# Two step FVE method

## A numerical modeling technique designed for error insight

J. Mooiman

2025-02-04 23:44:20

# Two step FVE method

A numerical modeling technique designed for error insight


**Authors**

Jan Mooiman


**Partners**

Members of the team (dd. 22 May 2024)
Adri Mourits
Bea Saggiorato
Frank Platzek
Jan Mooiman
Mart Borsboom
Sven Westerbeek
Thea Vuik

Cover photo: Fedderwardersiel, Germany

# Contents

# List of Tables

# List of Figures

# List of Symbols

| Symbol | Unit | Description |
|---|---|---|
| $\Delta t$ | s | Time increment |
| $\Delta x$ | m | Space increment, $\Delta x_{i+\frac{1}{2}} = x_{i+1} - x_i$ |
| $\Omega$ | - | Finite volume |
| $\Psi$ | $\mathrm{m^2\,s^{-1}}$ | Artificial smoothing coefficient |
| $\theta$ | - | $\theta$-method. If $\theta = 1$ then it is a fully implicit method and if $\theta = 0$ then it is a fully explicit method. |
| $\boldsymbol{E}$ | - | Error vector function, defined in computational space |
| $\xi$ | - | Relative coordinate |
| $\zeta$ | m | Water level above reference plane, positive upward |
| $c_\Psi$ | $1\ (.)^{-1}$ | Artificial smoothing variable |
| $g$ | $\mathrm{m\,s^{-2}}$ | Gravitational constant |
| $h$ | m | Total water depth |
| $i$ | - | node counter |
| $q$ | $\mathrm{m^2\,s^{-1}}$ | The water flux in $x$-direction, $q = hu$ |
| $r$ | $\mathrm{m^2\,s^{-1}}$ | The water flux in $y$-direction, $r = hv$ |
| $t$ | s | Time coordinate |
| $t_{reg}$ | s | The regularization time for the given time-series |
| $u$ | $\mathrm{m\,s^{-1}}$ | Velocity in $x$-direction |
| $v$ | $\mathrm{m\,s^{-1}}$ | Velocity in $y$-direction |
| $x$ | m | $x$-coordinate |
| $z_b$ | m | Bed level above reference plane, positive upward |

# List of To Do's

# 1 Introduction

Nature can be described by mathematical models, these models approximate the behaviour of nature. The main question is: "How well will these mathematical models describe nature?". This document is based on Borsboom (1998), where the mathematical model is called the "*difficult probem*".

To show that the mathematical model does not match with nature by using a numerical method, it is needed that the numerical error of the numerical method can be quantified. The numerical errors should be very small compared to the errors made in the mathematical model. In that case the results of the numerical model is a reliable approximation of the mathematical model. A mismatch in results of the numerical method w.r.t. the nature is then fully determined by the mismatch of the mathematical model.

In this report a derivation is reported of a numerical model that automatically is adjusted to assure that the numerical result is close enough to the solution of the mathematical model. To obtain such a numerical model the mathematical model should be adjusted to a state which is suitable to determine what and how large the mismatch is. The mathematical model is adjusted in that way that the second derivatives of all data is smooth. After smoothing of the data the mathematical model is called "*easy problem*". This step in the procedure is called "*regularization*" and is the first step of the two step FVE method (Finite Volume Elment method).

The regularization step has to ensure that the lowest-order terms of the residual of the discretization step are dominant, so that we can limit ourselves to the analysis of the leading terms of the error expansion. The regularization is assumed to be such that the easy problem can be discretized accurately on the available grid, and that the leading terms of the series expansions are dominant.

To obtain this goal the numerical scheme should be central in space, no dissipation is added to the model by the numerical method, just dispersion. All examples in this document will be performed with a fully implicit time-integrator using a iteration mechanism based on the Newton-linearization. The Newton-iteration process benefits of the regularized data. This is the second step of the two step FVE method (Finite Volume Elment method).

