be. This is caused by the fact that the upwind approach is used in the CCSM-method which gives a diffusive effect. Refining the grid decreases the diffusion in the CCSM-method.

In Tables 7.7 till 7.10 the analytical as well as the numerical values of the different parameters in the subsequent domains are given.

Table 7.7: Comparison of analytical and numerical values in Domain I.

Parameter	Analytical	RKDG-method	CCSM-method
u_c	$7.26 \cdot 10^{-1}$	$7.26 \cdot 10^{-1}$	$6.09 \cdot 10^{-1}$
H_c	$7.71 \cdot 10^{-2}$	$7.72 \cdot 10^{-2}$	$8.67 \cdot 10^{-2}$

Table 7.8: Comparison of analytical and numerical values in Domain II.

Parameter	Analytical	RKDG-method	CCSM-method
u_A	$3.27 \cdot 10^{-1}$	$3.27 \cdot 10^{-1}$	$3.40 \cdot 10^{-1}$
H_A	$2.13 \cdot 10^{-1}$	$2.13 \cdot 10^{-2}$	$2.12 \cdot 10^{-2}$
u_c	$8.82 \cdot 10^{-1}$	$8.61 \cdot 10^{-1}$	$7.60 \cdot 10^{-1}$
H_c	$7.93 \cdot 10^{-2}$	$8.10 \cdot 10^{-2}$	$9.01 \cdot 10^{-2}$
u_-	$1.49 \cdot 10^{0}$	$1.47 \cdot 10^{0}$	$1.07 \cdot 10^{0}$
H_-	$4.68 \cdot 10^{-2}$	$4.77 \cdot 10^{-2}$	$7.78 \cdot 10^{-2}$
u_+	$5.62 \cdot 10^{-1}$	$4.86 \cdot 10^{-1}$	$4.80 \cdot 10^{-1}$
H_+	$1.24 \cdot 10^{-1}$	$1.26 \cdot 10^{-1}$	$1.65 \cdot 10^{-1}$
$h_\star$	$5.83 \cdot 10^{-2}$	$6.09 \cdot 10^{-2}$	$7.50 \cdot 10^{-2}$
u_X	$3.64 \cdot 10^{-1}$	$3.65 \cdot 10^{-1}$	$3.74 \cdot 10^{-1}$
H_X	$1.92 \cdot 10^{-1}$	$1.92 \cdot 10^{-2}$	$1.94 \cdot 10^{-2}$

Table 7.9: Comparison of analytical and numerical values in Domain III.

Parameter	Analytical	RKDG-method	CCSM-method
u_A	$5.01 \cdot 10^{-1}$	$5.01 \cdot 10^{-1}$	$5.26 \cdot 10^{-1}$
H_A	$2.73 \cdot 10^{-1}$	$2.74 \cdot 10^{-2}$	$2.70 \cdot 10^{-2}$
u_c	$1.10 \cdot 10^{0}$	$1.08 \cdot 10^{-1}$	$9.93 \cdot 10^{-2}$
H_c	$1.24 \cdot 10^{-1}$	$1.26 \cdot 10^{-2}$	$1.36 \cdot 10^{-2}$
u_B	$2.08 \cdot 10^{0}$	$2.07 \cdot 10^{-1}$	$1.91 \cdot 10^{-1}$
H_B	$6.59 \cdot 10^{-2}$	$6.61 \cdot 10^{-2}$	$7.31 \cdot 10^{-2}$
u_X	$8.77 \cdot 10^{-1}$	$8.77 \cdot 10^{-1}$	$8.71 \cdot 10^{-1}$
H_X	$1.87 \cdot 10^{-1}$	$1.86 \cdot 10^{-2}$	$1.85 \cdot 10^{-2}$

Table 7.10: Comparison of analytical and numerical values in Domain IV.

Parameter	Analytical	RKDG-method	CCSM-method
u_c	$1.80 \cdot 10^{0}$	$1.80 \cdot 10^{0}$	$1.85 \cdot 10^{0}$
H_c	$2.96 \cdot 10^{-1}$	$2.96 \cdot 10^{-1}$	$2.81 \cdot 10^{-1}$

In Domain I, the velocity at the crest is too small for the CCSM-method. This can also be seen in Figure 7.32 where the velocity seems shifted downwards. The water depth at the crest in contrary is too big, so mass is still conserved. In Domain IV it is just the other way around. The velocity at the crest is too high and the water depth too small, but still mass is conserved. The velocity and water depth at the crest for the second order RKDG-method in Domain I and IV match very well with the analytical values.

