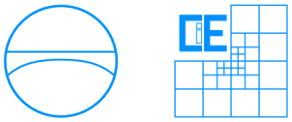


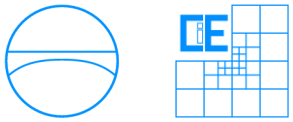
Faculty of Civil, Geo and Environmental Engineering
Chair for Computation in Engineering
Prof. Dr. rer. nat. Ernst Rank

Implementation of the Characteristic Based Split algorithm for the Shallow Water Equations in a high order finite element framework

Philipp Kopp

Bachelor's thesis
for the Bachelor of Science program Environmental Engineering

Author:	Philipp Kopp
Supervisor:	Prof. Dr. rer. nat. Ernst Rank
	Nils Zander, M.Sc.
Date of issue:	20 June 2014
Date of submission:	13 October 2014



Involved Organisations

Chair for Computation in Engineering
Faculty of Civil, Geo and Environmental Engineering
Technische Universität München
Arcisstraße 21
D-80333 München

Declaration

With this statement I declare, that I have independently completed this Bachelor's thesis. The thoughts taken directly or indirectly from external sources are properly marked as such. This thesis was not previously submitted to another academic institution and has also not yet been published.

München, October 13, 2014

Philipp Kopp

Philipp Kopp
e-Mail:philipp.kopp@tum.de

Contents

Chapter 1

Introduction

1.1 Shallow water type of flows as a common problem in environmental engineering

In environmental engineering problems of large scale, fluid flow with little vertical influences often arise. For example, in flood forecasting one might want to predict the water height at some important infrastructure points as a consequence of a dam break upstream. In another situation of long lasting super-regional rainfall, it might be to decide if it is necessary to evacuate a village that is situated beside a big river. In many cases, the influxes upstream are known by water level measurements (at water gauges) in connection with a water level discharge relation. To estimate the peak height at the village, the reaction of the river must be predicted. This can be modelled as a one dimensional shallow water flow. In another problem type, the peak water level at the village might be given and the subject of interest is a prediction of which areas are affected by the flood. As can be seen, the flow problems in environmental engineering are very versatile, and it would be desirable to have a generic method available for handling problems of the type described above.

1.2 General approaches to the solution of shallow water flows

There are different approaches to those kind of problems. A rough division can be made into black-box and white-box concepts. While black-box methods relate input and output variables by empirical laws, white box models try to completely describe the underlying physics of the problem.

A drawback of black-box models is the limited ability to be adapted to different situations. Additionally, the insight to the nature of the problems is in most cases very small. An example of such a method would be the water level prediction at some point of interest through the superposition of unit discharge hydrographs. A white-box approach would be the description of the fluid flow in terms of physical conservation laws resulting in a system of differential equations that has to be solved. For this kind of problem, the quantities that have to be preserved are mass and momentum. Requiring their conservation on an infinitesimal control volume results in the formulation of the famous Navier-Stokes equations for incompressible

fluids. As there are no restrictions on the shape of the flow domain, this could represent the generic tool mentioned in section 1.1.

However, in most real world cases it is not possible to derive an analytical solution for the Navier-Stokes equations, due to complicated geometries, initial conditions, boundary conditions or source terms. For this reason, numerical methods are applied, which can be very time consuming. Fortunately, it turns out, that under the assumption of a hydrostatic pressure distribution one space dimension of the Navier-Stokes equations can be eliminated without much loss of accuracy. Of course this is only exact if the fluid is at rest, but for a typical engineering accuracy in many cases it is sufficient that the horizontal scales are much larger than the vertical scales (see section 2.7.3). The resulting system of partial differential equations is called shallow water equations (or also Saint Venant equations, especially the 1-d version) and is described in detail in chapter 2. Although the problem of finding analytical solutions does not change by this modification, the reduction of one space dimension reduces the computational cost of numerical solutions greatly. In this work, the shallow water equations are solved with the finite element method, one of the most popular techniques for solving partial differential equations. In particular the characteristic-based split algorithm, presented in section 3.3, will be used to overcome problems arising from convective term in the shallow water equations. To start with an overview of existing numerical procedures suitable for this kind of problems, some popular methods are described in the following section.

1.3 Overview of existing numerical solution methods

There are many techniques for achieving an approximation to the analytical solution of the governing equations of fluid dynamics. Besides the related finite difference method, finite volume method and finite element method, alternative procedures like Lattice-Boltzmann methods were also able to give excellent results. But as they follow a different approach and this work is only concerned with the finite element method, the reader is referred to the extensive literature on this topic.

1.3.1 Finite difference method

In the finite difference method, the continuous *differential operators* of the partial differential equation are replaced by discrete *difference operators*. This requires a previous discretization of the geometry, such that the difference operators can be written in terms of the grid points. However, it is difficult to find difference operators for unstructured grids, so the method is in most cases restricted to structured grids. For a simple approximation of the first order derivative forward difference operators with first order accuracy

$$\frac{\partial \phi}{\partial x} \approx \frac{\phi_{i+1} - \phi_i}{\Delta x} \tag{1.1}$$

or central difference operators with second order accuracy

$$\frac{\partial \phi}{\partial x} \approx \frac{\phi_{i+1} - \phi_{i-1}}{2\,\Delta x} \tag{1.2}$$

can be used, where ϕ_i denotes the unknown quantity at node i. As an example, the one dimensional transport equation of the form

$$\frac{\partial \phi}{\partial t} + U \frac{\partial \phi}{\partial x} = 0 \tag{1.3}$$

would become by substituting forward differences in time and central differences in space the following difference equation:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} + U \frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\,\Delta x} = 0. \tag{1.4}$$

Note, that this is an *explicit* time-integration, because the convective term $U(\partial\phi/\partial x)$ is evaluated at $t = n$. If the evaluation would be at $t = n + 1$, it would be called an *implicit* scheme. However, there are various other difference operators with higher order accuracy or for approximating different derivatives, that have to be chosen independently for each problem.

The resulting system of equations requires the solution to fulfill the differential equation on a nodal basis. The solution process can be done by explicit methods like *Gaussian elimination* or by iterative procedures like the *conjugate gradient method*. After the solution process several post-processing techniques can be applied. The most important step missing is probably the interpolation of the solution field in-between the nodes.

Although the finite difference method is not subject of this work, it is important to note that transient finite element simulations often use a finite difference discretization in time. Additionally, the first good results in solving the equations of fluid dynamics numerically were achieved in a finite difference context by choosing *upwind difference operators*, as discussed in chapter 3.2.

1.3.2 Finite volume method

The finite volume method is a procedure for numerical solution of fluid mechanics equations which is currently the most commonly used method in commercial codes. The basic strategy is to discretize the domain by dividing it into finite volumes and requiring the partial differential equation to be satisfied in an integral over such a volume. In the next step, an integration by parts is performed on the volume integrals such that they are converted into surface integrals over the volumes faces. Now, those integrals are written in terms of the nodal degree of freedoms and a global system of equations is formulated. The time integration can similar to the finite element method be done by a finite difference time discretization.

One of the main problems of the finite volume method is that its accuracy is of order one and an extension to high orders is not straightforward. As a result, good accuracy requires high mesh refinement.

1.3.3 Finite element method

One of the most widely used procedures for solving partial differential equations especially in structural mechanics and electrical engineering is the finite element method. There exist

a variety of different procedures belonging to this family. However they share important similarities, such as:

- Reformulation of the partial differential equation in a weak form
- Discretization of the continuous weak form by choosing two spaces spanned by a finite amount of basis functions
- Assembly of the global stiffness matrix by summing up the individual element contributions

The general procedure starts with formulating the weak form of the problem as

$$\text{Find } u \in X \text{ such that } B(u,v) = f(v) \qquad \forall v \in Y, \tag{1.5}$$

where $B(u,v)$ is a bilinear form corresponding to the homogeneous part of the differential equation and $f(v)$ a linear mapping corresponding to the source term. At this point an integration by parts is often applied to transfer derivatives from u to v. As a result the continuity demand on u is reduced (or 'weakened'). The uniqueness of the solution of the weak form is given by the *Lax-Milgram Theorem* under the condition that X and Y are Hilbert spaces [Babuška, 1971].

In the next step, X and Y are substituted by the finite dimensional subspaces X^h and Y^h, such that the discretized solution $\hat{u}$ is represented by a set of basis functions $\mathbf{N}$ and their corresponding coefficients $\tilde{\mathbf{u}}$. Those coefficients are the unknowns which the finite element method attempts to find. This is done by extracting the coefficients from the discretized bilinear map $B(\hat{u},\hat{v})$ with the consequence that $B(\hat{u},\hat{v})$ does not anymore contain unknown quantities and therefore can be computed. Now, the resulting linear system can be solved for $\tilde{u}$.

An important family of finite element methods are weighted residual methods. In this case the weak form is achieved simply by requiring the L_2 inner product defined as

$$(a,b) = \int_\Omega a \cdot b \, d\Omega \tag{1.6}$$

of all $v \in Y$ with the differential equation dependent on $u \in X$ to vanish. A special weighted residual method called (Bubnov-)Galerkin method discretizes X and Y with the *same function space* which results in an *optimal approximation for self-adjoint problems* [Zienkiewicz *et al.*, 2005]. More details on this property can be found in section 3.1.1.

The choice of the function space can be very different, for instance low order finite elements are discretized with a piecewise linear ansatz. However, high order shape functions have a few important advantages:

- Exponential convergence in the energy norm for smooth solutions (and for non-smooth solutions at least equal convergence rate compared to low order finite elements)
- Better performance in case of large deformations
- Robustness with respect to locking effects
- Required for achieving good accuracy with blended elements

There are many possibilities for creating a high-order finite element basis, commonly used are Lagrange polynomials, Legendre polynomials, integrated Legendre polynomials or NURBS (Non-Uniform Rational B-Splines). Because of their *orthogonality* properties and good condi-

tion number for finite element matrices, only integrated Legendre polynomials are considered here. One main goal of this work is to explore the performance of the characteristic-based-spit algorithm introduced in chapter 3 in combination with high-order shape functions. A more detailed discussion about the integrated Legendre basis is given in section 4.1.2.

As indicated before, transient analysis with the finite element method is in most cases done in combination with a finite difference time discretization. Although it might in some cases be reasonable to use a space-time FEM formulation, generally the finite difference version is preferred.

1.4 Advantages of the finite element method

There are a few key benefits that explain the huge success of the finite element method over the last decades. An important point is the solid mathematical basis that allows systematic research regarding, for example, error estimation. The second and probably most important feature is the flexibility in handling different and complex geometries easily. Due to the definition of basis functions in a parameter space together with a local global mapping it becomes very easy to apply a finite element procedure to basically any structured or unstructured mesh. As a consequence, the number of degrees of freedom needed to describe some complex geometry can be reduced greatly. In contrast to that, the finite difference method can only be used efficiently with structured grids which is a big drawback for many practical engineering applications. Additionally, for the finite element method introducing high order schemes is straightforward and much easier compared to, for instance, the finite volume method. Also the imposition of Neumann boundary conditions can be done naturally in the finite element method while especially in the finite difference method the boundary conditions have to be discretized with at least as high order of accuracy than the discretization on the domain.

To make use of those benefits it is strongly desirable to develop a finite element environment for computing fluid dynamic problems. However this is not as straight forward as for typical structural engineering formulations, as it will be shown in chapter 3.1.

Chapter 2

Derivation of the shallow water equations

There are two general ways to derive the shallow water equations. The first possibility is to integrate the Navier-Stokes equations in their incompressible form over the water height and take averaged values for the velocities. The second alternative is to consider conservation laws for mass and momentum directly on an infinitesimal control volume. Because both strategies are in some way physically meaningful they will be presented separately in this chapter.

2.1 Prerequisites

In this section, a short introduction on general notation, hydrostatic pressure and (surface- and bottom-)boundary conditions will be given, because those subjects will be frequently addressed in this chapter.

2.1.1 Notation

As the notation is important for the following sections, an overview of the quantities used in this work is presented first. Figure 2.1 shows a schematic view of a shallow water problem, where η is the water surface elevation and H the bottom level, both relative to a horizontal reference level. Note that in figure 2.1 H would be negative. Defining H this way is slightly different than often done in the context of the CBS algorithm (see, for example, in Zienkiewicz *et al.* [2005]). However, it is more straightforward to think of a geodetic height that increases with x_3 than the other way round. Consequently the total water height is the difference between surface and bottom: $h = \eta - H$. Another commonly used abbreviation is the definition of $\mathbf{U} = \mathbf{u} \cdot h$ as the mass flow vector (to be precisely, as U does not contain the density ρ, it is more of a volume flow).

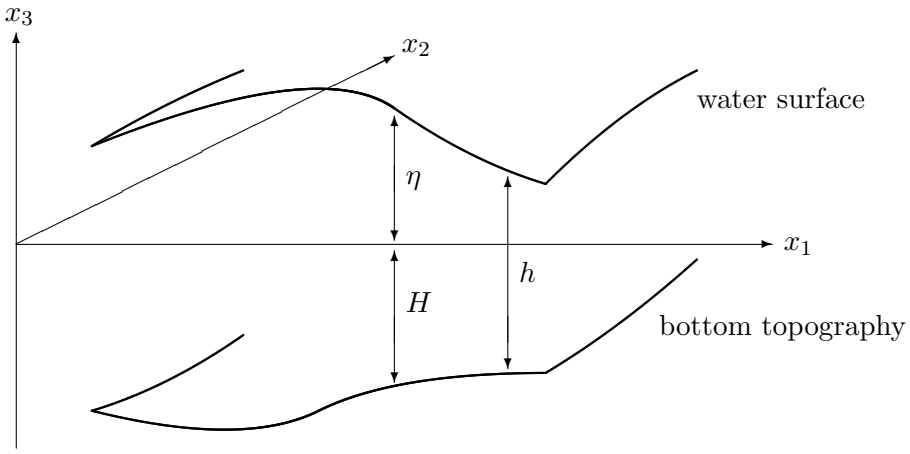


Figure 2.1: Notation

2.1.2 Hydrostatic pressure

The basic assumption in the shallow water theory is that the pressure distribution on a vertical profile is hydrostatic. This means, that the pressure increases linearly with higher fluid depth h. The constant factor is equal to the density ρ of the fluid times the gravitational constant g. So the pressure distribution can be written as

$$p(h) = \rho g h + p_0, \tag{2.1}$$

where p_0 is the atmospheric pressure acting on the water surface. However, this is only completely exact if the fluid is at rest.

2.1.3 Bottom and surface boundary conditions

To derive the shallow water equations, some assumptions for the bottom boundary and the water surface have to be introduced. Generally, it is assumed that the velocity normal to the bottom is zero. The dot product of the (downside pointing) normal vector defined as $\mathbf{n} = (\frac{\partial H}{\partial x_1}, \frac{\partial H}{\partial x_2}, -1)^T$ with the velocities $\mathbf{u} = (u_1^b, u_2^b, u_3^b)^T$ gives the following condition:

$$u_1^b \frac{\partial H}{\partial x_1} + u_2^b \frac{\partial H}{\partial x_2} - u_3^b = 0, \tag{2.2}$$

where the b superscript indicates evaluation at the bottom level. In combination with a no slip assumption ($u_1^b = u_2^b = 0$) the bottom boundary condition reduces to $u_3^b = 0$. However, in some cases of flow over infiltrating ground a velocity in normal direction may occur.

On the surface, the relative normal velocity needs to vanish to make sure that no particles leave the water continuum. As a result, the normal velocity must be equal to the time derivative of the surface elevation. The formulation arising from this condition is:

$$\frac{\partial \eta}{\partial t} + \mathbf{n} \cdot \mathbf{u} = 0.$$

Note that it is possible to interchange $\frac{\partial \eta}{\partial t}$ with $\frac{\partial h}{\partial t}$ as the bottom height H is normally assumed to be time invariant:

$$\frac{\partial h}{\partial t} = \frac{\partial (\eta - H)}{\partial t} = \frac{\partial \eta}{\partial t} - \cancelto{0}{\frac{\partial H}{\partial t}} = \frac{\partial \eta}{\partial t}.$$

Now, the surface condition can be written as:

$$\frac{\partial h}{\partial t} + u_1^s \frac{\partial \eta}{\partial x_1} + u_2^s \frac{\partial \eta}{\partial x_2} - u_3^s = 0, \tag{2.3}$$

with the s superscript indicating velocities on the water surface.

2.2 Integration of the Navier-Stokes equations over the height

The first and probably most popular way to derive the 2-d shallow water equations, as presented in Vreugdenhil [1994], is to integrate the 3-d Navier-Stokes equations over the height. As mentioned in chapter 1, this includes the assumption of a hydrostatic pressure, introduced in section 2.1.2, as well as the application of bottom and top boundary conditions described in section 2.1.3. To be able to start the integration, an introduction to the Navier-Stokes equations for incompressible fluid flow is given first.

2.2.1 The Navier-Stokes equations

In three dimensions, considering a infinitesimal control volume and formulating the mass balance

$$\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3} = 0$$

and momentum balance

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_1 u_i)}{\partial x_1} + \frac{\partial (u_2 u_i)}{\partial x_2} + \frac{\partial (u_3 u_i)}{\partial x_3} = f_i - \frac{\partial p}{\partial x_i} + \frac{1}{\rho}\left(\frac{\partial \tau_{1i}}{\partial x_1} + \frac{\partial \tau_{2i}}{\partial x_2} + \frac{\partial \tau_{3i}}{\partial x_3}\right) \tag{2.4}$$

leads to the Navier-Stokes equations (with i being the space dimension and f_i being a volume force acting in this direction). However, the system is not complete, as there are more unknowns than equations and some constitutive law has to be defined to relate strain rates and stresses. For example, assuming inviscid flow, the stress term vanishes and the Navier-Stokes equations simplify to the Euler equations:

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_1 u_i)}{\partial x_1} + \frac{\partial (u_2 u_i)}{\partial x_2} + \frac{\partial (u_3 u_i)}{\partial x_3} = f_i - \frac{\partial p}{\partial x_i}. \tag{2.5}$$

Alternatively, the stresses can be taken proportional to the strain rates. This would result in equation 2.5 with an additional diffusion term. However, the derivation of the shallow water equations starts commonly from equation 2.4, leaving the stresses unspecified in the beginning. In the following integration over the height, the stresses acting on the vertical faces of an infinitesimal volume (in this case τ_{2i} and τ_{1i}) are assumed to be zero. The boundary stresses τ_{3i}^b and τ_{3i}^s arising from the integration of τ_{3i} are then approximated by empirical laws, describing the interaction of the fluid with the boundary. A justification of this assumption

is that in real world situations the bottom and top influences are dominant [Zienkiewicz and Taylor, 2000]. Additionally, in this form it is not possible to impose, for example, traction boundary conditions.