When all the mentioned items are fulfilled then the numerical scheme is:

1 accurate (2nd order, due to the requirement that there is no numerical dissipation),
2 reliable (numerical errors are reduced to be much less then the modelling deviations),
3 robust (no numerical restrictions on time step other then physical restrictions),
4 flexible (separation of numerical and physical part, lot of numerical methods can be used without hampering the physical part),
5 efficient (Newton method is a second order method),
6 fast (fully implicit).

The feasibility of this method is shown by performing this method on the 1D shallow water equations. Towards these shallow water equations we will look first to the hyperbolic part of these equations. The boundary conditions are separated in a stricly outgoing and a strictly ingoing signal. When selecting a special combination of these signals a weakly-reflective or absorbing boundary condition can be prescribed, including a prescribed ingoing signal.

In chapter 2: Two-step numerical modeling, error minimizing, the error-minimizing integration method is presented. The error-minimizing integration method is based on the assumption that a function can be made smooth so that the numerical discretization and the regularized function are so close that the numerical error is negligible for that function.

In chapter 3: 1-D Space discretisation, the one dimensional space discretisation. Which consist of a finite volume method and central discretizations and piecewise linear functions between the nodes. Also an estimation of the regularization coefficient for a given function based on the second order of accuracy of the discretization is presented. So the user is able to justify the quality of the numerical solution and in that way to judge where to adjust the regularization or adapt the grid in certain regions.

In chapter 4: Time integration scheme, the fully implicit time integration is based on Newton iteration presented. Due to the regularization of the data the Newton iteration converges extremely well, that is second order in also the more complex areas.

In chapter 5: Towards the shallow water equations, the fully implicit time integration is presented for one dimensional shallow water equations. We start

with the implementation of the 1D advection/transport equation, then with the implementation of the wave equation without convection. This 1D wave equation consist of two independent advection/transport equation for a right and left going signal. We start with 1D advection/transport equation because this equation because this equation has the same nature as the right going signal of the wave equation.

In **??**: **??**, the fully implicit time integration is presented for the advection diffusion equation. Showing a flow from left to right with a interface in the diffusion coefficient for the transported constituent.

In **??**: **??**, the fully implicit time integration is presented for the non-linear wave equations. These equations are separated in a left- and right-going signal represented by a left and right transport equation. At the boundary these two equations are coupled and will therefor generate reflections in the numerical model.

# 2 Two-step numerical modeling, error minimizing

For the realization of our objective, an error analysis is required to gain insight in the relative importance of discretization errors. This has to be in the form of power series expansions to be genuinely generally applicable. Smoothness is required to ensure fast converging series and dominant lowest-order terms that can be used as a basis for reliable local error approximations. Artificial smoothing is added to satisfy this requirement, if necessary. To enable the physical interpretation of numerical errors afterwards, smoothing can only involve the artificial enhancement of physical dissipation. Taylor-series expansions can be used to determine the leading terms of the residual. The residual, however, is not a suitable error measure since it indi- cates the local discretization error in the equations, not in the solution. In order to be useful, the residual needs to be reformulated in terms of local solution errors. We did not find any existing scheme that allows for such a transformation, and so we developed a discretization method that does. The result turns out to be a method of finite volume type. The discretization consists of integrating the model equations over control volumes, using uniquely defined discrete approximations of all variables. The proposed numerical modeling technique solves the conceptual model problem in two steps (Figure 2.1: in the first step the difficult problem to be solved is changed into an easy problem by adding artificial smoothing; in the second step the easy problem is discretized.