The CCSM-method produces in Domain II the best results for the values of u_A and H_A before the ridge and the values of u_X and H_X after the ridge. For the values in between, at the crest and at the lee jump, the velocity is structurally too low and the water depth too big. The place of the lee jump is also too far to the right. In Domain III the errors are smaller, and again the values at the locations A and X are the best and the velocity is again structurally too low and the water depth too big.

For the second order RKDG-method the best results in Domain II are also obtained at the locations A and X. Just as for the CCSM-method the velocity is lower and the water depth higher in the regions in between, but less severe than for the CCSM-method. The lee jump is also slightly shifted to the right. In Domain III the RKDG-method has less problems and the

numerical values are very close to the analytical ones.

Conclusions For the CCSM-method only the upwind approach can be used, because the central method cannot deal with supercritical flow. For the RKDG-method the second order method has to be used, because the first order method cannot deal with changes in the bottom level. A slope limiter has to be applied in Domains II and III, but, as can be seen in the figures, the slope limiter has to be adjusted to deal with changes in the bottom level.

The CCSM-method is diffusive caused by the use of the upwind scheme. The second order RKDG-method only has some minor difficulties with the lee jump in Domain II.

Chapter 8

Conclusions and Recommendations

In Chapter 2 the derivation of the shallow water equations (SWE) from the Navier-Stokes equations is given. The final result of this derivation are the one-dimensional shallow water equations which are the equations of interest in this thesis. The assumptions, to derive the 1D shallow water equations from the Navier-Stokes equations, are:

- the horizontal length and velocity scales of the flow are much larger than the vertical ones,
- the length scales are much larger than the scales related to the width variations of the flow,
- the width of the flow is constant.

The primary variables in the 1D SWE are the average discharge and the total water depth. Only changes in the bottom level are taken into account, while for example viscosity, Coriolis forces and turbulence are neglected in this thesis.

Unstructured grids are introduced in Chapter 3 and their greater geometric flexibility compared to structured grids is the reason why the interest in this work is on numerical methods that are applicable to unstructured grids. Because of the ease of implementation, obtaining higher order accuracy of the advection terms and the knowledge present in literature, a collocated grid is preferred over a staggered grid. The choice for a vertex-centred approach is made, because in this way a unique linear interpolation can be defined within a grid cell and the evaluation of the gradients at control volume edges is easier than when a cell-centred approach is used.

Four different numerical methods that can be used to solve the shallow water equations are outlined in Chapter 4. The finite difference method and spectral method are quite difficult to apply on unstructured grids in contrast with the finite volume method and the finite element method for which this is much easier. For this reason a finite volume and a finite element method are further examined and compared. The two methods are outlined in detail in Chapter 5.

The first method, the so-called Collocated Coupled Solution Method (CCSM), is a collocated, vertex-centred finite volume method in which a linear approximation of the primary variables is used. Due to the use of a first order upwind scheme in the momentum equation the method is first order accurate in space. Discretization of the advection term in the continuity equation is either done by means of a central or a first order upwind approach. The system of ordinary differential equations is written in matrix form. The time marching procedure is based on solving the coupled system of equations at once by means of a θ-method. It is quite simple to make changes in the discretization, because it only implies a change in the computation of

certain matrix elements and not of the solution procedure. When θ equals a half, the linearized system is second order in time. Picard linearization or an iterative process is used to deal with the nonlinear terms and for this reason the CCSM-method is semi-implicit, which allows for relative large time steps. The choice for the time step is only limited by accuracy reasons.

The Runge-Kutta Discontinuous Galerkin finite element method is the second method examined in this thesis. It uses a discontinuous piecewise polynomial approximation for the variables, and the method is very local. More degrees of freedom are involved relative to finite volume methods. The HLLC approximate Riemann solver is used to locally solve the Riemann problem between two cells, as is common in most finite volume methods. A diagonal mass matrix is obtained by the use of orthogonal Legendre polynomials and this avoids the necessity to solve a local matrix system. For the solution procedure an explicit TVB Runge-Kutta scheme is used and this gives a CFL-restriction on the time step, which becomes more severe when the order of the method increases. A slope limiter can be applied to avoid unphysical oscillations in higher order schemes.