2.2.2 Integration of the mass conservation equation

Starting with the mass conservation, the integration from bottom to surface yields:

$$\int_H^\eta \Big(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3}\Big) dx_3 = 0$$

$$\Leftrightarrow \int_H^\eta \Big(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2}\Big) dx_3 + u_3\big|_b^s = 0. \tag{2.6}$$

For switching the integral and derivative operators the Leibniz rule for *differentiation under the integral sign* (see, for instance, Zienkiewicz *et al.* [2005]) has to be applied:

$$\int_a^b \frac{\partial}{\partial x} F(x,r)dr = \frac{\partial}{\partial x}\Big(\int_a^b F(x,r)dr\Big) - F(x,b)\frac{\partial b}{\partial x} + F(x,a)\frac{\partial a}{\partial x}, \tag{2.7}$$

where $a = a(x)$ and $b = b(x)$. Applied to equation 2.6, the derivatives are moved outside the integral and new boundary terms arise:

$$\frac{\partial}{\partial x_1}\int_H^\eta u_1\, dx_3 - u_1^s\frac{\partial \eta}{\partial x_1} + u_1^b\frac{\partial H}{\partial x_1} + \frac{\partial}{\partial x_2}\int_H^\eta u_2\, dx_3 - u_2^s\frac{\partial \eta}{\partial x_2} + u_2^b\frac{\partial H}{\partial x_2} + u_3^s - u_3^b = 0.$$

Because the exact distribution of u_i along the x_3 direction is not known, a mean value is taken, such that the integral of a velocity becomes:

$$\int_H^\eta u_i\, dx_3 = \overline{u}_i(\eta - H) = \overline{u}_i h, \tag{2.8}$$

where the overbar indicates a depth averaged quantity. Rearranging and inserting the boundary conditions of equations 2.2 and 2.3 for u_3^b and u_3^s it turns out that the boundary terms cancel except for the time derivative. The resulting mass conservation of the shallow water equations is:

$$\frac{\partial h}{\partial t} + \frac{\partial(\bar{u}_1 h)}{\partial x_1} + \frac{\partial(\bar{u}_2 h)}{\partial x_2} = 0$$

2.2.3 Integration of the momentum equations

The first step in deriving the shallow water momentum equations is the assumption that vertical velocities are small compared to horizontal velocities (and therefore the corresponding acceleration terms negligible). Consequently, the vertical Navier-Stokes momentum equation can be reduced to a hydrostatic pressure balance:

$$\frac{\partial p}{\partial x_3} + \rho g = 0 \tag{2.9}$$

Equation 2.9 can be integrated directly, resulting in the hydrostatic pressure distribution of equation 2.1.

The next step is to integrate the horizontal momentum balances of the Navier-Stokes equations (see equation 2.4) from H to η. As the derivation is identical in x_1 and x_2 dimensions for simplicity only the first momentum equation is considered:

$$\int_H^\eta \left(\frac{\partial u_1}{\partial t} + \frac{\partial (u_1^2)}{\partial x_1} + \frac{\partial (u_1 u_2)}{\partial x_2} + \frac{\partial (u_1 u_3)}{\partial x_3} + \frac{1}{\rho}\Big(\frac{\partial p}{\partial x_1} - \frac{\partial \tau_{11}}{\partial x_1} - \frac{\partial \tau_{21}}{\partial x_2} - \frac{\partial \tau_{31}}{\partial x_3} \Big) \right) = 0 \tag{2.10}$$

To transform the material derivative part

$$\begin{aligned} &\int_H^\eta \left(\frac{\partial u_1}{\partial t} + \frac{\partial (u_1^2)}{\partial x_1} + \frac{\partial (u_1 u_2)}{\partial x_2} + \frac{\partial (u_1 u_3)}{\partial x_3} \right) dx_3 \\ &= \int_H^\eta \left(\frac{\partial u_1}{\partial t} + \frac{\partial u_1^2}{\partial x_1} + \frac{\partial (u_1 u_2)}{\partial x_2} \right) dx_3 + (u_1 u_3)\big|_b^s, \end{aligned} \tag{2.11}$$

again the Leibniz rule for *differentiation under the integral sign*, shown in equation 2.7, is used to move the derivatives out of the integral. As before, the respective boundary terms cancel. It is worth to take a closer look at those manipulations, because the details of this step are often omitted in the literature.

An important difference compared to the mass conservation equation is that the momentum conservation contains non-linear, second order velocity terms. Splitting u into a mean value $\bar{u}$ and a variation u' with

$$\int_H^\eta \bar{u}\, dx_3 = \int_H^\eta u\, dx_3 \qquad \text{and} \qquad \int_H^\eta u'\, dx_3 = 0,$$

it is clear that, for example:

$$\int_H^\eta (u_1 u_2)\, dx_3 = \int_H^\eta (\bar{u}_1 + u_1')(\bar{u}_2 + u_2')\, dx_3 = \int_H^\eta (\bar{u}_1 \bar{u}_2 + \cancelto{0}{\bar{u}_1 u_2'} + \cancelto{0}{u_1' \bar{u}_2} + u_1' u_2')\, dx_3.$$

The mixed terms cancel as the integral of u' (multiplied with a constant $\bar{u}$) is equal to zero, but the so called *differential advection* term $u_1' u_2'$ is non-linear and does therefore (generally) not vanish. Together with $u' = u - \bar{u}$, this relation becomes:

$$\int_H^\eta (u_1 u_2)\, dx_3 = h \bar{u}_1 \bar{u}_2 + \overbrace{\int_H^\eta (u_1 - \bar{u}_1)(u_2 - \bar{u}_2)\, dx_3}^{\text{differential advection term} \approx 0} \approx h \bar{u}_1 \bar{u}_2. \tag{2.12}$$

As discussed in detail in section 2.8 of Vreugdenhil [1994], it is difficult to model those additional terms and therefore they are mostly omitted. However, if the influences cannot be neglected, a 3-d simulation might have to be considered.

Now, using $u_1^b = u_2^b = u_3^b = 0$ as well as $u_3^s = \frac{\partial \eta}{\partial t} + u_1^s \frac{\partial \eta}{\partial x_1} + u_2^s \frac{\partial \eta}{\partial x_2}$ both resulting from the boundary conditions described in chapter 2.1.3 and substituting in the averaged velocities of

equation 2.12, the terms of equation 2.11 can be developed as follows:

$$\int_H^\eta \frac{\partial u_1}{\partial t}\, dx_3 = \frac{\partial}{\partial t}\int_H^\eta u_1\, dx_3 - u_1^s\frac{\partial \eta}{\partial t} + u_1^b\frac{\partial H}{\partial t} = \frac{\partial (h\bar{u}_1)}{\partial t} - u_1^s\frac{\partial \eta}{\partial t} \tag{2.13}$$

$$\begin{aligned}\int_H^\eta \frac{\partial (u_1^2)}{\partial x_1}\, dx_3 &= \frac{\partial}{\partial x_1}\int_H^\eta u_1^2\, dx_3 - (u_1^s)^2\frac{\partial \eta}{\partial x_1} + (u_1^b)^2\frac{\partial H}{\partial x_1} \\ &= \frac{\partial (h\bar{u}_1^2)}{\partial x_1} - (u_1^s)^2\frac{\partial \eta}{\partial x_1}\end{aligned} \tag{2.14}$$

$$\begin{aligned}\int_H^\eta \frac{\partial (u_1u_2)}{\partial x_2}\, dx_3 &= \frac{\partial}{\partial x_2}\int_H^\eta (u_1u_2)\, dx_3 - (u_1^su_2^s)\frac{\partial \eta}{\partial x_2} + (u_1^bu_2^b)\frac{\partial H}{\partial x_2} \\ &= \frac{\partial (h\bar{u}_1\bar{u}_2)}{\partial x_2} - (u_1^su_2^s)\frac{\partial \eta}{\partial x_2}\end{aligned} \tag{2.15}$$

$$(u_1u_3)\big|_b^s = u_1^su_3^s - u_1^bu_3^b = u_1^s\frac{\partial \eta}{\partial t} + (u_1^s)^2\frac{\partial \eta}{\partial x_1} + u_1^su_2^s\frac{\partial \eta}{\partial x_2} \tag{2.16}$$

After inserting equations 2.13 - 2.16 into equation 2.10 and canceling the boundary terms, the momentum balance becomes:

$$\frac{\partial (h\bar{u}_1)}{\partial t} + \frac{\partial (h\bar{u}_1^2)}{\partial x_1} + \frac{\partial (h\bar{u}_1\bar{u}_2)}{\partial x_2} + \frac{1}{\rho}\int_H^\eta \left(\frac{\partial p}{\partial x_1} - \frac{\partial \tau_{11}}{\partial x_1} - \frac{\partial \tau_{21}}{\partial x_2} - \frac{\partial \tau_{31}}{\partial x_3}\right) dx_3 = 0$$

The pressure term can be developed in a similar manner using the hydrostatic pressure distribution (see section 2.1.2) and assuming constant density in all directions:

$$\begin{aligned}\int_H^\eta \frac{\partial p}{\partial x_1}\, dx_3 &= \int_H^\eta \frac{\partial}{\partial x_1}\Big[\rho g(\eta - x_3) + p_0\Big]\, dx_3 \\ &= \int_H^\eta \rho g\frac{\partial \eta}{\partial x_1} + \frac{\partial p_0}{\partial x_1}\, dx_3 \\ &= \rho g h\frac{\partial \eta}{\partial x_1} + h\frac{\partial p_0}{\partial x_1}.\end{aligned}$$

As the derivation of the shallow water equations is frequently done considering only inviscid flow (but still taking bottom and surface stresses into account, as mentioned in section 2.2.1), the integral of the stress terms can be simplified:

$$\int_H^\eta \left(\frac{\partial \tau_{11}}{\partial x_1} + \frac{\partial \tau_{21}}{\partial x_2} + \frac{\partial \tau_{31}}{\partial x_3}\right) dx_3 = \tau_{31}\big|_b^s$$

Together the momentum conservation part of the 2-d shallow water equations in x_1 direction is:

$$\frac{\partial (h\bar{u}_1)}{\partial t} + \frac{\partial (h\bar{u}_1^2)}{\partial x_1} + \frac{\partial (h\bar{u}_1\bar{u}_2)}{\partial x_2} + gh\frac{\partial \eta}{\partial x_1} + \frac{h}{\rho}\frac{\partial p_0}{\partial x_1} + \frac{1}{\rho}(\tau_{31}^b - \tau_{31}^s) = 0. \tag{2.17}$$

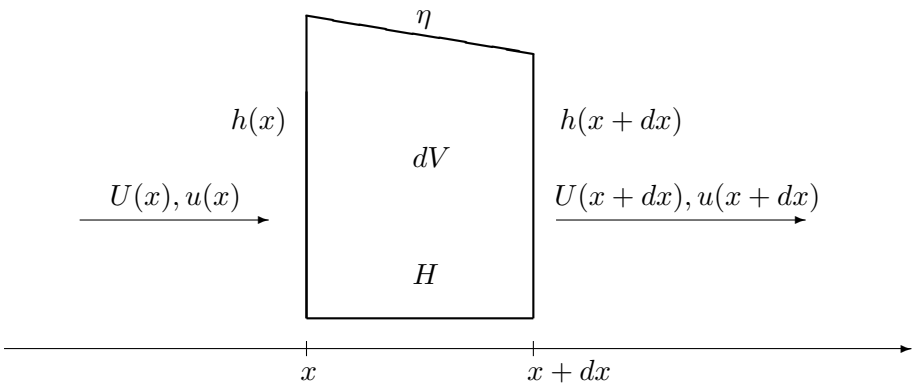


Figure 2.2: Mass balance on one dimensional infinitesimal control volume

2.3 Conservation laws on an infinitesimal control volume

The second and probably more straightforward way to derive the shallow water equations as presented in Plumb [2003] is to directly apply mass and momentum equations on an infinitesimal control volume. Because this is just an alternative way to the strategy described in the previous chapter, for simplification reasons just the one dimensional case is considered. Figure 2.2 describes an infinitesimal control volume with U being the mass flow. Now, the mass conservation requires the condition

$$\frac{\partial m}{\partial t} = \rho\, U(x) - \rho\, U(x + dx) \tag{2.18}$$

to hold. Substituting $U = u \cdot h$ and $m = \rho h dx$ into equation 2.18, the following expression arises:

$$\rho \frac{\partial h}{\partial t} dx = \rho u(x) h(x) - \rho u(x + dx) h(x + dx) = -\rho \frac{\partial (uh)}{\partial x} dx$$

Finally, after canceling ρ and dx, the 1-d shallow water mass conservation equation reads:

$$\frac{\partial h}{\partial t} + \frac{\partial (uh)}{\partial x} = 0.$$

The momentum equations are derived in a similar manner. The terms contributing to the momentum balance are:

- hydrostatic pressure on both sides
- mass in- and outflow (cancels in moving coordinates)
- atmospheric pressure on the water surface

as well source terms (including wind drag force and bottom friction) which are discussed in section 2.4 and therefore not considered here. A summary of all terms can be seen in in figure 2.3, where influences due to mass flow are written in parentheses as they do not contribute to the momentum balance in moving coordinates.

The net force resulting from the atmospheric pressure can be split in a horizontal part which contributes to the momentum balance and a vertical part which does not cause any change in

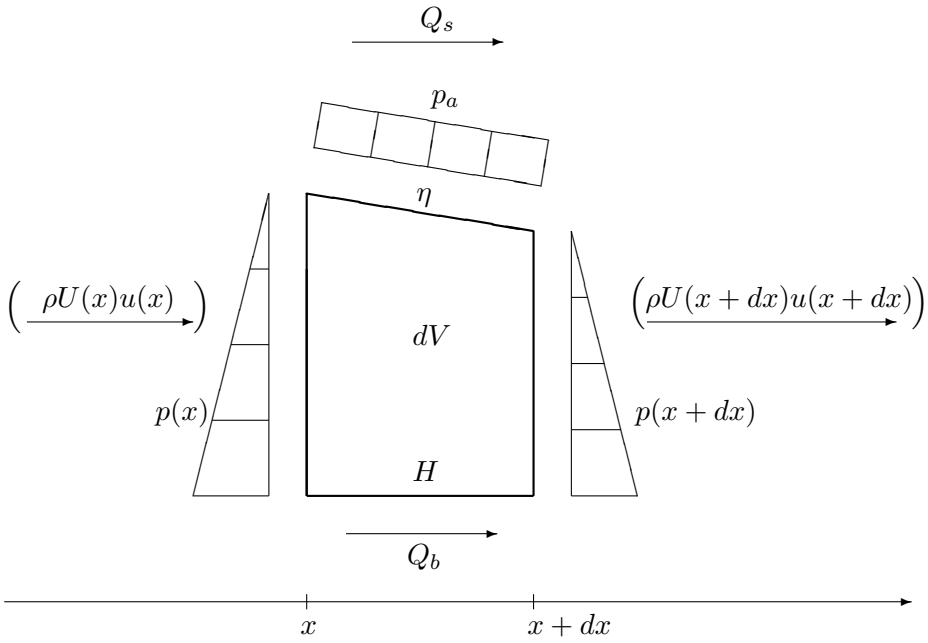


Figure 2.3: Momentum balance on one dimensional infinitesimal control volume

the motion of the fluid. The horizontal part can be extracted by multiplying the atmospheric pressure p_0 with the difference in the surface elevation $\Delta h = \frac{\partial h}{\partial x} dx$:

$$P_{0,h} = p_0 \frac{\partial h}{\partial x} dx.$$

In the next step, the force arising from differences in the pressure acting on the left and right side is calculated. The integral of the hydrostatic pressure (see section 2.1.2) is:

$$P = \frac{\rho g h^2}{2} + p_0 h,$$

and thus the difference between the net forces on both side of the control volume gives:

$$\begin{aligned} P_{left} - P_{right} &= \frac{1}{2}\rho g h(x)^2 + p_0 h(x) - \frac{1}{2}\rho g h(x+dx)^2 - p_0 h(x+dx) \\ &= \frac{1}{2}\rho g\big(h(x)^2 - h(x+dx)^2 \big) + p_0\big(h(x) - h(x+dx) \big) \\ &= \frac{1}{2}\rho g\Big(h(x) + h(x+dx) \Big) \Big(h(x) - h(x+dx) \Big) \\ &\quad + p_0\Big(h(x) - h(x+dx) \Big) \\ &= -\rho g h \frac{\partial h}{\partial x}\, dx - p_0 \frac{\partial h}{\partial x}\, dx \end{aligned}$$

Applying Newtons law of motion $F = ma = m\frac{Du}{Dt}$, the momentum balance for moving coordinates or in other words in Lagrange formulation can be written as:

$$m\frac{Du}{Dt} = P_{left} - P_{right} + P_{0,h}, \tag{2.19}$$

with Du/Dt being the material derivative, defined as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x}.$$

Substituting the corresponding net pressure terms as well as $m = \rho h\, dx$, equation 2.19 becomes:

$$\begin{aligned} \rho h \frac{Du}{Dt}\, dx &= -\rho g h \frac{\partial h}{\partial x}\, dx - p_0 \frac{\partial h}{\partial x}\, dx + p_0 \frac{\partial h}{\partial x} dx \\ \Leftrightarrow \frac{Du}{Dt} &= -g\frac{\partial h}{\partial x}. \end{aligned}$$

After applying the material derivative to change into a Eulerian coordinate system, the resulting expression is the final 1-d momentum conservation equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial h}{\partial x}.$$

2.4 Source terms

As the shallow water equations are used to describe many different types of problems also on different scales, there are a lot of source terms that could be considered. This ranges from Coriolis force dominated oceanic flow to small scale flows in rivers with a high influence of bottom friction and gravity. The most commonly used (and in this section presented) terms are: wind drag force, bottom friction, gravity in combination with sloped bottom topography and Coriolis forces.

2.4.1 Bottom friction

The bottom friction is of particular importance if the viscosity is assumed to be zero. In combination with a sloped bottom, the total energy of the system would grow constantly and a steady state would not be reached if no friction is considered. However, the exact value is generally unknown and has to be estimated.

A commonly used empiric formula to determine the mean velocity of turbulent open channel flows is the Manning-Strickler formula (see, for instance, Rössert [1999]):

$$u = k_{\text{st}} \cdot \sqrt{I} \cdot R_h^{\frac{2}{3}}, \tag{2.20}$$

where the Strickler coefficient k_{st} is a empiric constant, I is the absolute bottom slope (i.e. for 1% slope I would be 0.01) and R_h is the hydraulic radius. The hydraulic radius is defined as the ratio of the cross section area and the wetted perimeter. In the case of shallow water flow it can be reduced to just the water height h (because the horizontal scale is assumed to be dominant). Table 2.1 shows a few examples for Strickler-coefficients for different types of soil. To develop a shallow water model, that coincides with the Manning-Strickler formula, first the force balance between gravity and friction is formulated:

$$\tau_{3i}^b = \rho g h I. \tag{2.21}$$

material	k_{st}
sand	50
fine gravel	40
gravel	35
torrent with coarse gravel	25 - 28
torrent with moving coarse gravel	19 - 22
very rough rock	15 - 20
low vegetation	30 - 35
high vegetation	20 - 25
smoothed concrete	90
rough concrete	55
iron	96 - 120
iron, strongly rusted	67 - 80

Table 2.1: Strickler coefficients for different soil types [Rössert, 1999, 47,48]

Subsequently, the Manning-Strickler equation (2.20) is reformulated and substituted into equation 2.21, leading to the following expression for the bottom friction source term:

$$\tau_{3i}^{b} = \frac{\rho g u_i |\mathbf{u}|}{k_{st}^2 h^{\frac{1}{3}}}.$$

Note that the same result can be achieved by substituting $C = k_{\mathrm{st}} h^{\frac{1}{6}}$ into the Chézy expression, defined as

$$\tau_{3i}^{b} = \frac{\rho g u_i |\mathbf{u}|}{C^2},$$

with the Chézy coefficient C [Weiyan, 1992].