Showing the two-step method in general

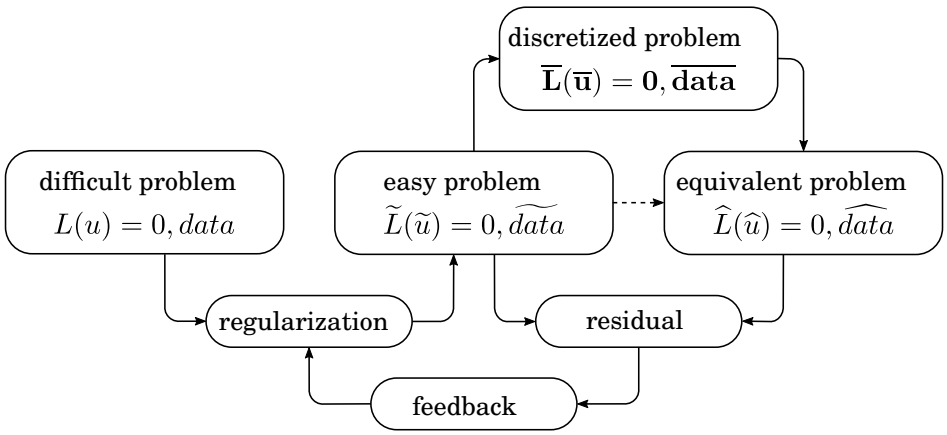



***Figure 2.1:*** *Graphical presentation of the error-minimizing two step method*

Notation agreements:

| | |
|---|---|
| $u$ | Non-regularized/non-smoothed function, to be determined numerically. |
| $\widetilde{u}$ | Regularized/smoothed function, denoted by the wavy line. |
| $\overline{u}$ | Piecewise linear function, denoted by the bar. |
| $\widehat{u}$ | Numerical solution on the nodes. |

A tilde ($\widetilde{u}$) indicates the variables and differential operators of the easy problem. Their discretizations are indicated by a bar, $\overline{u}$. Next, we define the smooth and infinitely differentiable function $\widehat{u}$ that is a very close approximation of numerical solution $\overline{u}$. By means of an error analysis we determine the differential problem that $\widehat{u}$ is a solution of. Note that the data pertaining to the computational model are also included in the procedure. Independent variables describing, e.g., the geometry and initial and boundary conditions also need to be discretized and hence need to be sufficiently smooth, to ensure that all higher-order error terms are sufficiently small and can be neglected. Sufficient smoothness is obtained automatically by using smoothing coefficients that are a function of the discretization errors. See also Borsboom (2001).

# 3 1-D Space discretisation

We are looking for a second order central space discretization together with the finite volume approach. So the higher order terms in the Taylor series expansion are negligible. For this numerical discretization method no dissipation is added, only there is some dispersion for the shorter waves, i.e. dependent on the third derivative in the Taylor expansion. To fulfill this requirement the data should be smooth according to the truncation error of the numerical discretization. If not, the data should be made smooth with a procedure in which you can see on which location the smoothing is severe. The process of smoothing is called regularization. If the truncation error (high values of second derivatives) is severe on locations that the user do not expect and do not accept then the user can adjust the discretization in that part of the domain. There are two options to adjust the data:

1 increase the smoothing coefficient or
2 choose a smaller grid size.

## 3.1 Finite volume approach

We will discuss the finite volume approach for the one dimensional case of the function $u(x,t)$, for the grid shown in Figure 3.1

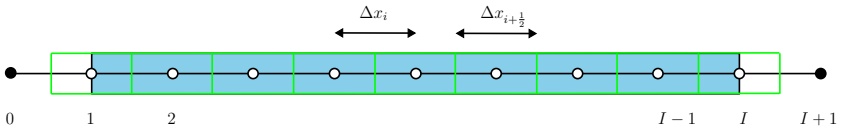



***Figure 3.1:*** *Water body (blue area), finite volumes (green boxes), computational points (open dots), virtual computational points (black dots), boundary points are at* $i = 1$ *(west boundary) and* $i = I$ *(east boundary)*

The control volume is defined on the interval $x_{i-\frac{1}{2}}$ to $x_{i+\frac{1}{2}}$, see Figure 3.1. The grid in Figure 3.1 is equidistant but that is not necessarily, the method which is described in this document also holds for a non-equidistant grid.