In Chapter 7 some numerical test cases are performed to test the numerical methods and to compare them with each other. A linear wave is examined for which the first order and second order RKDG-methods indeed converge with respectively order one and two. The CCSM-method converges also with order two, even though it is a first order method. This is caused by the linearity of the solution, because the CCSM-method is only a first order method due to the first order upwind scheme used for the nonlinear advection term. When the solution contains a reasonable amount of nonlinearity, such as in the test case with one of the Riemann variables being constant, the convergence of the CCSM-method is, as expected, of order one. Because the RKDG-method is explicit, the method becomes unstable when the CFL-number exceeds one. The time step of the CCSM-method with a central approach in the continuity equation can be chosen up to a factor 10 larger and still produce quite accurate results for the linear wave problem. For the upwind approach this is not possible, probably due to the simple design of the upwind scheme.

For accurate solutions of the problem with one of the Riemann variables being constant, the second order RKDG-method needs a slope limiter and in the CCSM-method the upwind approach in the continuity equation must be applied to avoid overshoots in the neighbourhood of large gradients. As expected, the second order RKDG-method with slope limiter gives the best results. The first order RKDG-method and the CCSM-method with upwind approach in the continuity equation give both similar results, while being both more diffusive than the second order RKDG-method.

For the dam break problem the second order RKDG-method gives again the best solution and needs a slope limiter to avoid over- and undershoots. The CCSM-method with upwind approach in the continuity equation and one iteration step gives practically the same result as the first order RKDG-method. Only some small wiggles, which are probably caused by the initial discontinuity, are generated in the CCSM-method.

The solution to the other Riemann problem consists of two rarefractions and a nearly dry bed. A slope limiter has to be used for the second order RKDG-method to avoid negative water depths and this method gives the best results. The first order RKDG-method does not stay behind much and the upwind CCSM-method with one iteration step also works well, but is a bit more diffusive. A central approach cannot be used in the CCSM-method, because then the water depth becomes negative.

To mimic tidal movement a time dependent boundary condition is imposed. The implementation of the boundary condition for the CCSM-method is modified in some ad hoc fashion to avoid odd-even decoupling. The first order RKDG-method and the upwind CCSM-method us-

ing one iteration step give practically the same results. The second order RKDG-method gives the best solution, but a slope limiter is needed to avoid overshoots. Increasing the time step is not possible for the CCSM-method with upwind approach as was also the case for the linear wave. Increasing the time step for the central CCSM-method makes the method very diffusive, although the shape of the solution even remains correct when the time step is increased by a factor four.

When changes in the bottom level are present, the first order RKDG-method, as used in this work, can not be applied, because water starts to flow even when it should remain stationary. The second order RKDG-method and the CCSM-method do not have this problem.

When examining flow over an isolated ridge, the CCSM-method can only be applied with the upwind approach, because the central method cannot deal with supercritical flow. For the RKDG-method the second order method has to be used, since the first order method inherently cannot cope with changes in the bottom level. A slope limiter has to be applied when the flow contains discontinuities, but the slope limiter has to be adjusted to deal with changes in the bottom level. The CCSM-method is diffusive which is probably caused by the diffusive effect of the upwind scheme. The second order RKDG-method only has some minor difficulties with the lee jump in one of the solution domains.

It can be concluded that both the RKDG-method and the CCSM-method are capable of computing 1D shallow water flows. For flow involving low Froude numbers the CCSM-method introduced in this report can use larger time-steps than the RKDG-method used in this M.Sc. thesis, but the RKDG-method is better suited to deal with discontinuities in the flow.

Recommendations for further investigations

- The implementation of Newton-Raphson iteration in the CCSM-method, instead of the Picard linearization or the simple iteration process, will give a better treatment of the nonlinearity in the method.

- Investigation of the reason why no large time steps can be used when the CCSM-method with upwind approach in the continuity equation is used. The cause of the oscillations that occur when using the CCSM-method should also be investigated.

- The implementation of the scalar Engquist-Osher flux in the CCSM-method, instead of using the simple upwind scheme, may avoid the problems that are encountered when the velocity changes sign.

- Using a second order discretization for the advection term in the momentum and mass equation, will make of the CCSM-method a second order scheme. In this way a better comparison between the two methods can be made.

- The slope limiter in the RKDG-method has to be modified to be able to deal with changes in the bottom level, see Schwanenberg (2003).

- In this thesis only a comparison between the accuracy of the both methods is made, but the efficiency of the methods is not considered yet.