However, this treatment of the bottom friction is just a vague estimation and in some cases where higher accuracy must be achieved, different models might have to be implemented. An overview of bottom friction models for free surface flows is given by Olivier *et al.* [2009].

2.4.2 Sloped bottom topography

Often, the bottom topography is not totally flat. In those cases, the gravitation has to be taken into account. Substituting $\eta = h + H$ into the pressure term of (2.17), the influence of the bottom slope can be seen directly:

$$gh\frac{\partial \eta}{\partial x_1} = gh\frac{\partial h}{\partial x_1} + gh\frac{\partial H}{\partial x_1},$$

where the second term can be interpreted as the horizontal component of the gravitational force $\rho g h$ (equation 2.17 was divided by the density, so ρ doesn't appear in the above equation).

Another possibility often used together with the CBS algorithm presented in section 3.3 is

to use the alternative pressure term and its corresponding ground slope source term:

$$gh\frac{\partial \eta}{\partial x_i} = \frac{\partial}{\partial x_1}\left(g\,\frac{h^2 - H^2}{2} \right) + g(h+H)\frac{\partial H}{\partial x_i} \tag{2.22}$$

Which can be validated by some simple manipulations using $\eta = h + H$:

$$\begin{aligned}
\frac{\partial}{\partial x_1}\Big[\frac{1}{2}g(h^2 - H^2)\Big] &= \frac{1}{2}g\,\frac{\partial}{\partial x_1}\Big[(h-H)(h+H)\Big] \\
&= \frac{1}{2}g\Big[(h-H)\frac{\partial}{\partial x_i}(h+H) + (h+H)\frac{\partial}{\partial x_1}(h-H)\Big] \\
&= \frac{1}{2}g\Big[h\frac{\partial h}{\partial x} - H\frac{\partial h}{\partial x} + h\frac{\partial H}{\partial x} - H\frac{\partial H}{\partial x} + h\frac{\partial h}{\partial x} + H\frac{\partial h}{\partial x} - h\frac{\partial H}{\partial x} - H\frac{\partial H}{\partial x}\Big] \\
&= gh\frac{\partial h}{\partial x_i} - gH\frac{\partial H}{\partial x_i} = gh\frac{\partial(\eta - H)}{\partial x_i} - g(\eta - h)\frac{\partial H}{\partial x_i} \\
&= gh\frac{\partial \eta}{\partial x_i} - g\eta\frac{\partial H}{\partial x_i}
\end{aligned}$$

2.4.3 Other source terms

It is also straight forward to include Coriolis forces f. These give after depth integration the following source terms:

$$\begin{aligned}
f_1 &= -h\hat{f}\bar{u}_2 \\
f_2 &= h\hat{f}\bar{u}_1,
\end{aligned}$$

where $\hat{f}$ is the Coriolis parameter [Zienkiewicz *et al.*, 2005].

If a wind drag force is considered, an empiric expression similar to the one used for describing the bottom friction needs to be applied. A commonly used model is:

$$\tau_i^s = \gamma v_i |\mathbf{v}|$$

with γ being a constant model factor and v being a given wind velocity field (see, for instance, Vreugdenhil [1994]).

2.5 Boundary conditions

As a system of partial differential equations is not complete by itself, specifying the right boundary conditions is crucial. However, there are certain limitations for the application of boundary conditions, depending on the type of flow on the boundary. For example, on a region with super-critical inflow, all values (water surface elevation as well as both velocity components) have to be specified. The physical interpretation is that no information can be transported upstream (wave celerity > velocity), and thus there is no way that values at the inflow boundary can be determined by their values downstream. On the other hand for sub-critical inflow only 2 boundary conditions can be imposed, as the values are now indeed

flow type	number of boundary conditions
super-critical inflow	3
sub-critical inflow	2
sub-critical outflow	1
super-critical outflow	0

Table 2.2: Number of boundary-conditions for different flow types [Vreugdenhil, 1994]

dependent on the flow downstream. Table 2.2 shows the number of boundary conditions that have to be imposed, dependent on the Froude number

$$Fr = \frac{\text{flow velocity}}{\text{wave celerity}} = \frac{u}{\sqrt{gh}}, \tag{2.23}$$

which describes the flow type ($Fr > 1$ for super-critical and $Fr < 1$ for sub-critical flow).

Besides conditions on h and **U**, different other types of boundary conditions can be applied. Especially the simulation of an open domain is in many cases necessary due to the limited computational capacity. In this case, waves have to be able to leave the domain without reflections. For more detailed information about advanced boundary conditions the reader is referred to the different methods presented in the literature (see, for example, Vreugdenhil [1994]). However, in section 4.4 it is shown that using the Manning-Strickler formula to set outflow velocities is in many practical cases sufficient for simulating an open domain.

2.6 The vector form of the shallow water equations

The full 2-d shallow water equations are commonly written in a vector form of the type

$$\frac{\partial \mathbf{\Phi}}{\partial t} + \frac{\partial \mathbf{F_i}}{\partial x_i} + \mathbf{Q} = \mathbf{0},$$

with $i = 1, 2$ as well as

$$\mathbf{\Phi} = \begin{Bmatrix} h \\ h\overline{u}_1 \\ h\overline{u}_2 \end{Bmatrix}$$

$$\mathbf{F_i} = \begin{Bmatrix} h\overline{u}_i \\ h\overline{u}_1\overline{u}_i + \delta_{1i}\frac{1}{2}g(h^2 - H^2) \\ h\overline{u}_2\overline{u}_i + \delta_{2i}\frac{1}{2}g(h^2 - H^2) \end{Bmatrix}$$

$$\mathbf{Q} = \begin{Bmatrix} 0 \\ -h\hat{f}\bar{u}_2 + g(h+H)\dfrac{\partial H}{\partial x_1} + \dfrac{h}{\rho}\dfrac{\partial p_a}{\partial x_1} - \dfrac{1}{\rho}\tau^s_{31} + \dfrac{g\bar{u}_1|\bar{\mathbf{u}}|}{k^2_{st}h^{1/3}} \\ h\hat{f}\bar{u}_1 + g(h+H)\dfrac{\partial H}{\partial x_2} + \dfrac{h}{\rho}\dfrac{\partial p_a}{\partial x_2} - \dfrac{1}{\rho}\tau^s_{32} + \dfrac{g\bar{u}_2|\bar{\mathbf{u}}|}{k^2_{st}h^{1/3}} \end{Bmatrix},$$

where δ_{ij} is the Kronecker Delta with the values 1 for $i = j$ and 0 for $j \neq 0$, $\hat{f}$ is the Coriolis force, τ^s_{3i} is the surface traction and k_{st} is the Strickler coefficient. Additionally, for repeated indices the Einstein summation convention applies. Note that $k^2_{st}h^{1/3}$ could also be substituted by the square of the Chézy coefficient C^2.

2.7 Summary of properties related to the shallow water equations

The shallow water equations are a special system of partial differential equations. The 2-d version describes a 3-d incompressible fluid flow with a hydrostatic pressure assumption. This section serves as a summary of the behavior of the shallow water equations and recapitulates some important assumptions, that have been made during the derivation.

2.7.1 Comparison with high speed compressible gas flow

As that the shallow water equations describe a fluid flow of one dimension higher than the dimensionality of the system of equations (so 3-d flow would be described by a 2-d equation system), there is an additional direction where the fluid 'can go' if the net pressure increases. As a result, although the fluid is incompressible, the shallow water equations are of a compressible type similar to the Euler equations for compressible flow. That means that solutions can form shock waves similar to those observed in compressible gas flows. An example of this phenomenon can be seen in figure 5.3 where the smooth Gaussian bell initial condition forms a shock after some time. This kind of information is very important for numerical solution procedures as it is generally difficult to approximate shocks without major oscillations around the discontinuity. Strategies to handle such phenomena numerically are discussed in section 4.7.

2.7.2 Correspondence to the incompressible Navier-Stokes equations

As pointed out in Vreugdenhil [1994], the derivation of the shallow water equations was done by considering incompressible Newtonian flow. Consequently, the resulting solutions become identical in the case of zero water surface fluctuations. As a consequence, for small deviations of the surface height compared to the total water height, the 3-d Navier-Stokes equations for incompressible flow with a frictionless fixed upper bound give similar results.

2.7.3 Valid circumstances for the ‘shallow water‘ assumption

The basic assumption of a *hydrostatic pressure* (see section 2.1.2) is not generally valid for a three dimensional fluid flow. For example, breaking waves involve a pressure distribution very different from hydrostatic. It has to be examined under what circumstances this assumption is valid. According to Vreugdenhil [1994], all horizontal scales have to be much larger than all vertical scales where the factor of ‘much larger‘ is not exactly defined but a guiding value could be about 20 times. Vertical scales are: water depth h, thickness of boundary layers, variation of the bottom topography H and variation of the water surface η. Horizontal scales include: size of the considered flow geometry, size of the bottom topography, distances between variations of external influences and the wave length. So the quality of the shallow water assumption can be estimated by comparing the the relevant vertical and horizontal scales. Additionally, it is important to remember that the derivation was done including the assumptions of constant density in all directions, time invariant bottom topography and inviscid fluid flow.

Chapter 3

Finite element approaches to the governing equations

As seen in section 2, one major characteristic of the shallow water equations is the convective term, which renders the problem *non-self-adjoint*. To study the behavior of different finite element methods for fluid problems, the convection diffusion equation is commonly used as a model problem. Using the summation convention for repeated indices, the steady state problem is formulated as follows:

$$\frac{\partial(U_i\phi)}{\partial x_i} - \frac{\partial}{\partial x_i}\left(k\frac{\partial\phi}{\partial x_i}\right) + Q = 0 \qquad \text{in } \Omega \tag{3.1a}$$

$$\phi = \phi_D \qquad \text{at } \Gamma_D \tag{3.1b}$$

$$n \cdot k\frac{d\phi}{dx} = q_n \qquad \text{at } \Gamma_N \tag{3.1c}$$

Applying the product rule to equation 3.1a yields:

$$U_i\frac{\partial(\phi)}{\partial x_i} + \phi\frac{\partial(U_i)}{\partial x_i} - \frac{\partial}{\partial x_i}\left(k\frac{\partial\phi}{\partial x_i}\right) + Q = 0$$

which can be reduced to

$$U_i\frac{\partial(\phi)}{\partial x_i} - \frac{\partial}{\partial x_i}\left(k\frac{\partial\phi}{\partial x_i}\right) + Q = 0 \tag{3.2}$$

if the divergence $\nabla \cdot U = \partial U_i/\partial x_i$ of the velocity field equals zero (for flow fields resulting from incompressible fluids). In many cases only the one dimensional version

$$U\frac{d\phi}{dx} - \frac{d}{dx}\left(k\frac{d\phi}{dx}\right) + Q = 0 \tag{3.3}$$

is considered.

3.1 Difficulties arising from the application of standard Galerkin methods to convection dominated problems

Applying the standard Galerkin approximation to the 1-d steady state convection-diffusion equation, as described in section 1.3.3, the weak form

$$\int_\Omega w\Big[U\frac{d\phi}{dx} - \frac{d}{dx}\Big(k\frac{d\phi}{dx}\Big) + Q\Big]d\Omega = 0 \tag{3.4}$$

is discretized with

$$\phi \approx \hat{\phi} = \mathbf{N}\tilde{\boldsymbol{\phi}},$$

where $\boldsymbol{N}$ is the vector of basis functions and $\tilde{\boldsymbol{\phi}}$ is the vector of degree of freedom values. After integrating the second term by parts, the weak form becomes:

$$\int_\Omega \Big(\hat{w}\,U\frac{d\hat{\phi}}{dx} + \frac{d\hat{w}}{dx}k\frac{d\hat{\phi}}{dx} + \hat{w}Q\Big)d\Omega + \int_{\Gamma_N} \hat{w}q_n\,d\Gamma_N = 0. \tag{3.5}$$

Inserting the Galerkin discretization yields to the compact matrix form

$$\mathbf{K}\tilde{\boldsymbol{\phi}} = \mathbf{f}$$

with

$$\mathbf{K} = \int_\Omega \boldsymbol{N}^T U\frac{d\boldsymbol{N}}{dx} + \frac{d\boldsymbol{N}^T}{dx}k\frac{d\boldsymbol{N}}{dx}d\Omega$$
$$\mathbf{f} = -\int_{\Gamma_N} \boldsymbol{N}^T q_n\,d\Omega - \int_\Omega \boldsymbol{N}^T Q d\Omega.$$

As shown by Donea and Huerta [2003], for the choice of standard linear basis functions the discrete system arising from the Galerkin method is identical (except for the source term) to the one arising from the finite difference method using central difference operators . Before starting to analyze the finite element solutions for different combinations of U and k, the element Péclett number is defined as

$$\mathrm{Pe} = \frac{Uh}{2k}$$

to determine if the problem is convection- or diffusion dominated. The higher the Péclett number the more the problem is (locally) dominated by the convective term.

In chapter 2 of Donea and Huerta [2003], the steady convection-diffusion equation was solved with the standard Galerkin method for different Péclett numbers and it was shown that the solutions becomes inaccurate and oscillatory if $Pe > 1$ (see also figure 3.4).

A straight forward explanation for this phenomenon is that the value of ϕ at one point is in the case of pure convection only dependent on the solution upstream. However, the numerical influence of one node in the Galerkin method is equally distributed in both directions (in 1-d case) as shown in figure 3.1. Consequently, the achieved solution is not optimal as the range of the numerical influence is larger than the actual physical range of influence.

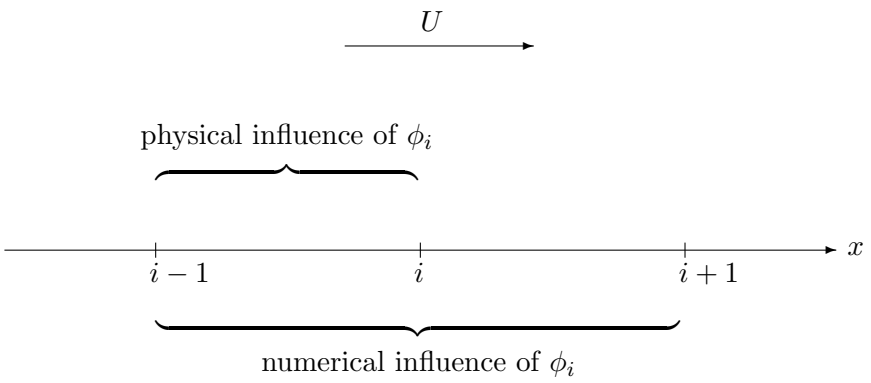


Figure 3.1: Numerical influence in the finite element method

3.1.1 Loss of the best-approximation property

In problems in fluid mechanics, the existence of convective terms renders the corresponding operator non self-adjoint. The purpose of this section is to follow a more mathematical approach and investigate how the error of the Galerkin method behaves for these types of operators.

The property of an operator being self-adjoint can be seen as generalization of a matrix (i.e. a discrete operator) being its own transpose (and therefore symmetric).

Just as for the scalar product $<\cdot,\cdot>$ the transpose A^T of a real valued matrix satisfies the following identity:

$$<Av, w> = <v, A^T w>$$

for the generalized inner product $(\cdot,\cdot)$, the adjoint A^* is defined as the operator that makes the statement

$$(Av, w) \;=\; (v, A^* w)$$

valid. Consequently if

$$(Av, w) \;=\; (v, Aw),$$

A is its own adjoint (i.e. self-adjoint).

Generally the procedure for obtaining the adjoint operator is to transfer the derivatives acting on v by an integration by parts to w. If there is no more derivative acting on v, the operator on w is the adjoint of A. For illustrating this procedure, the adjoint of the Laplace operator

$$A = \Delta = \left(\frac{\partial^2}{\partial x^2} \right)$$

applied to the finite element solution u is computed. Starting from the weak form (meaning the L_2 inner product of $A(u)$ with a test function v), the calculation can be done as follows

(without loss of generality assuming homogeneous Dirichlet boundary conditions):

$$\int_{x_1}^{x_2} v \cdot A(u)\, dx = \int_{x_1}^{x_2} v \cdot \frac{\partial^2 u}{\partial x^2}\, dx = \int_{x_1}^{x_2} -\frac{\partial v}{\partial x} \cdot \frac{\partial u}{\partial x}\, dx + \cancelto{0}{v\frac{\partial u}{\partial x}\Big|_{x_1}^{x_2}}$$

$$= \int_{x_1}^{x_2} \frac{\partial^2 v}{\partial x^2} \cdot u\, dx - \cancelto{0}{\frac{\partial v}{\partial x} u\Big|_{x_1}^{x_2}} = \int_{x_1}^{x_2} A^*(v) \cdot u\, dx$$

$$\text{with } A^* = \left(\frac{\partial^2}{\partial x^2}\right) = A,$$

where the boundary evaluation terms arising from the integration by parts cancel either due to the Dirichlet boundary conditions ($u(0) = u(1) = 0$) or by the definition of v being 0 at the Dirichlet boundary. This is valid as the definition of an operator is connected with a domain and boundary conditions. With homogeneous Dirichlet boundaries (like in this example) the domain of definition of A and its adjoint is:

$$D(A) = \{\ u \text{ on } \Omega \mid u = 0 \text{ on } \partial\Omega\ \}$$
$$D(A^*) = \{\ v \text{ on } \Omega \mid v = 0 \text{ on } \partial\Omega\ \}.$$

This example shows that the Laplace operator is *self-adjoint* as the two minus signs cancel. However, applying the same procedure to the convective term of the convection-diffusion equation

$$\int_{x_1}^{x_2} v \cdot U\frac{\partial \phi}{\partial x}\, dx = \int_{x_1}^{x_2} -U\frac{\partial v}{\partial x} \cdot \phi\, dx + \cancelto{0}{vU\phi\Big|_{x_1}^{x_2}}$$

a minus sign remains after canceling the additional term due to the Dirichlet boundary conditions. Now it can be seen that

$$A = \left(U\frac{\partial}{\partial x}\right) \neq \left(-U\frac{\partial}{\partial x}\right) = A^*,$$

and therefore the convection-diffusion equation as well as all fluid-dynamics equations that contain a convective term are a *non-self-adjoint* problems.

It is also clear that if the bilinear form B is required to be symmetric (i.e. $B(v,u) = B(u,v)$) the differential operator must be self-adjoint (for the Galerkin method):

$$B(v,u) = (v, Au) = (Au, v) = \underline{(u, A^*v) = (u, Av)} = B(u,v),$$

which is only valid if $A = A^*$.

To explain why the Galerkin method fails for convection dominated problems, its optimality in the case of self-adjoint problems is proven. In the process of this prove, it is pointed out where the property $B(u,v) = B(v,u)$ is needed. Starting from the discrete problem

$$\text{Find } \hat{u} \in X^h \text{ such that } B(\hat{u}, \hat{v}) = f(\hat{v}) \qquad \forall \hat{v} \in Y^h \tag{3.6}$$

as explained in Hughes [2000] (now using $e = \hat{u} - u$ as the discretization error) the *best-approximation property* of the Galerkin method for self-adjoint problems is formulated as

(proof follows afterwards):

$$B(e,e) \;\leq\; B(\hat{U}-u,\hat{U}-u) \qquad \text{for all } \hat{U}\in X^h, \tag{3.7}$$

meaning that the error in the *energy norm* defined as

$$\|e\|_{E_\Omega} = \sqrt{\frac{1}{2}B(e,e)} \tag{3.8}$$

is for any arbitrary element $\hat{U}$ from the trial function space X^h bigger or equal to the error e of the finite element solution. Now, because $Y^h \subset Y$ it is possible to substitute $\hat{v}$ for v in the continuous problem of equation 1.5, resulting in:

$$B(u,\hat{v}) = f(\hat{v}) \qquad \forall \hat{v}\in Y^h \tag{3.9}$$

Subtracting equation 3.9 from 3.6 the right hand size cancels, and the left side can be combined due to the linearity of B:

$$\begin{aligned} & B(\hat{u},\hat{v}) - B(u,\hat{v}) = 0 \qquad && \forall \hat{v}\in Y^h \\ \Leftrightarrow\quad & B(e,\hat{v}) = 0 && \forall \hat{v}\in Y^h, \end{aligned} \tag{3.10}$$

meaning that the error is orthogonal to Y^h and because $X^h = Y^h$ also to the finite element subspace (therefore 3.10 is often called *Galerkin-orthogonality*). Moving on by adding an arbitrary $\hat{v}\in Y^h$ leads to:

$$\begin{aligned} B(e+\hat{v},e+\hat{v}) &= B(e+\hat{v},e) + B(e+\hat{v},\hat{v}) \\ &= B(e,e) + B(\hat{v},e) + B(e,\hat{v}) + B(\hat{v},\hat{v}) \\ &= B(e,e) + 2\,B(e,\hat{v}) + B(\hat{v},\hat{v}) \end{aligned}$$

in which the last step can only be done if B is *symmetric* (such that $B(\hat{v},e) = B(e,\hat{v})$). Because of equation 3.10 and the fact that $B(\hat{v},\hat{v}) \geq 0$, it follows that

$$B(e,e) \leq B(e+\hat{v},e+\hat{v}). \tag{3.11}$$

Now, any $\hat{U}\in X^h$ can be written as a combination of $\hat{u}$ and $\hat{v}$ (because the spaces X^h and Y^h are equal in the Galerkin method) and therefore

$$\begin{aligned} e+\hat{v} &= \hat{u}-u+\hat{v} \\ &= \hat{U}-u \end{aligned}$$

can be substituted into 3.11 and equation 3.7 be proven.

Summarized the Galerkin method does not perform optimal for non-self adjoint problems because although the error is orthogonal for $B(\hat{e},\hat{v})$ it is not for $B(\hat{v},\hat{e})$ and as a consequence the best-approximation property is lost. An error analysis on the finite element method applied to the convection-diffusion equation is given by Babuška *et al.* [1982].

3.1.2 Under-diffusive behavior

Having pointed out that the Galerkin method is not optimal for convection dominated problems, it might be interesting to find out, in what way this will affect the behavior of the solution. A detailed analysis on this field, given in Donea and Huerta [2003], shows that in fact the standard Galerkin weighting introduces a truncation error with the form of a negative diffusion. Consequently the corresponding problem for which the finite element solution is optimal and in fact exact (for the 1-d convection-diffusion equation) is formulated as follows:

$$U\frac{d\phi}{dx} - \left[k - \bar{k}\right]\frac{d^2\phi}{dx^2} + Q = 0,$$

$$\text{with} \qquad \bar{k} = \alpha\,\frac{\sinh^2 Pe}{Pe}\,k \qquad \text{and} \qquad \alpha = \coth Pe - \frac{1}{Pe}.$$

Note: the derivation was done for equally sized elements with linear basis functions and the Dirichlet boundary conditions $u = 0$ for both sides. It is important to emphasize at this stage, that stabilizing techniques (some of which are introduced in the following chapter) are trying to *balance* this lack of diffusion by different strategies. Therefore, if they would perfectly succeed in doing this, no error would be introduced by the additional diffusion.

3.2 Different strategies for stabilizing the finite element solution

To achieve good result for convection dominated problem with the standard Galerkin method some modifications have to be made. A lot of different approaches were developed in the past but none of them really succeeded in dominating the field. While some methods like the discontinuous Galerkin method are heading a different direction, others try to adopt the Galerkin FEM by applying stabilization techniques. As a comprehensive description of those methods would go beyond the scope of this work, only an overview of important procedures is given in this section, following the works of Brooks and Hughes [1982], Zienkiewicz *et al.* [2005], Donea and Huerta [2003] and Löhner *et al.* [1984].

3.2.1 Upwind schemes in the finite difference method

The first progress in stabilizing oscillations were made in the finite difference context. The use of upwind differences with first order accuracy instead of central differences with second order accuracy proved to produce stable solutions at the cost of a over-diffusive behavior. By reducing the dependency of the solution to the node upstream, it was possible so simulate the direction dependent information propagation of the convection-diffusion equation. Inserting

$$\frac{d\phi}{dx} \approx \frac{\tilde{\phi}_i - \tilde{\phi}_{i-1}}{h} \qquad \text{and} \qquad \frac{d^2\phi}{dx^2} \approx \frac{\tilde{\phi}_{i-1} - 2\tilde{\phi}_i + \tilde{\phi}_{i+1}}{h^2}$$

into equation 3.3 yields the upwind difference equation for node i

$$U\frac{\tilde{\phi}_i - \tilde{\phi}_{i-1}}{h} - k\frac{\tilde{\phi}_{i-1} - 2\tilde{\phi}_i + \tilde{\phi}_{i+1}}{h^2} + Q = 0,$$

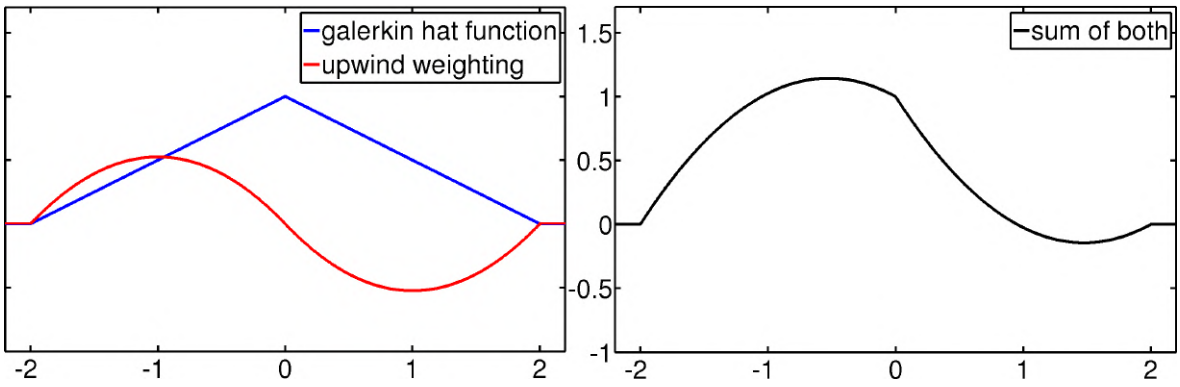


Figure 3.2: Original Petrov-Galerkin weighting

which can be rewritten in terms of the element Péclett number after multiplication with h^2/k and some reordering:

$$(-2Pe-1)\tilde{\phi}_{i-1} + (2+2Pe)\tilde{\phi}_i - \tilde{\phi}_{i+1} + \frac{Qh^2}{k} = 0$$

With this discretization, exact nodal values are obtained for pure convection (Pe $= \infty$) and pure diffusion (Pe $= 0$). For mixed problems the nodal values deviate from the exact solution but no oscillations arise.

3.2.2 Petrov-Galerkin methods for the 1D convection diffusion equation

To combine the benefits of the finite element method with the stability of upwinding techniques, discovered in the finite difference method, the family of *Petrov-Galerkin methods* was developed. The basic straightforward idea is to take test functions from a different function space than the trial functions to respects the directed dependency by heavier weighting of the upwind region. As an example, for linear shape functions the new set of test functions were achieved by adding an upwind part

$$\hat{w}_i = N_i + \alpha \hat{w}_i^*,$$

with the upwinding parameter α. Clearly, the additional part is dependent on the flow direction $sign(U)$ or $\frac{\mathbf{U}}{|\mathbf{U}|}$ and to enable comparisons with the finite difference method its integral is chosen to be $\frac{h}{2}$:

$$\int_0^L \hat{w}_i^* dx = \pm\frac{h}{2}.$$

The first upwind weighting functions were 'bubbles' as shown in figure 3.2, represented by parabolas with value 0 at the nodes. However, this requires higher order shape functions and therefore it is desirable to develop a formulation, that retains a C^0 continuity demand. A different choice, plotted in figure 3.3, takes the standard Galerkin hat function and adds a portion of its derivative, resulting in the formulation

$$\hat{w}_i^* = \frac{h}{2}\frac{dN_i}{dx}\text{sign}(U)$$

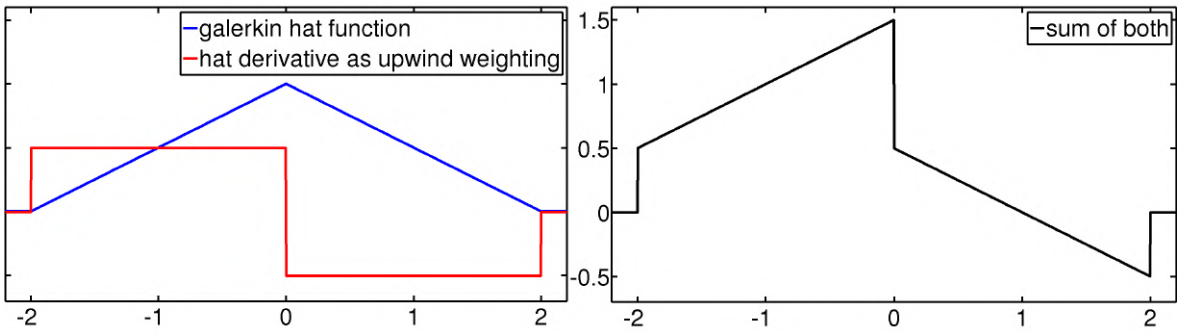


Figure 3.3: Weighting with the shape function derivative for introducing asymmetry

This idea was later also used and generalized to higher dimensions by the SUPG method, introduced in section 3.2.3. However, it is obvious that this weighting function is discontinuous. Because the integral is well defined this doesn't lead to difficulties as long as no integration by parts is performed. In this case a smoothing procedure can be applied. Applied to equation 3.3 the nodal equation becomes:

$$[-\mathrm{Pe}(\alpha+1)-1]\tilde{\phi}_{i-1}+[2+2\alpha(\mathrm{Pe})]\tilde{\phi}_i+[-\mathrm{Pe}(\alpha-1)-1]\tilde{\phi}_{i+1}+\frac{Qh^2}{k}=0, \tag{3.12}$$

which simplifies to the standard Galerkin method if $\alpha = 0$ and is equivalent to the upwind difference discretization if $\alpha = 1$. Moreover, it has been shown in Donea and Huerta [2003] that the exact solution for all nodes are obtained if the upwind parameter α is chosen as follows:

$$\alpha = \alpha_{\mathrm{opt}} = \coth|Pe| - \frac{1}{|Pe|}.$$

However, the presented method has serious drawbacks concerning treatment of source terms, time dependent behavior and multidimensional generalizations. A comparison of the Petrov-Galerkin method for different values of α with the standard Galerkin method is given in figure 3.4. It can be seen that similar to the upwind discretization in the finite difference method, choosing $\alpha = 1$ results in a very stable but over-diffusive solution.

3.2.3 The Streamline-Upwind Petrov-Galerkin (SUPG) method

Experimenting with upwind differences and Petrov-Galerkin formulations it was soon realized that a nodal equation identical to 3.12 can be obtained by adding an additional *balancing diffusion* to the original differential equation. This coincides with the *under-diffusive* behavior of the standard Galerkin method as shown in section 3.1. As the implementation of an additional diffusion is much easier than the implementation of the Petrov-Galerkin weighting functions, special attention was devoted to this perspective. However, adding only an additional diffusion leads to an inconsistency of the method because only one part of the weak form has been modified. Additionally, in the multiple dimensional case the balancing diffusion has to be directed upstream to prevent undesirable crosswind diffusion. The SUPG method takes this into account and adds an additional upwind weighting term to the standard test functions:

$$\hat{w}_i = N_i + \alpha\hat{w}_i^* = N_i + \alpha\frac{h}{2}\frac{\boldsymbol{U}}{|\boldsymbol{U}|}\cdot\nabla N_i,$$

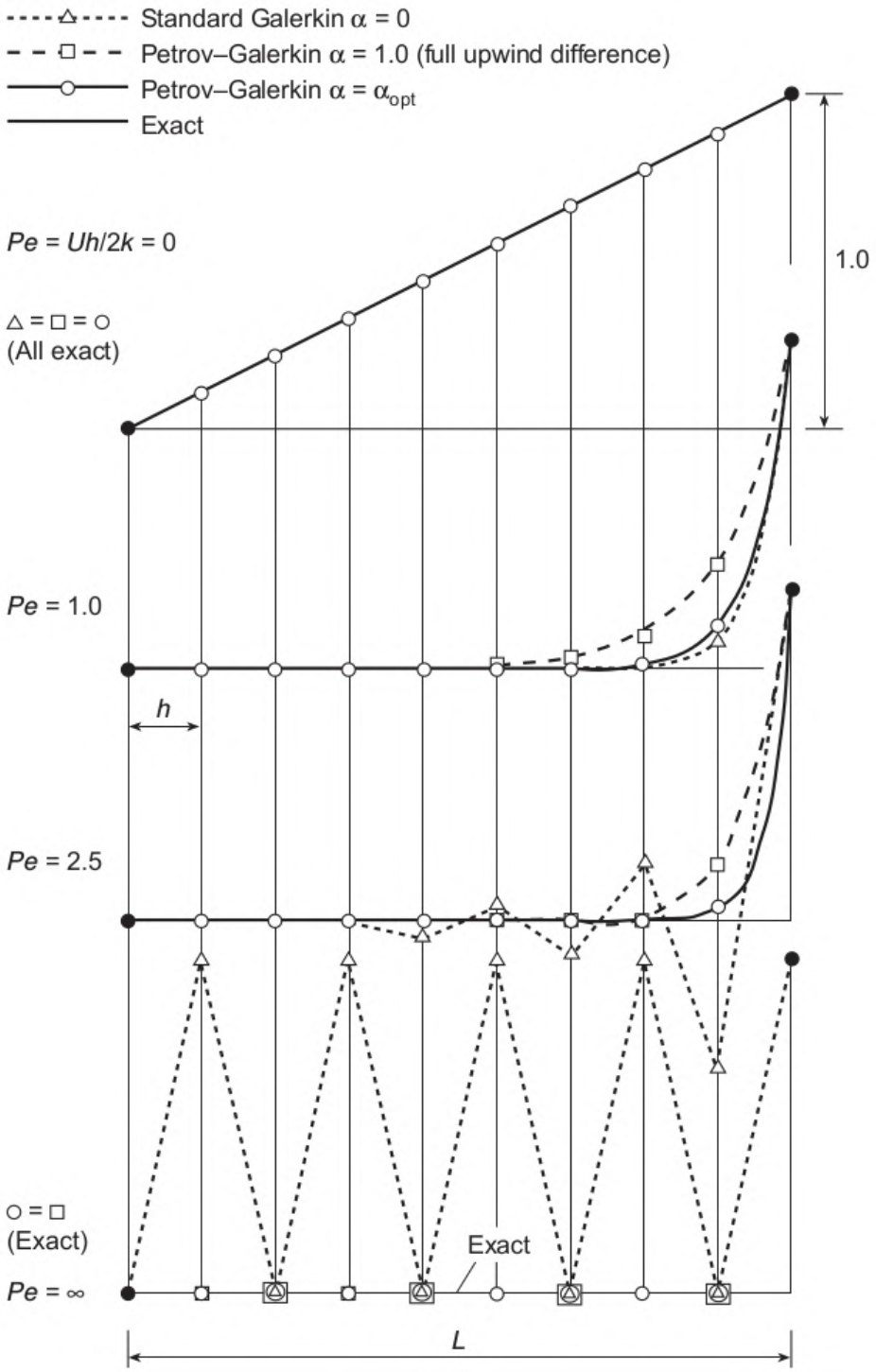


Figure 3.4: comparison of the Petrov-Galerkin method for different values of α and Pe [Zienkiewicz *et al.*, 2005, p.35]

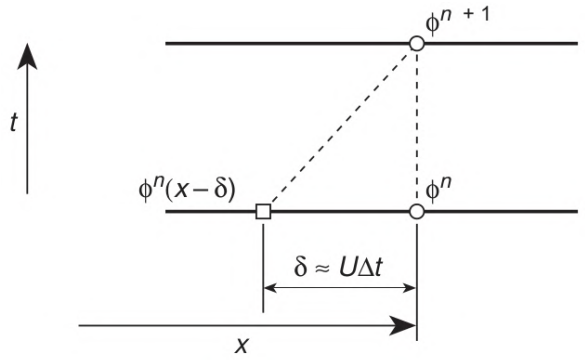


Figure 3.5: discretization along the characteristics [Zienkiewicz *et al.*, 2005, p.56]

where α is the upwind parameter, $\frac{\boldsymbol{U}}{|\boldsymbol{U}|}$ gives the normalized flow direction, $\frac{h}{2}$ scales the term to the element size and ∇N_i adds the asymmetry. Writing out the terms, the Streamline-Upwind Petrov-Galerkin weighting function is

$$\hat{w}_i = N_i + \frac{\alpha h}{2} \frac{U_1(\partial N_i/\partial x_1) + U_2(\partial N_i/\partial x_2)}{|\mathbf{U}|}.$$

Requiring the inner product with the discretized residual $\mathcal{R}(\hat{\phi})$ to vanish, the weak form of SUPG method becomes:

$$\int_\Omega \hat{w}\, \mathcal{R}(\hat{\phi})\, d\Omega = \int_\Omega \left(N_i + \alpha \frac{h}{2} \frac{\boldsymbol{U}}{|\boldsymbol{U}|} \cdot \nabla N_i \right) \mathcal{R}(\hat{\phi})\, d\Omega = 0$$

Note: a similar form is obtained by the Galerkin least squares (GLS) method in which the additional term of the weighting function is equal to the discretized differential operator of the original equation. For more information on this method see Donea and Huerta [2003].