- The RKDG-method and CCSM-method are only able to deal with positive water depths. To be able to deal with moving boundaries due to flooding and drying, a flooding and drying algorithm has to be implemented. Investigation and testing of such algorithms is thus a important field of research.

- Problems on non-uniform grids should be investigated, because they show some of the problems that one encounters when going to unstructured grids in 2D.

- Other processes involved in shallow water flow should be implemented, such as viscosity, bottom roughness and turbulence to get a more realistic approximation.

Bibliography

User Manual Delft3D-Flow, WL | Delft Hydraulics, 2001.

Anderson J. J.D., (ed.) *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill, Inc., 1995.

Balzano A., Evaluation of Methods for Numerical Simulation of Wetting and Drying in Shallow Water Flow Models, *Coastal Engineering*, vol. 34, no. 1, pp. 83–107, 1998.

Barth T.J. & Jespersen D.C., The Design and Application of Upwind Schemes on Unstructured Meshes, Tech. Rep. AIAA-89-0366, NASA Ames research centre, California, 1989.

Bates P.D. & Anderson M.G., A Two-Dimensional Finite-Element Model for River Flow Inundation, in *Proceedings: Mathematical and Physical Sciences*, vol. 414, 1993.

Bates P.D. & Hervourt J.M., A New Method for Moving Boundary Hydrodynamic Problems in Shallow Water, in *Proceedings of the Society of London A. Mathematical Physical and Engineering Sciences*, vol. 455, pp. 3107–3128, 1999.

Bokhove O., Flooding and Drying in Finite-Element Discretizations of Shallow-Water Equations. Part 1: One Dimension., 2003, submitted to *Journal of Scientific Computing*.

Bonekamp H. & Borsboom M., Drying and Flooding in TRITON, Tech. Rep. H3976.40, WL | Delft Hydraulics, 2002.

Bradford S.F. & Sanders B.F., Finite Volume Model of Shallow Water Flooding of Arbitrary Topography, *Journal of Hydraulic Engineering*, vol. 128, no. 3, pp. 289–298, 2002.

Brenner S.C. & Scott L.R., *The Mathematical Theory of Finite Element Methods*, no. 15 in Texts in Applied Mathematics, Springer-Verlag New York, 1994.

Brufau P., Vázquez-Cendón M.E. & García-Nevarro P., A Numerical Model for the Flooding an Drying of Irregular Domains, *International Journal for Numerical Methods in Fluids*, vol. 39, no. 3, p. 247, 2002.

Cockburn B., Discontinuous Galerkin Methods for Convection Dominated Problems, 1998.

Cockburn B., Lin S.Y. & Shu C.W., TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws. III. One-Dimensional Systems, *Journal of Computational Physics*, vol. 84, no. 1, pp. 90–113, 1989.

Cockburn B. & Shu C.W., TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws. II. General Framework, *Mathematics of Computation*, vol. 52, no. 186, pp. 411–435, 1989.

Defina A., Two-Dimensional Shallow Flow Equations for Partially Dry Areas, *Water Resources Research*, vol. 36, no. 11, pp. 3251–3264, 2000.

Engquist B. & Osher S., One-sided Difference Approximations for Nonlinear Conservation Laws, *Mathematics of Computation*, vol. 36, pp. 321–351, 1981.

Ertürk S.N., Bilgili A., Swift M.R. et al., Simulation of the Great Bay Estuarine System : Tides with Tidal Flats Wetting and Drying, *Journal of Geophysical Research*, vol. 107, no. C5, pp. 6.1–6.11, 2002.

Faber T.E., *Fluid Dynamics for Physicists*, Cambridge University Press, 1995.

de Goede E.D., Delft3D/TRINSULA, Conceptual Model and Algorithmic Implementation, Tech. rep., WL | Delft Hydraulics, 1995.

Heniche M.e.a., A Two-Dimensional Finite Element Drying-Wetting Shallow Water Model for Rivers and Estuaries, *Advances in Water Resources*, vol. 23, no. 4, pp. 359–372, 2000.

Horritt M.S., Evaluating Wetting and Drying Algorithms for Finite Element Models of Shallow Water Flow, *International Journal for Numerical Methods in Engineering*, vol. 55, no. 7, pp. 835–851, 2002.

Houghton D.D. & Kasahara A., Nonlinear Shallow Fluid Flow Over an Isolated Ridge, *Communications on Pure and Applied Mathematics*, vol. 21, pp. 1–23, 1968.