3.2.4 The characteristic Galerkin scheme

One drawback of methods based on adding balancing diffusion to stabilize the Galerkin finite element solution is that the upwind parameter α has to be determined. As this is not trivial for complicated problems, Löhner *et al.* [1984] introduced a different approach for time dependent problems that adds automatically balancing diffusion without the need of determining an additional parameter. The basic idea is to introduce a change of variables from the original independent variable x to the moving coordinates x' with the relation

$$dx' = dx - U dt \tag{3.13}$$

(which is also called the characteristic equation) such that

$$\frac{dx'}{dt} = -U \qquad \text{and} \qquad \frac{dx'}{dx} = 1 \tag{3.14}$$

because dx and dt are independent. Substitution of the moving coordinate system into the time dependent convection-diffusion equation

$$\frac{\partial\phi(x,t)}{\partial t} + U\frac{\partial\phi(x,t)}{\partial x} - \frac{\partial}{\partial x}\Big(k\frac{\partial\phi(x,t)}{\partial x}\Big) + Q(x,t) = 0$$

yields:

$$\frac{\partial\phi(x',t)}{\partial t} + U\frac{\partial\phi(x',t)}{\partial x} - \frac{\partial}{\partial x}\Big(k\frac{\partial\phi(x',t)}{\partial x}\Big) + Q(x',t) = 0. \tag{3.15}$$

As those partial derivatives are not known directly, the chain rule has to be applied:

$$\begin{aligned}
\frac{\partial\phi(x',t)}{\partial t} &= \frac{\partial\phi(x',t)}{\partial x'}\overbrace{\frac{\partial x'}{\partial t}}^{-U} + \frac{\partial\phi(x',t)}{\partial t}\overbrace{\frac{\partial t}{\partial t}}^{1} = -U\frac{\partial\phi(x',t)}{\partial x'} + \frac{\partial\phi(x',t)}{\partial t} \\
\frac{\partial\phi(x',t)}{\partial x} &= \frac{\partial\phi(x',t)}{\partial x'}\overbrace{\frac{\partial x'}{\partial x}}^{1} + \frac{\partial\phi(x',t)}{\partial t}\overbrace{\frac{\partial t}{\partial x}}^{0} = \frac{\partial\phi(x',t)}{\partial x'}.
\end{aligned} \tag{3.16}$$

The diffusion term can be determined in a similar manner, i.e. by again applying the chain rule to equation 3.16. Consequently, the convective term of equation 3.15 cancels after substitution. In this way, the problem reduces to the well known time dependent diffusion equation:

$$\frac{\partial\phi(x',t)}{\partial t} - \frac{\partial}{\partial x'}\Big(k\frac{\partial\phi(x',t)}{\partial x'}\Big) + Q(x',t) = 0 \tag{3.17}$$

for which the standard Galerkin approximation produces optimal results (this is often referred to as the change from an Eulerian to a Lagrange perspective). As the deformations in flow problems are generally expected to be very large, a mesh updating procedure will produce huge element distortions and thus lead to serious difficulties. Instead, the characteristic Galerkin procedure now introduces a time discretisation of equation 3.17 (as seen in figure 3.5):

$$\begin{aligned}
\frac{1}{\Delta t}(\phi^{n+1}\big|_x - \phi^n\big|_{x-\delta}) \approx\ & \theta\left[\frac{\partial}{\partial x}\Big(k\frac{\partial\phi}{\partial x}\Big) - Q\right]^{n+1}\Bigg|_x \\
& + (1-\theta)\left[\frac{\partial}{\partial x}\Big(k\frac{\partial\phi}{\partial x}\Big) - Q\right]^{n}\Bigg|_{x-\delta}
\end{aligned} \tag{3.18}$$

which is explicit for $\theta = 0$ and implicit for $\theta = 1$. Equation 3.18 computes ϕ^{n+1} by evaluating ϕ at an yet unknown upstream position $x - \delta$. This upstream evaluation is done by calculating the distance δ travelled during the current time step and using a Taylor expansion to approximate the value of $\phi^n|_{x-\delta}$.

Starting with δ, an exact value can be computed by taking

$$\delta = \bar{U}\Delta t$$

with $\bar{U}$ being the average velocity during the current time step. However, $\bar{U}$ is generally unknown and has to be approximated, with different approaches leading to different stabilization terms. One possibility would be:

$$\bar{U} \approx \frac{1}{2}\Big(U^{n+1} + U^n\big|_{(x-\delta)}\Big),$$

where the yet unknown second term is again approximated by a Taylor expansion, neglecting higher order terms:

$$\begin{aligned}\bar{U} &\approx \frac{1}{2}\Big(U^{n+1} + \Big[U^n - \Delta t\, U^n \frac{\partial U^n}{\partial x} + O(\Delta t^2)\Big]\Big) \\ &\approx U^{n+1/2} - \frac{\Delta t}{2} U^n \frac{\partial U^n}{\partial x}.\end{aligned}$$

However, in the CBS algorithm presented in the next section the explicit approximation is used, in which case $\bar{U}$ is taken to be just U^n, such that:

$$\delta = U^n \Delta t.$$

Knowing the distance δ, the quantities of equation 3.18 evaluated at $x - \delta$ are determined by a Taylor expansion in space:

$$\phi^n\big|_{(x-\delta)} \approx \phi^n - \delta\frac{\partial \phi^n}{\partial x} + \frac{\delta^2}{2}\frac{\partial^2 \phi^n}{\partial x^2} + O(\Delta t^3)$$

$$\left[\frac{\partial}{\partial x}\Big(k\frac{\partial \phi}{\partial x}\Big) - Q\right]^n\Bigg|_{x-\delta} \approx \left[\frac{\partial}{\partial x}\Big(k\frac{\partial \phi}{\partial x}\Big) - Q\right]^n - \delta\frac{\partial}{\partial x}\left[\frac{\partial}{\partial x}\Big(k\frac{\partial \phi}{\partial x}\Big) - Q\right]^n + O(\Delta t^2)$$

With $\theta = \frac{1}{2}$, $\delta = U^n \Delta t$ and substituting the Taylor expansions into equation 3.18 the scheme can be written as:

$$\begin{aligned}\phi^{n+1} - \left[\phi^n - \Delta t\, U^n \frac{\partial \phi^n}{\partial x} + \Delta t^2 \frac{(U^n)^2}{2}\frac{\partial^2 \phi^n}{\partial x^2}\right] &= \frac{\Delta t}{2}\left[\frac{\partial}{\partial x}\Big(k\frac{\partial \phi^{n+1}}{\partial x}\Big) - Q^{n+1}\right] \\ &+ \frac{\Delta t}{2}\left[\frac{\partial}{\partial x}\Big(k\frac{\partial \phi^n}{\partial x}\Big) - Q^n\right] - \frac{\Delta t^2}{2} U^n \frac{\partial}{\partial x}\left[\frac{\partial}{\partial x}\Big(k\frac{\partial \phi^n}{\partial x}\Big) - Q^n\right]\end{aligned}$$

After approximating quantities evaluated at $t = n + 1$ by their values at $t = n$ and some reordering, the one dimensional fully explicit characteristic Galerkin scheme becomes:

$$\begin{aligned}\Delta\phi = \phi^{n+1} - \phi^n = &\overbrace{-\Delta t\left[U^n \frac{\partial \phi^n}{\partial x} - \frac{\partial}{\partial x}\Big(k\frac{\partial \phi^n}{\partial x}\Big) + Q^n\right]}^{\text{original differential equation}} \\ &\underbrace{+\frac{\Delta t^2}{2} U^n \frac{\partial}{\partial x}\left[U^n \frac{\partial \phi^n}{\partial x} - \frac{\partial}{\partial x}\Big(k\frac{\partial \phi^n}{\partial x}\Big) + Q^n\right]}_{\text{stabilizing term}}\end{aligned} \tag{3.19}$$

in which it can be seen that the amount of stabilization is dependent on Δt and U. However, this does not give the same result as taking $\theta = 0$ from the beginning of the derivation, as the stabilizing diffusion and source terms would not have a factor of $\frac{1}{2}$. Writing equation 3.19 in conservative form for multi-dimensions, omitting the n superscript and using a lower-case u (for later compatibility) yields:

$$\Delta\phi = -\Delta t\left[\frac{\partial (u_j \phi)}{\partial x_j} - \frac{\partial}{\partial x_i}\Big(k\frac{\partial \phi}{\partial x_i}\Big) + Q\right] + \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left[\frac{\partial (u_j \phi)}{\partial x_j} - \frac{\partial}{\partial x_i}\Big(k\frac{\partial \phi}{\partial x_i}\Big) + Q\right] \tag{3.20}$$

It is also worth noticing that this method was also influenced by the strongly related *Taylor-Galerkin method* presented in Donea [1984], which produces similar results and introduces even less numerical dissipation compared to the present method [Quecedo and Pastor, 2002].

3.3 The characteristic-based-split (CBS) algorithm

In this section the general idea of the characteristic-based-split scheme will be discussed, the direct application to the shallow water equations will follow in the next section.

Originally introduced by Zieniewicz and Codina [1995], the CBS algorithm makes an operator split to separate the pressure from the momentum equations. Thus the velocity increment is divided into a so called *intermediate momentum* increment ΔU_i^* and a pressure evaluation part ΔU_i^{**}:

$$\Delta U_i = \Delta U_i^* + \Delta U_i^{**} \tag{3.21}$$

There are two kind of splits available: In split A, the pressure is completely removed from the momentum equations, while in split B it is evaluated at time $t = n$ as a source type quantity. In both cases the momentum equations include a convective term and thus need special treatment. As the name of the CBS algorithm indicates, this is done by applying the characteristic Galerkin scheme (see section 3.2.4) in the form of equation 3.20. For a general system of conservation laws like the Navier-Stokes equations, the mass flow U is substituted for the primary variable ϕ. Additionally, third order terms from the stabilization part are neglected. After successful calculation of ΔU_i^*, the pressure increment is solved using the mass conservation equation as well as equation 3.21. In the third step, the final increment of U is achieved by computing ΔU_i^{**} and adding it to ΔU_i^*.

3.4 Adaption of the CBS scheme to the shallow water equations

For solving the shallow water equations with the CBS scheme, as introduced in Zienkiewicz and Ortiz [1995], the equivalent pressure term shown in equation 2.22 is used and the viscous terms are neglected. Thus omitting the overbar for depth averaged quantities the shallow water equations in their conservative form are:

$$\frac{1}{c^2}\frac{\partial p}{\partial t} + \frac{\partial U_i}{\partial x_i} = 0 \tag{3.22}$$

$$\frac{\partial U_i}{\partial t} + \frac{\partial (u_j U_i)}{\partial x_j} + \frac{\partial p}{\partial x_i} + Q = 0$$

with

$$p = \frac{1}{2} g (h^2 - H^2) \qquad \text{and} \qquad c = \sqrt{gh}$$

Noting that

$$\frac{1}{c^2}\frac{\partial p}{\partial t} = \frac{1}{c^2}\frac{g}{2}\frac{\partial}{\partial t}(h^2 - H^2) = \frac{1}{gh}\frac{g}{2}\frac{\partial (h^2)}{\partial t} = \frac{1}{2h} 2h \frac{\partial h}{\partial t} = \frac{\partial h}{\partial t}.$$

As mentioned in section 3.3, the solution of the momentum equations by the CBS scheme involves evaluation of the pressure as a *source type* quantity in the first step. For clarity a new auxiliary source term Q' is introduced at this stage:

$$Q' = \frac{\partial p}{\partial x_i} + Q.$$

Thus, the intermediate momentum equations become:

$$\frac{\partial U_i}{\partial t} + \frac{\partial (u_j U_i)}{\partial x_j} + Q' = 0.$$

Now, the characteristic Galerkin scheme of equation 3.20 can be applied, leading to the following formulation (noting that the shallow water equations in this form do not contain diffusion terms):

$$\begin{aligned}\Delta U_i &= -\Delta t\left[\frac{\partial (u_j U_i)}{\partial x_j} + Q'\right] + \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left[\frac{\partial (u_j U_i)}{\partial x_j} + Q'\right], \\ &= -\Delta t\left[\frac{\partial (u_j U_i)}{\partial x_j} + \frac{\partial p^{n+\theta_2}}{\partial x_i} + Q\right] + \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left[\frac{\partial (u_j U_i)}{\partial x_j} + \frac{\partial p^{n+\theta_2}}{\partial x_i} + Q\right],\end{aligned}$$

where the pressure p is evaluated at time $t = t^n + \theta_2 \Delta t$:

$$p^{n+\theta_2} = p^n + \theta_2 \Delta p,$$

and all other quantities are evaluated at t = t^n. Now following the procedure presented in Morandi-Cecchi and Venturin [2006] by using split A, the momentum equations become

$$\Delta U_i = \Delta U_i^* + \Delta U_i^{**} \tag{3.23}$$

with

$$\Delta U_i^* = -\Delta t\left[\frac{\partial (u_j U_i)}{\partial x_j} + Q\right] + \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left[\frac{\partial (u_j U_i)}{\partial x_j} + Q\right] \tag{3.24}$$

and

$$\Delta U_i^{**} = -\Delta t\left[\frac{\partial p^{n+\theta_2}}{\partial x_i}\right] + \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left[\frac{\partial p^{n+\theta_2}}{\partial x_i}\right] \tag{3.25}$$

evaluating the pressure at time $t = t^n + \theta_2 \Delta t$. After computing ΔU_i^* in the first CBS step from equation 3.24, the second step is to determine the pressure increment Δp by substituting

$$\frac{\partial U_i^{n+\theta_1}}{\partial x_i} \approx \frac{\partial U_i}{\partial x_i} + \theta_1 \frac{\partial (\Delta U_i)}{\partial x_i} \approx \frac{\partial}{\partial x_i}(U_i + \theta_1 \Delta U_i^*) - \Delta t\, \theta_1 \frac{\partial^2 p^{n+\theta_2}}{\partial x_i^2} \tag{3.26}$$

(from equations 3.23 and 3.25 as well as neglecting third order terms) into the mass conservation equation (3.22):

$$\left(\frac{1}{c^2}\right)\frac{\Delta p}{\Delta t} + \frac{\partial}{\partial x_i}(U^n + \theta_1 \Delta U_i^*) - \Delta t \theta_1 \left(\frac{\partial^2 p}{\partial x_i^2} + \theta_2 \frac{\partial^2 (\Delta p)}{\partial x_i^2}\right) = 0.$$

After rearranging terms, the second CBS step becomes:

$$\left(\frac{1}{c^2}\right)\Delta p - \Delta t^2\theta_1\theta_2\frac{\partial^2(\Delta p)}{\partial x_i^2} = -\Delta t\frac{\partial}{\partial x_i}(U^n + \theta_1\Delta U_i^*) + \Delta t^2\theta_1\,\frac{\partial^2 p}{\partial x_i^2} \tag{3.27}$$

In the third and last step, using equation 3.23 and 3.25 (this time including all terms, but evaluating the stabilizing part at $t = t^n$), the increment in U can be computed:

$$\Delta U_i = U_i^* - \Delta t\left[\frac{\partial p^{n+\theta_2}}{\partial x_i}\right] + \frac{\Delta t^2}{2}u_k\frac{\partial}{\partial x_k}\left[\frac{\partial p}{\partial x_i}\right]. \tag{3.28}$$

Now the three CBS steps are transformed into their weak form by multiplying with a test function and integrating over the domain. Spatial discretization by a standard Galerkin procedure with

$$U \approx \mathbf{N}\tilde{\mathbf{U}}, \qquad u \approx \mathbf{N}\tilde{\mathbf{u}}, \qquad p \approx \mathbf{N}\tilde{\mathbf{p}}$$

leads to the following matrix form of the CBS algorithm for the shallow water equations (with discretization matrices below):
CBS Step 1: Intermediate momentum equations (from equation 3.24):

$$\mathbf{M}\Delta\tilde{\mathbf{U}}_i^* = -\Delta t(\mathbf{C}_u\tilde{\mathbf{U}}_i^n + \mathbf{M}\tilde{\mathbf{Q}}_i^n) - \frac{\Delta t^2}{2}(\mathbf{K}_u\tilde{\mathbf{U}}_i^n + \mathbf{f}_{Q_i}) \tag{3.29}$$

CBS Step 2: Pressure equation (from equation 3.27):

$$(\mathbf{M}_c + \Delta t^2\theta_1\theta_2\mathbf{H})\Delta\tilde{\mathbf{p}} = \Delta t\mathbf{G}_i(\tilde{\mathbf{U}}_i^n + \Delta\tilde{\mathbf{U}}_i^*) - \Delta t^2\theta_1\mathbf{H}\tilde{\mathbf{p}}^n - \Delta t\,\mathbf{f}_p \tag{3.30}$$

CBS Step 3: Momentum equations (from equation 3.28):

$$\mathbf{M}\Delta\tilde{\mathbf{U}}_i = \mathbf{M}\Delta\tilde{\mathbf{U}}_i^* - \Delta t\mathbf{G}_i^T(\tilde{\mathbf{p}}^n + \theta_2\Delta\tilde{\mathbf{p}}) - \frac{\Delta t^2}{2}\mathbf{P}_{u,i}\tilde{\mathbf{p}}^n \tag{3.31}$$

where u and c subscript are indicating dependencies on the solutions fields and thus require to be computed at each time step. To assign the matrix names some functionality, $\mathbf{H}$ can be seen as diffusion/Laplace operator, $\mathbf{K}_u$ as stabilizing diffusion operator, $\mathbf{C}$ as convection operator, $\mathbf{G}$ as divergence operator, $\mathbf{P}$ together with $\mathbf{G}^T$ as gradient operators and $\mathbf{M}$ as mass matrix. Note also that the switched signs in equations 3.29 to 3.31 compared to the continuous formulation originate from an integration by parts. The boundary terms arising from this were neglected by Morandi-Cecchi and Venturin [2006] and their justification it is not entirely clear yet. As mentioned in Zienkiewicz *et al.* [2005], boundary terms from stabilizing diffusion parts are neglected, but nevertheless there would be additional contributions originating from, for example, the convective part in step 1.

The discretization matrices are defined as:

$$\mathbf{M} = \int_\Omega \mathbf{N}^T\mathbf{N}d\Omega \qquad \mathbf{H} = \int_\Omega \frac{\partial\mathbf{N}^T}{\partial x_j}\frac{\partial\mathbf{N}}{\partial x_j}d\Omega$$

$$\mathbf{M}_c = \int_\Omega \mathbf{N}^T\left(\frac{1}{c^2}\right)^n\mathbf{N}d\Omega \qquad \mathbf{G}_i = \int_\Omega \frac{\partial\mathbf{N}^T}{\partial x_j}\mathbf{N}d\Omega$$

$$\mathbf{C}_u = \int_\Omega \mathbf{N}^T \frac{\partial}{\partial x_j}(\mathbf{u}_j \mathbf{N}) d\Omega = \int_\Omega \mathbf{N}^T \left[\frac{\partial \mathbf{u}_j}{\partial x_j}\mathbf{N} + \frac{\partial \mathbf{N}}{\partial x_j}\mathbf{u}_j \right] d\Omega$$
$$= \int_\Omega \mathbf{N}^T \left[\left(\frac{\partial \mathbf{N}}{\partial x_j}\tilde{\mathbf{u}}_j \right)\mathbf{N} + \frac{\partial \mathbf{N}}{\partial x_j}\left(\mathbf{N}\tilde{\mathbf{u}}_j\right) \right] d\Omega$$

$$\mathbf{K}_u = \int_\Omega \frac{\partial}{\partial x_k}(\mathbf{u}_k \mathbf{N}^T) \frac{\partial}{\partial x_j}(\mathbf{u}_j \mathbf{N}) d\Omega = \int_\Omega \left[\frac{\partial \mathbf{u}_k}{\partial x_k}\mathbf{N}^T + \frac{\partial \mathbf{N}^T}{\partial x_k}\mathbf{u}_k \right] \cdot \left[\frac{\partial \mathbf{u}_j}{\partial x_j}\mathbf{N} + \frac{\partial \mathbf{N}}{\partial x_j}\mathbf{u}_j \right] d\Omega$$
$$= \int_\Omega \left[\left(\frac{\partial \mathbf{N}}{\partial x_k}\tilde{\mathbf{u}}_k \right)\mathbf{N}^T + \frac{\partial \mathbf{N}^T}{\partial x_k}\left(\mathbf{N}\tilde{\mathbf{u}}_k\right) \right] \cdot \left[\left(\frac{\partial \mathbf{N}}{\partial x_j}\tilde{\mathbf{u}}_j \right)\mathbf{N} + \frac{\partial \mathbf{N}}{\partial x_j}\left(\mathbf{N}\tilde{\mathbf{u}}_j\right) \right] d\Omega$$

$$\mathbf{P}_{u,i} = \int_\Omega \frac{\partial}{\partial x_k}(\mathbf{u}_k \mathbf{N}^T) \frac{\partial \mathbf{N}}{\partial x_i} d\Omega = \int_\Omega \left[\left(\frac{\partial \mathbf{N}}{\partial x_k}\tilde{\mathbf{u}}_k \right)\mathbf{N}^T + \frac{\partial \mathbf{N}^T}{\partial x_k}\left(\mathbf{N}\tilde{\mathbf{u}}_k\right) \right] \frac{\partial \mathbf{N}}{\partial x_i} d\Omega$$

$$\mathbf{f}_{Q_i} = \int_\Omega \frac{\partial}{\partial x_k}(\mathbf{u}_k \mathbf{N}^T)\mathbf{N} d\Omega \cdot \tilde{\mathbf{Q}}_i = \int_\Omega \left[\left(\frac{\partial \mathbf{N}}{\partial x_k}\tilde{\mathbf{u}}_k \right)\mathbf{N}^T + \frac{\partial \mathbf{N}^T}{\partial x_k}\left(\mathbf{N}\tilde{\mathbf{u}}_k\right) \right] \mathbf{N}\, d\Omega \cdot \tilde{\mathbf{Q}}_i$$

Note that especially for $\mathbf{K}_u$ a time consuming nested sum over k and j has to be implemented. The factor $\frac{1}{c^2}$ in $\mathbf{M}_c$ can be taken as an average over one element but for better accuracy especially with higher order ansatz spaces it is evaluated at every integration point. The source term was factored out for pre-computed values but generally for arbitrary source terms a projection onto the ansatz space has to be done. The boundary term $\mathbf{f}_p$ defined as (overbar indicating prescribed boundary values)

$$\mathbf{f}_p = \int_\Gamma \mathbf{N}^T n_i \bar{U}_i \, d\Gamma \tag{3.32}$$

can be used to impose in- or outflow boundary conditions in a weak manner.

Once the three steps have been computed it is possible to recover the actual water height h from the equivalent pressure term by the following relation (see equation 2.22):

$$h = \sqrt{H^2 + \frac{2p}{g}} \tag{3.33}$$

A problem that is encountered at this stage is, that recovering the height from the pressure term as well as computing the velocities from the mass flow by

$$u = \frac{U}{h} \tag{3.34}$$

are nonlinear operations and thus can not be represented by the same ansatz space. A more detailed discussion of this problem is given in section 4.2.

Chapter 4

Implementation of the characteristic-based-split algorithm

The implementation of the CBS algorithm was done in three steps. To gain basic experience with finite elements and in particular the presented algorithm, a 1-d version was written in MATLAB®. As this gave reasonable results, a new 2-d version of the code was then implemented in the same environment with the restriction of low order quadrilaterals. Here the behavior of the CBS algorithm in solving the full shallow water equations was analyzed including source terms and boundary conditions. With this knowledge the high-order finite element code *AdhoC++* developed at the chair for *Computation in Engineering* was extended to provide the CBS methodology for solving the shallow water equations. To give some information related to the implementation, the framework is shortly introduced in the following section.

4.1 The high order finite element framework AdhoC++

AdhoC++ is a framework written in C++ and primarily designed for solving problems of elasticity and heat conduction. Being a research code, the first goal is not to optimize the performance rather than providing a versatile structure allowing extensions without much effort. In this section, first a short description of the code, together with the tasks that needed to be done to implement the CBS algorithm, is given. Subsequently a more detailed discussion on the high-order part of the framework follows.

4.1.1 Organization of the code

As shown in figure 4.1, the basic structure of the code involves a separation of physical description and discretization. Consequently, there was no need to change anything related to the discretization during the implementation of the CBS algorithm, which of course includes high order ansatz spaces. Furthermore, a variety of post-processing features already existed such that the only remaining task was to add new math models to compute the system matrices from section 3.4 and to create a new problem class that implements the CBS procedure. However, it was necessary to introduce a second computational mesh for

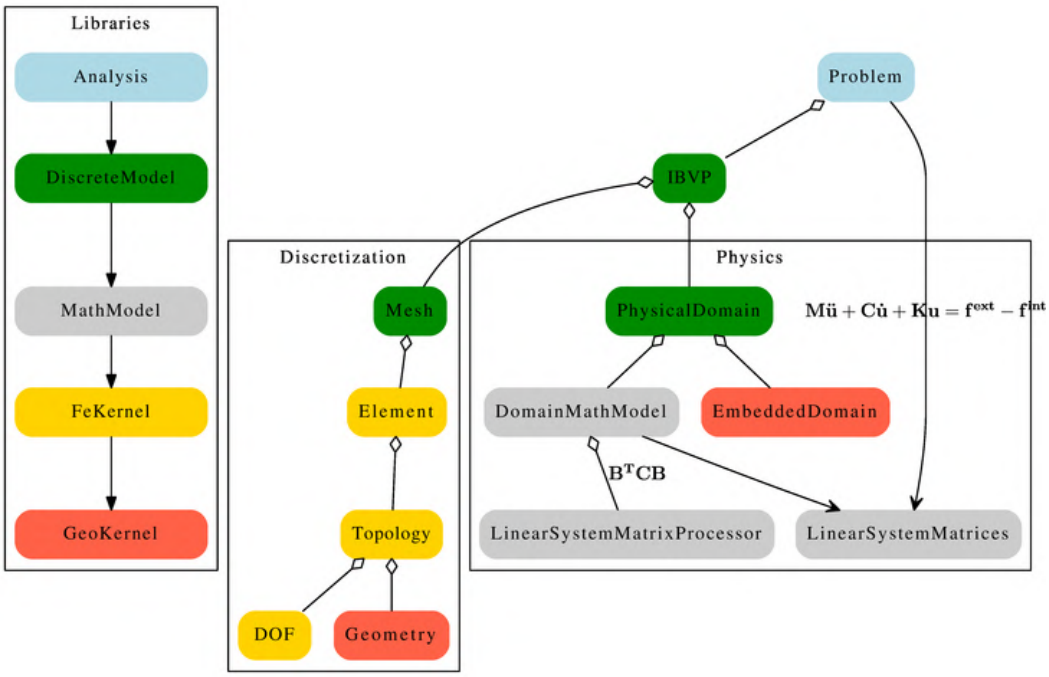


Figure 4.1: Key classes of the AdhoC++ framework [Redmine page of AdhoC++, Chair for Computation in Engineering, 2014]

the pressure/height field, because the problems the code was originally designed for involve only solution fields with equal treatment for each field component. Although this represents an additional memory consumption, the overall performance is not affected, as pressure and velocities are computed separately.

4.1.2 Integrated Legendre polynomials as a finite element basis

As already mentioned, the integrated Legendre polynomials as presented in Babŭska and Szymczak [1981] offer some remarkable features that makes them a favorable choice for a finite element basis. Their derivative, the Legendre polynomials, given as

$$L_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n, \quad x \in (-1,1), \quad n = 0,1,2,\ldots$$

are orthogonal in $[-1,1]$. Consequently, stiffness matrices arising from a integrated Legendre basis contain a lot of zeros [Düster, 2008]. Additionally, the condition number of the stiffness matrix improves drastically compared to the Lagrange basis functions, as figure 4.2 shows.

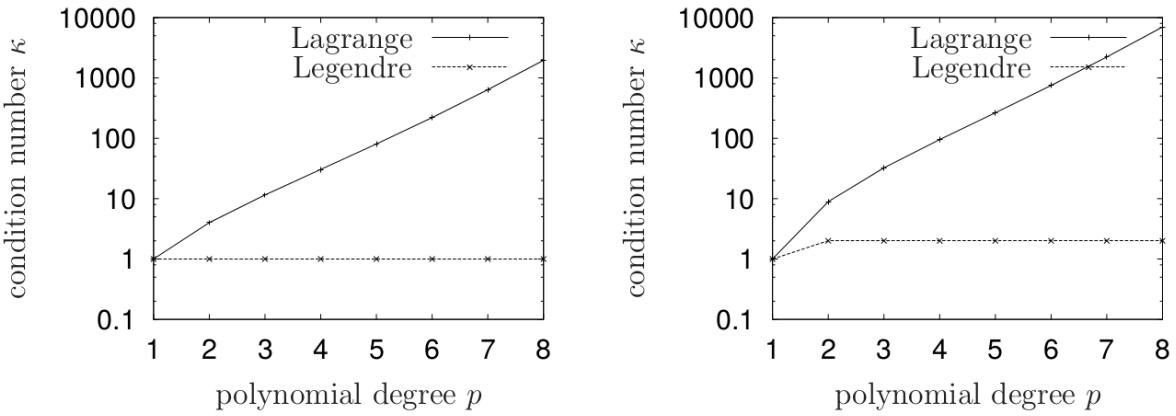


Figure 4.2: Condition number of the local (left-hand side) and global (right-hand side) stiffness matrix [Düster, 2008]

The hierarchic shape functions are given by

$$N_1(\xi) = \frac{1}{2}(1-\xi)$$
$$N_2(\xi) = \frac{1}{2}(1+\xi)$$
$$N_i(\xi) = \phi_{i-1}$$
$$\text{with } \phi_j = \frac{1}{\sqrt{2(2j-1)}}\big(L_j(\xi) - L_{j-2}(\xi)\big)$$

For more information see [Szabó and Babuška, 1991]. The two-dimensional basis can now be created by taking the tensor-product of the one-dimensional basis. This means basically that every combination of the 1-d shape-functions serves now as a 2-d basis-function.

4.2 Non-linear operations on solution fields

During the solution process it is necessary to perform calculations other than addition and scalar multiplication with a result that cannot be represented by the same finite element ansatz-space. Examples are interchanging velocity and mass flow vectors (see also equation 3.34) as well as transforming between alternative pressure and water height. In this work, two possible treatments are discussed: operating directly on the degree of freedom vector and performing a least-squares projection. The latter is also important e.g. if initial conditions are not discretized in terms of the finite element subspace and will therefore be discussed in some detail.

Additionally, the two methods are compared by considering a simple example for one element with linear shape functions

$$\mathbf{N}(x) = \frac{1}{2}\begin{pmatrix}1-x, & 1+x\end{pmatrix} \qquad \text{on } \Omega = [-1,1]$$

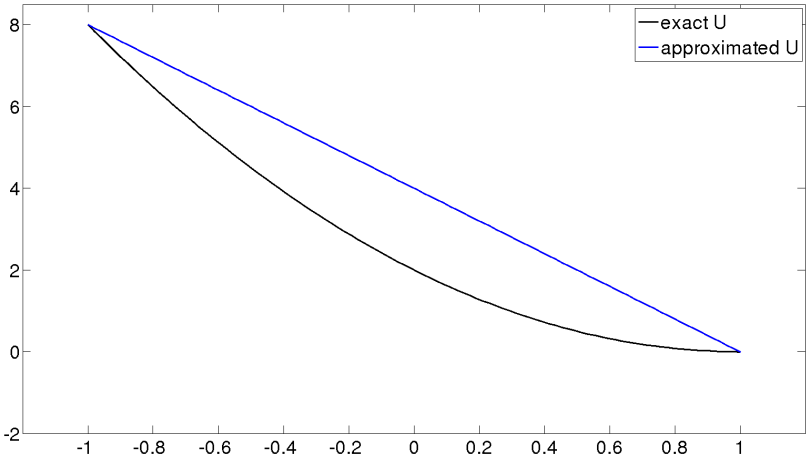


Figure 4.3: Product of linear shape functions approximated by multiplying the coefficients

and the multiplication operation $U = u \cdot h$. The discretized solution fields are chosen as:

$$\hat{u} = \mathbf{N}\tilde{\mathbf{u}} = \begin{pmatrix} N_1 & N_2 \end{pmatrix} \begin{pmatrix} 4 \\ 0 \end{pmatrix} = 2(1-x) \qquad \text{and} \qquad \hat{h} = \mathbf{N}\tilde{\mathbf{h}} = \begin{pmatrix} N_1 & N_2 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = 1 - x \tag{4.1}$$

with the exact solution

$$U_{\text{ex}} = \hat{u} \cdot \hat{h} = 2(1-x)^2 \qquad \text{and} \qquad \int_{-1}^{1} U_{\text{ex}}\, dx = \frac{16}{3}.$$

4.2.1 Direct manipulation of the coefficient vector

One possibility is to perform those non-linear operations directly on the degree of freedom vectors. However, the deviation in the integrals of approximation and exact solution might be very large and the resulting mass (or momentum) loss is definitely not desirable. The application to the example discretization of equation 4.1 can be seen in figure 4.3. The integral of the approximation

$$\int_{-1}^{1} \hat{U}\, dx = \int_{-1}^{1} 4(1-x)\, dx = 8$$

introduces an error of 50% compared to the exact integral. Moreover, for high order shape functions, nonlinear operations on the degree of freedom vector might produce completely wrong results. A problem encountered if integrated Legendre polynomials are used is that for constant functions the coefficients of internal modes are equal to zero. Dividing by that degree of freedom vector will then result in a division by zero. Consequently this procedure, if any, should be used only for linear or Lagrange shape functions, although being very efficient.

4.2.2 Least squares projection

Better, but more expensive, is a least squares projection onto the ansatz space, which produces an orthogonal and thus minimal error with respect to the L_2 norm. The projection for a

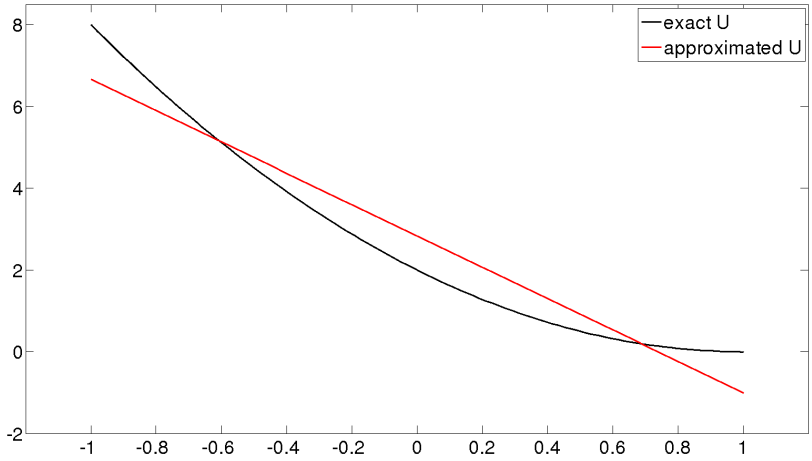


Figure 4.4: Product of linear shape functions approximated by least squares projection

general function f is stated as follows:

$$\begin{aligned} \text{find } \hat{f} \in W \text{ such that} \qquad & (v, \hat{f}) = (v, f) \qquad \forall v \in V \\ \Leftrightarrow\ & (v, f - \hat{f}) = 0 \qquad \forall v \in V, \end{aligned} \tag{4.2}$$

with $\hat{f}$ being the projection of f and (a, b) denoting the L_2 inner product, defined in 1.6. Now if V is taken to be the same as W, the error $f - \hat{f}$ is orthogonal to the space W. With the discretization $\hat{f} = \mathbf{N}\tilde{\mathbf{f}}$, the least squares projection becomes:

$$\mathbf{M}\tilde{\mathbf{f}} = \mathbf{b}$$

$$\text{where } \mathbf{M} = \int_\Omega \mathbf{N}^T\mathbf{N}\, d\Omega \qquad \text{and} \qquad \mathbf{b} = \int_\Omega \mathbf{N}^T \cdot f\, d\Omega$$

For example, in recovering the height from the alternative pressure by equation 3.33, f would be:

$$f = h(\hat{\mathbf{H}}, \hat{\mathbf{p}}) = \sqrt{(\mathbf{N}\hat{\mathbf{H}})^2 + \frac{2}{g}\mathbf{N}\hat{\mathbf{p}}}.$$

Another benefit of least squares projections is that the integrals of the function f and its projection $\hat{f}$ over the domain are equal. As the fluid dynamics equations represent a system of conservation laws this is a very important property. By substituting $v = 1$ into equation 4.2, it follows that

$$(1, f - \hat{f}) = 0$$

and thus, by using (1.6)

$$\int_\Omega \hat{f}\, d\Omega = \int_\Omega f\, d\Omega.$$

Of course this relation is only valid if $1 \in V$, which is, however, typically true. The result of the least squares projection $\hat{f} = U_{\text{lq}} = 8/3 - 4x$ of the example given in equation 4.1 is shown in figure 4.4.

Although this procedure might have a strong influence on the performance of the code, storing the factorized mass matrix will avoid too much impact. Moreover, during the computation of the right hand side on an element integration level, the shape functions $\mathbf{N}$ have to be

computed only once and thus the projection will not take too much time compared to the computation of the discretization matrices of the CBS procedure.

However, a problem where operating directly on the coefficients might perform better is the approximation of discontinuities, such as the initial dam break condition. As can be seen in the projection of the initial water surface in figure 5.7 this can lead to some oscillations.

4.3 The ground slope source term

As the slope of the bottom level contributes to the momentum equations it is important to discuss the best way of approximating those derivatives. Supposing that the bottom topography is discretized in terms of the finite element ansatz space as

$$H = \boldsymbol{N}\tilde{\boldsymbol{H}}, \tag{4.3}$$

the straight forward and currently implemented way to proceed is to evaluate the derivative by taking

$$\frac{\partial H}{\partial x_i} = \frac{\partial \boldsymbol{N}}{\partial x_j}\tilde{\boldsymbol{H}}. \tag{4.4}$$

If the original data was given as an arbitrary function it must be discretized for instance by one of the methods presented in 4.2.