Hu K., Minghan C.G. & Causon D.M., Numerical Solution of Wave Overtopping of Coastal Structures, *Coastal Engeneering*, vol. 41, no. 4, pp. 433–465, 2000.

Hubbard M.E. & Dodd N., A 2D Numerical Model of Wave Run-Up and Overtopping, *Coastal Engineering*, vol. 47, no. 1, pp. 1–26, 2002.

Johnson R.W., (ed.), *The Handbook of Fluid Dynamics*, chap. 27, 28, 29 and 45, CRC Press, 1998, ISBN 0-8493-2509-9 3-540-64612.

Kulikovskii A.G., Pogorelov N.V. & Semenov A.Y., *Mathematical Aspects of Numerical Solution of Hyperbolic Systems*, no. 118 in Monographs and Surveys in Pure and Applied Mathematics, Chapman & Hall/CRC, 2001.

Li R., Chen Z. & Wu W., *Generalized Difference Methods for Differential Equation: Numerical Analysis of Finite Volume Methods*, no. 226 in Monographs and Textbooks in Pure and Applied Mathematics, Marcel Dekker, 2000.

Lucquin B. & Pironneau O., *Introduction to Scientific Computing*, 1998.

Lynch D.R. & Gray W.G., Finite Element Simulation of Shallow Water Problems Moving Boundaries, in *Proceedings of the Second International Conference on Finite Elements in Water Resources*, 1978.

Morton K.W. & Mayers D.F., *Numerical Solution of Partial Differential Equations*, Cambridge University Press, 1994.

Quecedo M. & Pastor M., A reappraisal of Taylor-Galerkin algorithm for drying-wetting areas in shallow water computations, *International Journal for Numerical Methods in Fluids*, vol. 38, pp. 515–532, 2002.

Schwanenberg D., *Die Runge-Kutta-Discontinuous-Galerkin-Methode zur Lösung konvektionsdominierter tiefengemittelter Flachwasserprobleme*, Ph.D. thesis, Rheinisch-Westfälischen Technischen Hochschule Aachen, 2003.

Schwanenberg D. & Harms M., Discontinuous Galerkin Method for Dam-Break Flows, in D. Bousmar & Y. Zech, (eds.) *River Flow 2002 - Proceedings of the International Conference on Fluvial Hydraulics*, vol. 1, pp. 443–448, 2003.

Shu C.W. & Osher S.J., Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes, *Journal of Computational Physics*, vol. 77, pp. 439–471, 1988.

Shyy W., Udaykumar H.S., Rao M.M. et al., *Computational Fluid Dynamics with Moving Boundaries*, Taylor & Francis, 1996.

Sleigh P.A., Gaskell P.H., Berzins M. et al., An Unstructured Finite-Volume Algorithm for Predicting Flow in Rivers and Estuaries, *Computers & Fluids*, vol. 27, no. 4, pp. 479–508, 1998.

Stelling G., *On the construction of computational methods for shallow water flow problems*, Ph.D. thesis, Delft University of Technology, 1983.

Tchamen G.W. & Kahawita R.A., Modelling Wetting and Drying Effects over Complex Topography, *Hydrological Processes an International Journal*, vol. 12, no. 8, pp. 1151–1182, 1998.

Toro E.F., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer-Verlag, 1997.

Toro E.F., *Shock-Capturing Methods for Free-Surface Shallow Flows*, John Wiley & Sons, 2001.

van der Vegt J.J.W. & Bokhove O., Finite Element Methods for Partial Differential Equations, 2003, lecture notes, University of Twente.

Vreugdenhil C.B., *Numerical Methods for Shallow-Water Flow*, no. 13 in Water Science and Technology Library, Kluwer Academic Publishers, 1994.

Wendt J.F., (ed.) *Computational Fluid Dynamics: An Introduction*, Springer, second edn., 1996.

Wenneker I., *Computation of Flows Using Unstructered Staggered Grids*, Ph.D. thesis, Technische Universiteit Delft, 2002.

Wenneker I., Solution of the Shallow Water Equations using Unstructured Grids, Tech. Rep. X0267.10, WL | Delft Hydraulics, 2003.

Young D.F., Munson B.R. & Okiishi T.H., *A Brief Introduction to Fluid Mechanics*, John Wiley & Sons Inc, 1997.