However, this way of computing the ground slope contribution might not be optimal with respect to the three physical conditions of:

- flat water surface for quiescent flow over an irregular bottom
- force conservation during hydraulic jumps
- equal friction and bed slope for uniform flow.

It has yet to be tested how the ‘standard‘ method behaves and if nessesary an alternative way to evaluate the bottom source term has to be found.

One possibility, introduced by Valiani and Begnudelli [2006], uses an alternative formulation:

$$-gh\frac{\partial H}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{1}{2}gh^2\right)\Big|_{\eta=\eta_*}$$

where $|_{\eta=\eta_*}$ means evaluation for a constant value for η, that is chosen as the element average water surface level. It can be shown, that this method is indeed optimal with respect to the above criteria. Consequently, for a standard finite element discretization, η_* can be computed as:

$$\eta_* = \frac{1}{A_e}\int_{\Omega_e} \hat{\eta}\, d\Omega_e = \frac{\displaystyle\int_{\Omega_e} \mathbf{N}\cdot\tilde{\boldsymbol{\eta}}\, d\Omega_e}{\displaystyle\int_{\Omega_e} 1\, d\Omega_e}$$

However, as the CBS scheme uses the alternative pressure (see section 2.4), the bottom slope source term is formulated slightly different and it has yet to be investigated how or if the divergence form can be applied in this case.

4.4 Implementation of boundary conditions

As mentioned in chapter 3, equation 3.32 can be used to impose flow boundary conditions in a weak manner. In addition velocities need to be prescribed in the correction step of the CBS algorithm [Ortiz *et al.*, 2004]. For inflow boundaries it is also necessary to specify either the tangential flow or the water height (for sub-critical flow), as table 2.2 shows.

The water surface can be set by simply constraining the equation of the second CBS step. For this purpose common techniques can be used, such as the penalty method, Nitsche's method or just choosing the trial functions to satisfy the boundary values.

In order to simulate river flows, it is important to be able to model the outflow boundary. A first approach would be to set equal values for in- and outflow, but this does certainly not represent the reality, as the appearance of water level fluctuations will always lead to slightly different discharge values. In numerical river hydraulics, this problem is commonly solved by setting outflow values equal to the prediction of some empiric flow formula. It is important that the empiric law, used for predicting outflow velocities, coincides with the bottom friction model. If this is the case, in a steady state flow, where bottom friction and gravity are balanced, the water level will then be linearly in the boundary region.

4.5 The dry-wet problem

One of the most challenging tasks in implementing the shallow water equations is definitely handling drying and wetting areas.

A procedure recommended in Zienkiewicz *et al.* [2005] is to distort the boundary elements according to the movement of the fluid, as presented in Lynch and Gray [1980]. However, if the differences in the surface elevation become big, this produces huge deformations and requires remeshing. Additionally, the implementation of such remeshing procedures is generally not trivial and should be avoided if possible.

A better alternative is to includes dry areas in the solution process. According to Medeiros and Hagen [2006], there are 4 basic categories of algorithms following this approach by:

- setting a thin film on the whole domain
- removing (deactivating) dry elements
- extrapolating the height
- allowing a negative depth

The last category can be excluded because for methods using an equivalent pressure, like the CBS algorithm, the water height h is not allowed to be negative (Quecedo and Pastor [2002]). This follows from the fact that if values of the equivalent pressure p are such that h would be negative, the square root in equation 3.33 will contain negative values and thus produce complex solutions. Additionally, h can not be equal to zero as equation 3.34 includes division by h.

The first approach of setting h to a small value on the total domain gives reasonable results for very small gradients of the water surface and would probably also work for some steady state solutions. However, high accelerations due to large gradients of η led to oscillations in

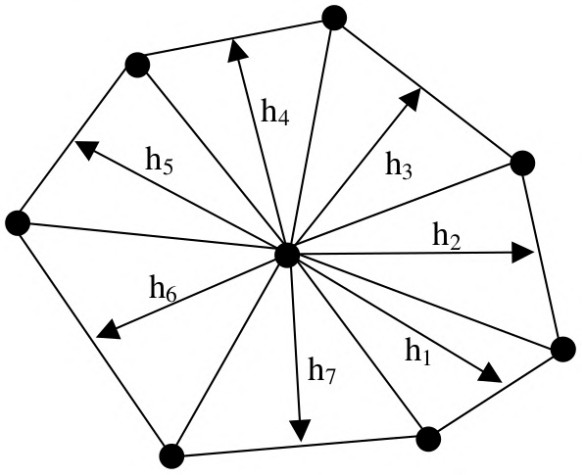


Figure 4.5: Standard element size computation for node i by orthogonal projection [Thomas and Nithiarasu, 2004]

transition area between wet and dry regions. In combination with the small water height on those dry regions the solution became negative resulting in complex values for h as described above.

The remaining two approaches have not been tested yet and thus their compatibility with the CBS algorithm can not be judged.

4.6 Estimation of the critical time step

As mentioned in section 3.2.4, the derivation of the characteristic Galerkin method includes an explicit time discretization. As this scheme is used by the CBS algorithm to solve the intermediate momentum equations, the time step length can not be chosen arbitrary. However, it is possible to formulate a critical time step condition only in terms of the flow velocity and not dependent on the wave celerity [Zienkiewicz *et al.*, 2005]:

$$\Delta t \leq \frac{d}{|\mathbf{u}|},$$

with d being the element size. The independence of the wave celerity is a big advantage when problems with small Froude numbers (see equation 2.23) are computed. Note: At this stage the notation in the literature differs as sometimes U (which corresponds to the mass flow) is used instead of u. However, there is no justification for this choice and thus the velocity u was taken.

Clearly, to calculate the critical time step the element size has to be determined first. While this is easy for 1-d elements it gets more complicated in 2-d. Generally there is not a single algorithm for estimating the element size, and different strategies are used for different purposes and elements shapes. A a simple way to achieve good results for triangular elements is to calculate the relevant size for some node i as the minimum of all surrounding elements lengths [Thomas and Nithiarasu, 2004]. As shown in figure 4.5, the length is taken to be the distance from node i to its orthogonal projection onto the opposite edge. A similar strategy

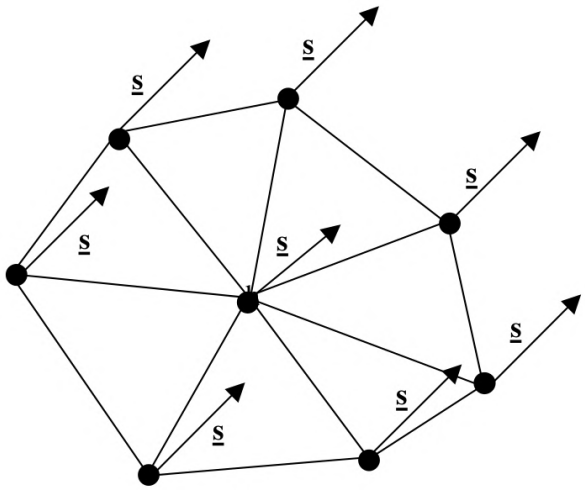


Figure 4.6: Streamline element size computation for node i [Thomas and Nithiarasu, 2004]

could be used for quadrilaterals by considering the projections on two sides. However, it was also shown in Thomas and Nithiarasu [2004] that an improved accuracy can be achieved if the element size is computed in the streamline direction (see figure 4.6). A further advantage of this procedure is that it represents a general methodology, independent of the element shape. On the other hand, it is not possible to cache the element size as the velocity directions typically change during the simulation and thus the computation has to be done for every time step.

4.7 Shock capturing

As pointed out in chapter 2, the compressible behavior of the shallow water equations can lead to shock waves. Unfortunately, this generally causes problems in numerical solutions due to oscillations around the discontinuity. This type of oscillation can arise even if the most accurate stabilizing diffusion method is applied [Zienkiewicz *et al.*, 2005]. As the dam break example, presented in section 5.1 (see figure 5.2), shows, the CBS algorithm produces reasonable results, but nevertheless for some extreme situations a shock capturing strategy has to be applied. Although normally some minor oscillations are acceptable, for the CBS algorithm special care has to be taken, because due to the alternative pressure term negative values for h are not allowed.

According to Donea and Huerta [2003], shock capturing techniques can be grouped in *artificial diffusion* and *high resolution* methods. While the last group is generally more accurate for complex flow situations, artificial diffusion methods convince through their simplicity concerning theory and implementation. Therefore, in this work only the most popular method from the first group, presented by Lapidus (1967), is presented. The simple idea is to add diffusion proportional to the gradient of the considered quantity and the element size. For the scalar convection-diffusion equation (see equation 3.3) the resulting additional diffusion term would be:

$$\tilde{k} = C_{\mathrm{Lap}}\, h^2 \left| \frac{\partial \phi}{\partial x} \right|$$

with $\tilde{k}$ being the additional diffusion, h the element size and C_{Lap} a coefficient to control the influence of the method. C_{Lap} is usually set to values between 0 and 2 [Donea and Huerta, 2003]. A multidimensional version shown in Zienkiewicz *et al.* [2005] includes a certain amount of anisotropy:

$$\tilde{k}_{ij} = C_{\text{Lap}}\, h^2 \frac{|V_i V_j|}{|\mathbf{V}|} \qquad \text{with} \qquad V_i = \frac{\partial \phi}{\partial x_i}$$

for which in the case of the shallow water equations ϕ is substituted by h. However, it is important to note that if a fine resolution is desired, alternatives have to be considered, because this method tends to behave too dissipative. In [Donea and Huerta, 2003, 176], a good overview of this field is given.

As the CBS algorithm performed well for most of the test cases presented in chapter 5 there was no need for implementing a shock capturing procedure until now.

Chapter 5

Results

In this chapter some of the results gained so far are presented. The test cases were computed exclusively with *AdhoC++* (see 4.1) including post-processing. The visualization of the height fields was done with *Paraview*, making use of the *Warp by Scalar* function.

5.1 The dam break model problem

One of the most often considered model problems is the 1-d dam break problem. The name comes from the fact that it models a dam that separates two different water levels and at $t = 0$ suddenly disappears, or 'breaks'. The following propagation of a shock often serves as a benchmark for the capability of a numerical scheme to handle such discontinuities. Additionally an exact solution was given by Stoker [1957]. Figure 5.1 shows the initial setup of the problem. For the simulation with the presented CBS algorithm the parameters were chosen as: $h_1 = 2$, $h_2 = 1$, $\theta_1 = 1$, $\theta_2 = 1$, $\Delta t = 0.002$ and a domain length of 5. The water height was split such that $H = -1$, $\eta_1 = 1$ and $\eta_2 = 0$. Figure 5.2 shows the result at $t = 0.2$ for different ansatz orders. The combination of order and number of elements was chosen such that the total amount of degrees of freedom is equal to 160. It can be seen that although high order shape functions are generally suitable for smooth solutions, they perform at least in this test as good as low order shape functions. In this example the best compromise might be the choice of ansatz order 4.

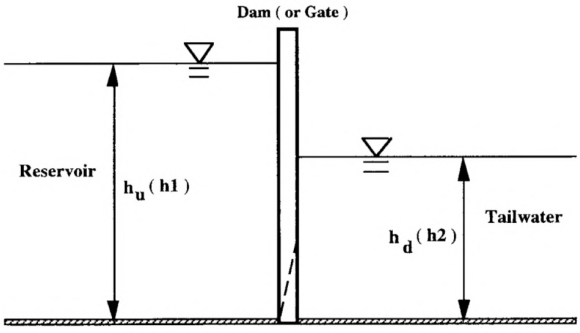


Figure 5.1: 1-d dam break model problem [Hsu and Yeh, 2002]

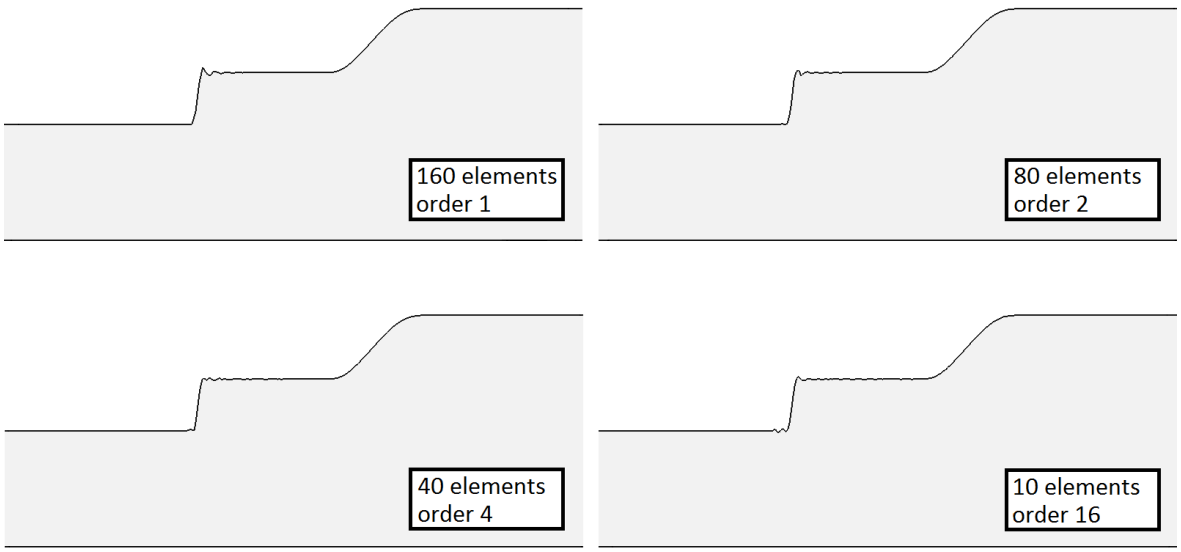


Figure 5.2: 1-d dam break problem computed with the CBS scheme

If the ratio of the both initial heights is larger, such that the gradient of the surface increases, the oscillations at the discontinuity reach an unacceptable level. In this case it is indispensable to apply shock capturing method, as presented in section 4.7.

5.2 Shock development despite an initial smooth Gaussian bell surface

As discussed in section 2.7.1, the shallow water equations have a similar form compared to the Euler equations for compressible fluids. To show the formation of shocks, similar to those observed in the solution of Euler equations, a 2-d example with a perfectly smooth initial Gaussian bell surface was computed. The results shown in figure 5.3 were obtained using a square domain with length 10 and the following initial conditions:

$$\eta^0(x_1, x_2) = 2.5 \cdot \exp\left(- \frac{(x_1 - 5)^2 + (x_2 - 5)^2}{1.5^2} \right)$$

$$u_1^0(x_1, x_2) = u_2^0(x_1, x_2) = 0$$

$$H(x_1, x_2) = -1;$$

The number of elements in each direction was set to 100, the ansatz order to 2 and the time step length to 0.008. It can be seen that the wave front moves slower than the top, because the wave celerity is proportional to the height. As a result, the slope increases until it forms a discontinuous shock, which can, of course, not be represented exact.

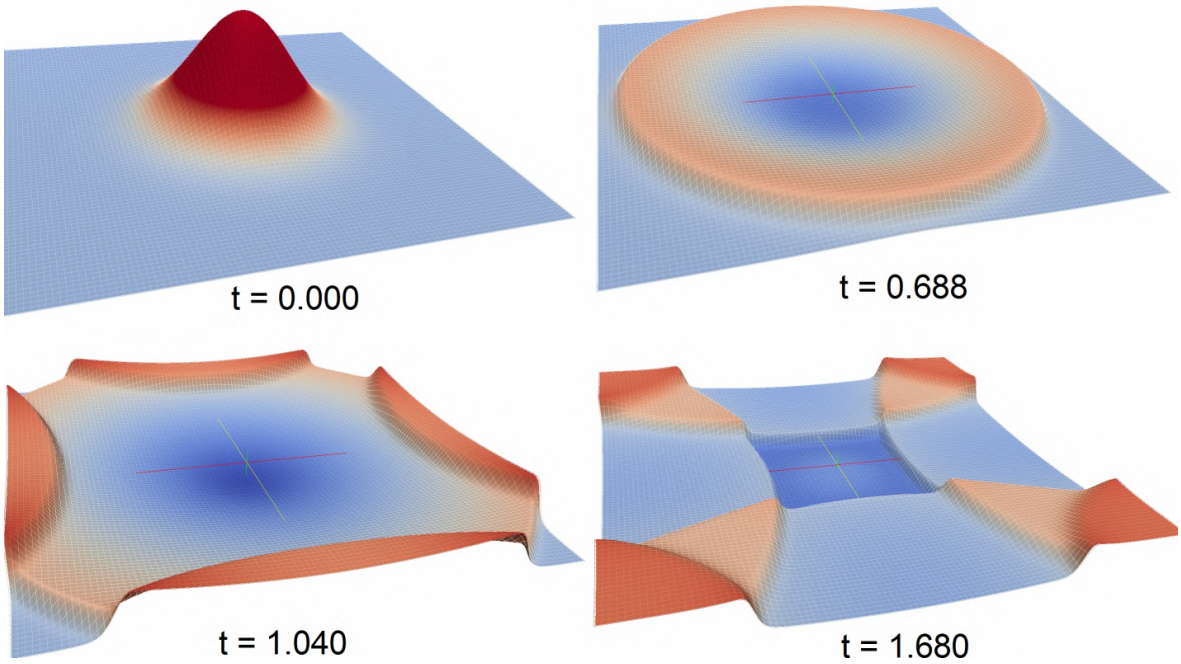


Figure 5.3: Gaussian bell initial condition

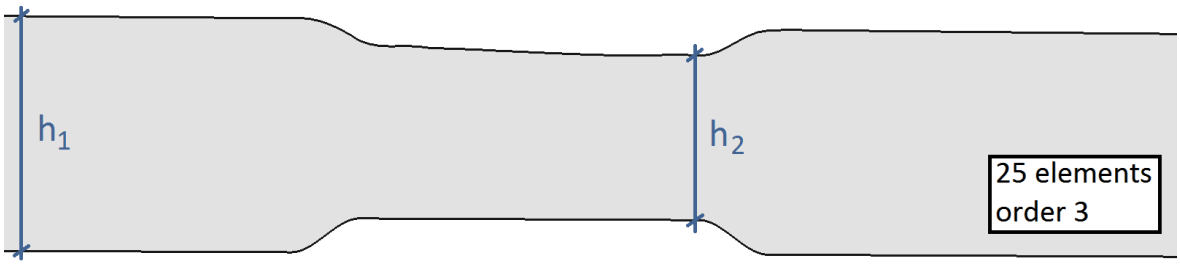


Figure 5.4: Sub-critical flow over bump

5.3 Other validation examples

As the shallow water equations are capable of describing various phenomena, a few additional examples were set up, to 'confront' the presented algorithm with other situations.

The first test case is a 1-d (almost) steady-state flow over a bump, that causes the water level to decrease, as shown in figure 5.4. This happens because the energy height relative to the ground is on the bump 0.5 m less than elsewhere. The validation can be carried out by Bernoulli's equation

$$H = h + \frac{u^2}{2g} = h + \frac{U^2}{h^2 2g},$$

which relates the energy height H with the flow depth h and the kinetic energy height $v^2/2g$. The example shown in figure 5.4 gave $h_1 = 3.45\,\mathrm{m}$ and $h_2 = 2.385\,\mathrm{m}$ which corresponds to the energy heights of $H_1 = 3.8782\,\mathrm{m}$ and $H_2 = 3.281\,\mathrm{m}$ (by inserting h and $U = 10\,\mathrm{m}$ into Bernoulli's equation). It can be seen that the difference in the energy height ($H_1 - H_2 = 0.5972$) is with an error of 0.097 m equal to the height of the bump ($\Delta x = 0.5\,\mathrm{m}$). This gives an error of 2.96% relative to H_2.

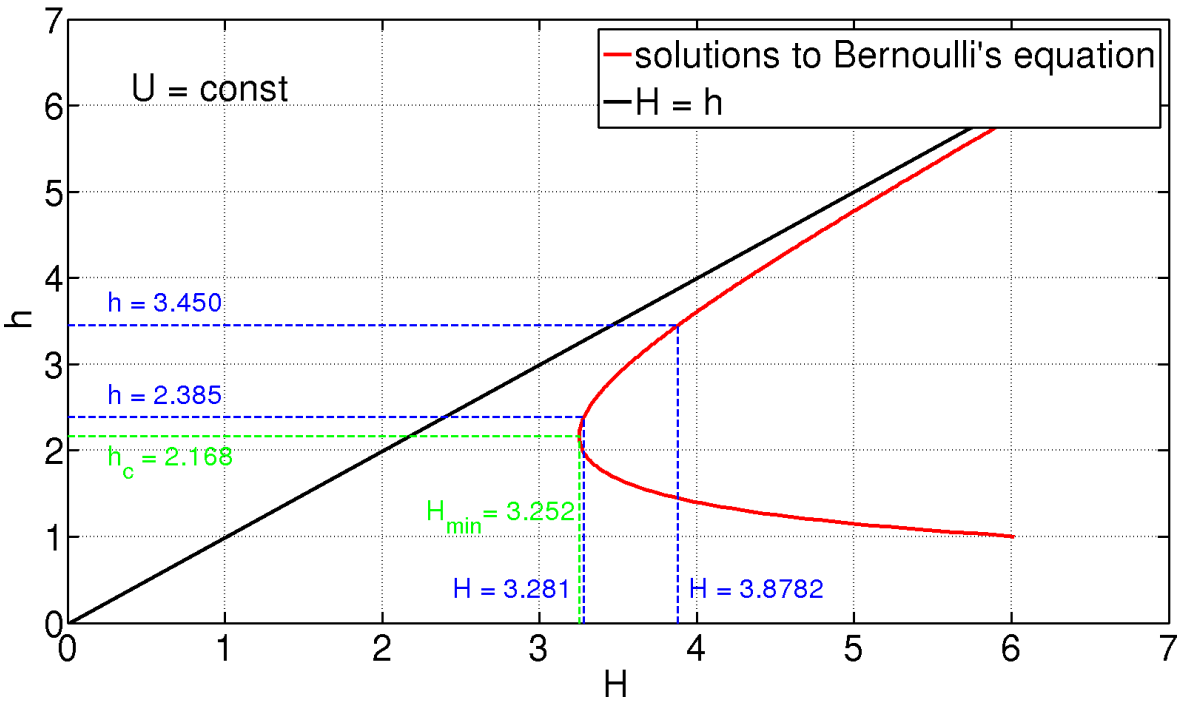


Figure 5.5: Plot of solutions to Bernoulli's equation for U = 10

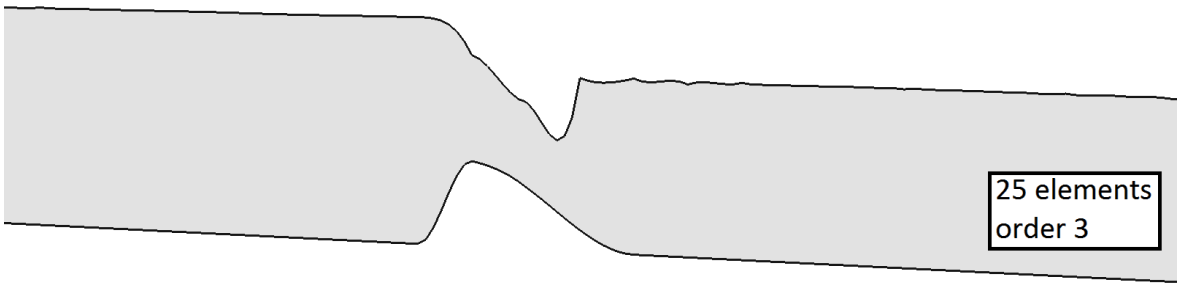


Figure 5.6: Flow over bump followed by a beginning hydraulic jump (scaled by a factor of 10)

If Bernoulli's equation is plotted for a constant value of U (often called specific discharge q), it can be seen nicely how the fluid depth behaves for changes in the energy height. In figure 5.5, this was done for $U = 10$. By differentiating Bernoulli's equation as

$$\frac{dH}{dh} = 0,$$

it turns out, that the extreme value $H_{\min}$ and its corresponding water depth h_c can be determined by:

$$h_c = \sqrt[3]{\frac{U^2}{g}} \qquad \text{and} \qquad H_{\min} = \frac{3}{2} h_c.$$

It can also be seen in figure 5.5 that the energy of the fluid on the bump was very close to $H_{\min}$ in the test case.

If the energy of the fluid is not enough to pass the obstacle, the height increases until the energy on the bump is exactly equal to $H_{\min}$ and continues with a super-critical flow after-

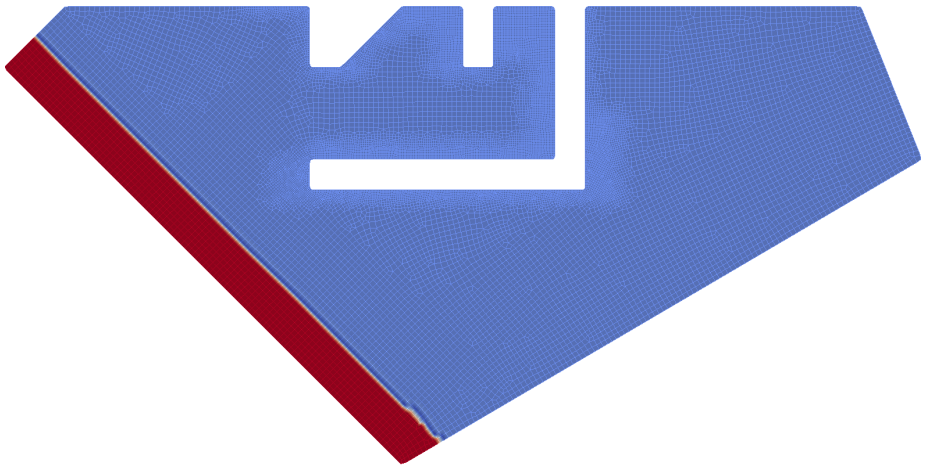


Figure 5.7: wave simulation by taking a dam break as initial condition

wards. If the flow is normally sub-critical, at some point after the bump, a hydraulic jump appears, as the high amount of bottom friction slows the fluid down. In a second example, shown in figure 5.6, this was tested with the CBS algorithm. The beginning hydraulic jump and the differences in the water level can be seen nicely, however, the simulation becomes instable in the next time steps, as oscillations lead to a negative height. This example shows, that describing such phenomena requires the application of shock capturing methods (see section 4.7).

Additionally, it was possible to validate bottom friction and ground slope terms, as the simulation completely coincided with the initial conditions, set according to the Manning-Strickler formula (see also section 2.4.1):

$$u = \frac{U}{h} = k_{\text{st}} \cdot \sqrt{I} \cdot h^{\frac{2}{3}} \qquad \text{or} \qquad h = \left(\frac{U}{k_{\text{st}} \sqrt{I}} \right)^{\frac{3}{5}}$$

In both examples the bottom topography was parameterized by cubic polynomials, which could be represented exactly in the test case when using ansatz order ≥ 3. Consequently, discontinuities in the derivatives, usually arising from a first order approximation, were omitted.

5.4 Wave entering a harbour

To show that the CBS algorithm is capable of handling real world scenarios, a complex harbour shape was created to simulate a wave entering it. The wave was generated by using a ‘dam break‘ initial condition near the left boundary, as seen in figure 5.7. As a result a long wave with half of the initial height propagates in the direction of the harbour at an angle of 45° (see figure 5.8). The mesh consists of $20k$ elements with a linear ansatz. η at was chosen at $t = 0$ as 1 on the left size of the ‘dam‘ and 0 otherwise. Additionally H was set to -1 and the time step length to 0.03. Because previous computations have shown instabilities at sharp corners they were smoothed by a radius of 1 to get a stable solution.

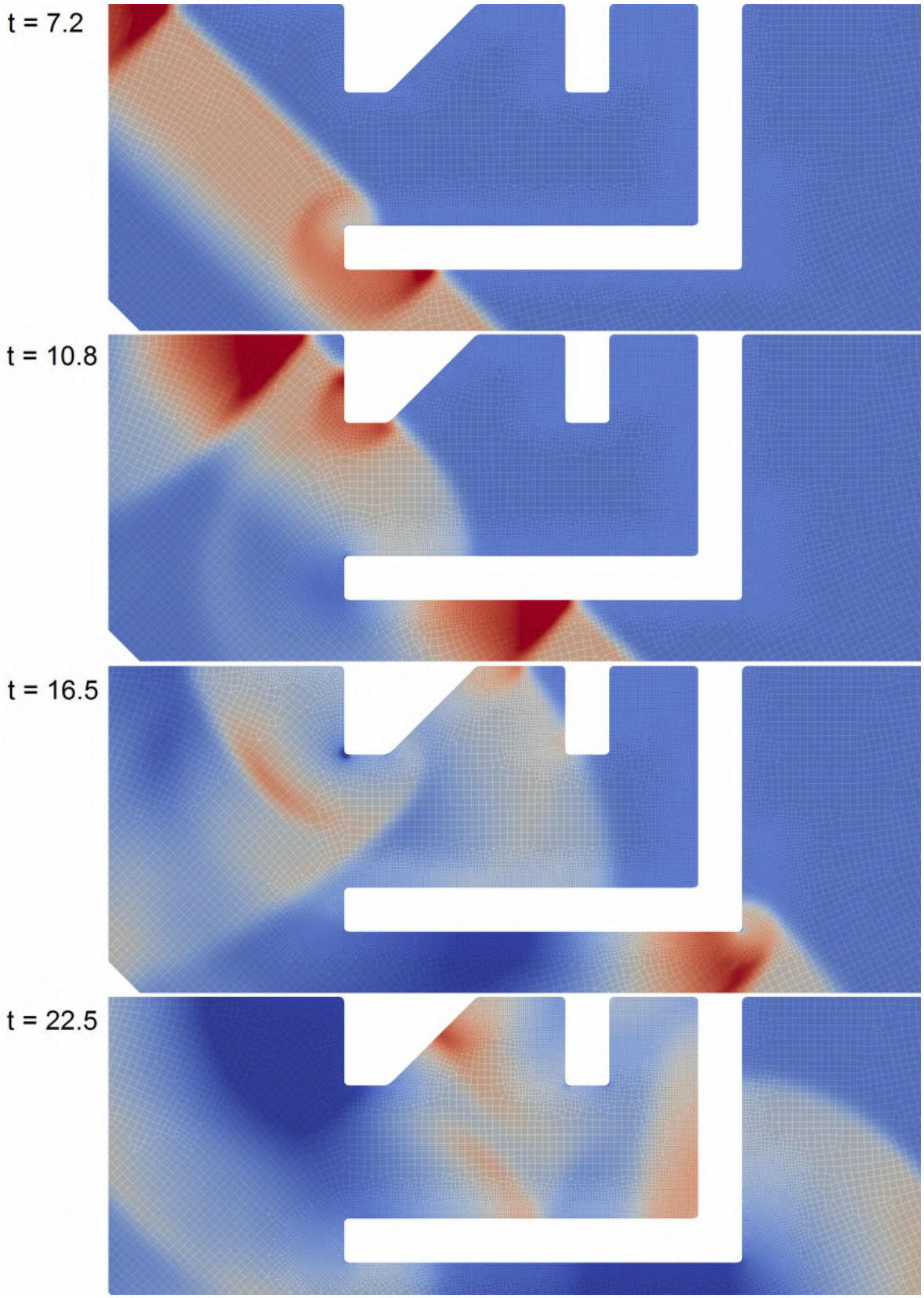


Figure 5.8: Long wave entering harbour

Chapter 6

Conclusion

After a period of analysing the Characteristic Based Split algorithm it can be concluded that the overall performance related to the presented benchmarks is satisfactory. Compared to other schemes implementing some kind of balancing diffusion, accuracy and capability to represent shock-waves proved to be at least as good. One advantage that should be emphasized is the flexibility of the algorithm to handle different problems with the same procedure, meaning that the additional afford for implementing for instance compressible Euler equations or incompressible Navier-Stokes equations is manageable. Having a generally applicable algorithm for the solution of fluid dynamics equations is definitely beneficial. Moreover, the combination with high order shape functions turned out to give remarkable results, especially for smooth geometries.

However, there are still some difficulties directly related to the CBS algorithm that do require some investigation and if possible improvement. An important phenomenon often observed at extreme points, such as corners, are sudden instabilities that lead to a total crash of the simulation. In order to predict such instabilities, a more detailed analysis on the conditions of their appearance has to be done. Additionally, it has to be investigated if there are any possibilities to overcome the severe restriction of prohibiting negative height values. As other schemes manage to avoid this, there might be a way to practically circumvent the negative square root arising in those cases (see equation 3.33). In connection with that a robust procedure for allowing moving boundaries has yet to be found. The method suggested by [Zienkiewicz *et al.*, 2005] to absorb boundary movement by repositioning element nodes is not suitable for typical environmental engineering problems, as here large variations in the 'wet' domain need to be expected.

Besides, the implementation of a complete package for solving shallow water problems is not complete yet. Although basic functionality for treatment of boundary conditions is already available, the implementation of source terms requires still further work. This includes analysing the accuracy of the currently used bottom friction model from Manning-Strickler as well as investigating the possibility to implement the divergence form of the bed slope source term (see 4.3).

Regarding the goal of finding a robust finite element scheme for handling environmental engineering problems the CBS algorithm is especially for situations without extreme conditions that do not require moving boundaries a good choice. Furthermore, the combination with high

order shape functions enables an exact representation of smooth geometries parameterized, for example, by cubic splines or NURBS, arising from CAD applications. Thus the presented method has indeed the potential to be successful in practical engineering applications.

List of Figures

List of Tables

Bibliography

I. Babŭska and W. G. Szymczak. The p-version of the finite element method. *SIAM Journal on Numerical Analysis, pp. 515-545*, 1981.

I. Babuška, B. A. Szabó, and I. N. Katz. An error analysis for the finite element method applied to convection diffusion problems. *Computer Methods in Applied Mechanics and Engineering, vol. 31, pp. 19-42*, 1982.

I. Babuška. Error-bounds for finite element method. *Numerische Mathematik*, 1971.

A.N. Brooks and T.J.R. Hughes. Streamline upwind/petrov-galerkin formulation for convection dominated flows with particular emphasis on the incompressible navier-stokes equation. *Computer Methods in Applied Mechanics and Engineering*, 1982.

J Donea and Antonio Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons Inc., New Jersey, 2003.

J. Donea. A taylor-galerkin method for convective transport problems. *International Journal for Numerical Methods in Engineering*, 1984.

Alexander Düster. High-Order FEM. Lectures notes, Chair for Computation in Engineering, Technische Universität München, 2008.

Chih-Tsung Hsu and Keh-Chia Yeh. Iterative explicit simulation of 1d surges an dam-break flows. *International Journal for Numerical Methods in Fluids*, 2002.

Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, New York, 2000.

Daniel R. Lynch and William G. Gray. Finite element simulation of flow in deforming regions. *Journal of Computational Physics*, 1980.

R. Löhner, K. Morgan, and O. C. Zienkiewicz. The solution of non-linear hyperbolic equation systems by the finite element method. *International Journal for Numerical Methods in Fluids*, 1984.

Stephen C. Medeiros and Scott C. Hagen. Review of wetting and drying algorithms for numerical tidal flow models. *International Journal for Numerical Methods in Fluids*, 2006.

M. Morandi-Cecchi and M. Venturin. Characteristic-based split (cbs) algorithm finite element modelling for shallow waters in the venice lagoon. *International Journal for Numerical Methods in Fluids*, 2006.

Machiels Olivier, Erpicum Sébastien, Archambeau Pierre, Dewals Benjamin, and Pirotton Michel. Bottom friction formulations for free surface flow modeling. *Fond National de la Recherche Scientifique*, 2009.

Pablo Ortiz, Olgierd C. Zienkiewicz, and Joanna Szmelter. CBS Finite Element Modelling of Shallow Water and Transport Problems. *European Congress on Computational Methods in Applied Sciences and Engineering*, 2004.

R. Alan Plumb. Athmospheric and Oceanic circulations. Lectures notes, Department of Earth, Atmospheric, and Planetary Sciences, Massachusetts Institute of Technology, 2003.

M. Quecedo and M. Pastor. A reappraisal of taylor–galerkin algorithm for drying–wetting areas in shallow water computations. *International Journal for Numerical Methods in Fluids*, 2002.

Redmine page of AdhoC++, Chair for Computation in Engineering. Key classes of the analysis pipeline, 2014.

R Rössert. *Hydraulik im Wasserbau.* R. Oldenburg Verlag, München, 1999.

J. J. Stoker. *Water Waves: The Mathematical Theory with Applications.* Interscience Publishers, Inc., New York, 1957.

Barna Szabó and Ivo Babuška. *Finite Element Analysis.* John Wiley Sons, Inc., New York, 1991.

Craig G. Thomas and Perumal Nithiarasu. The characteristic based split (cbs) approach for invicid compressible flow problems. effect of element size in the streamline direction. *European Congress on Computational Methods in Applied Sciences and Engineering*, 2004.

Alessandro Valiani and Lorenzo Begnudelli. Divergence form for bed slope source term in shallow water equations. *Journal of hydraulic engineering*, 2006.

C. B. Vreugdenhil. *Numerical Methods for Shallow-Water Flow.* Kluwer Academic Publishers, Dordrecht, 1994.

Tan Weiyan. *Shallow Water Hydrodynamics: Mathematical Theory and Numerical Solution for a Two-dimensional System of Shallow Water Equations.* Elsevier Science, Amsterdam, 1992.

O. C. Zieniewicz and R. Codina. A general algorithm for compressible and incompressible flow-part i. the split, characteristic-based scheme. *International Journal for Numerical Methods in Fluids*, 1995.

O. C. Zienkiewicz and P. Ortiz. A split-characteristic based finite element model for the shallow water equations. *International Journal for Numerical Methods in Fluids*, 1995.

O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method for Fluid Dynamics, Fifth Edition.* Elsevier, Amsterdam, 2000.

O. C. Zienkiewicz, R. L. Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics, Sixth Edition.* Elsevier, Amsterdam, 2005.