### 3.1.1 Quadrature rule, source term

The finite volume approach for the function $u(x,t)$ reads:

$$\int_{\Omega} u \, d\Omega = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u \, dx \tag{3.1}$$

Using piecewise linear functions between the non-equidistant nodes the quadrature rule reads:

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u \, dx = \int_{x_{i-\frac{1}{2}}}^{x_i} u \, dx + \int_{x_i}^{x_{i+\frac{1}{2}}} u \, dx \approx \tag{3.2}$$

$$\approx \frac{\Delta x_{i-\frac{1}{2}}}{2} \frac{u_{i-1} + 3u_i}{4} + \frac{\Delta x_{i+\frac{1}{2}}}{2} \frac{3u_i + u_{i+1}}{4} \tag{3.3}$$

where the control volume is split into two sub adjacent volumes $[x_{i-\frac{1}{2}}, x_i]$ and $[x_i, x_{i+\frac{1}{2}}]$. A graphical interpretation is given in Figure 3.2.

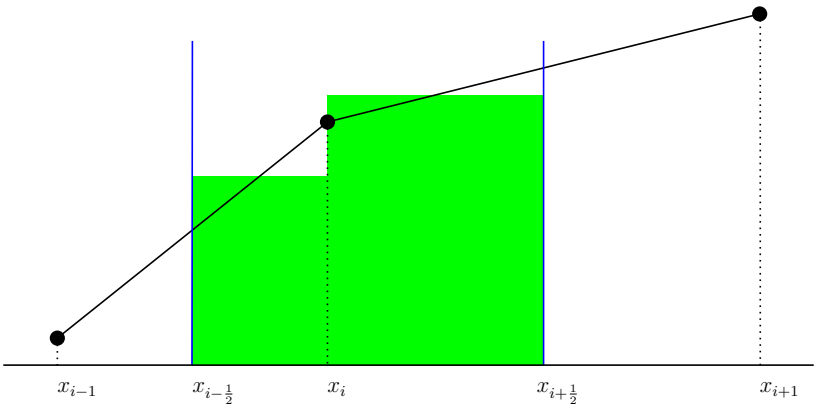



***Figure 3.2:*** *Integration at $x_i$ over the control volume from $x_{i-\frac{1}{2}}$ to $x_{i+\frac{1}{2}}$.*

Higher order quadrature rules may be chosen, but they are not discussed in this report.

### 3.1.2 Quadrature rule, flux term

### 3.1.3 Boundary conditions

At the west and east boundary boundary conditions need to be supplied. If it is a prescribed boundary condition (ingoing signal) it is called an **essential**-boundary condition, if the boundary condition is determined by the outgoing signal it is called a **natural**-boundary condition (Logan, 1987; Kan et al., 2008). In case of a wave equation we have a right and left going signal. When no reflection at an open boundary is required then at the west boundary a signal

need to prescribed without disturbing the left going signal. And at the east boundary the other way around.

Further more we have the prescribed boundary signal at the boundary node of the grid. This assumption is made because user are acquitained with the fact that open boundaries are given at nodes. But for the outgoing signal this is not necessarily, for the outgoing signal we stick to the finite volume approach.

When we first consider a right going one dimensional advection equation (see section 5.2) the boundary at the west side is located at $x_1$ and the outflow condition is located at $x_{I+\frac{1}{2}}$. The prescribed function reads:

$$c(0,t) = f(t) \tag{3.4}$$

Because we use a 3-point stencil for the interior, we aim for a 3-point stencil for the essential boundary condition. One possible quadratic interpolation is a parabolic fit through the grid points, giving an underspecification of the solution at the boundary regardless of its position.

$$\begin{aligned} c_{GP}(\xi) &= c_0\,(1-\xi) + c_1\,\xi + \frac{1}{2}(c_0 - 2\,c_1 + c_2)\,(\xi - 1)\,\xi = \\ &= (1-\xi)\,\left(1 - \frac{1}{2}\xi\right) c_0 + \xi\,\,(2-\xi)\,c_1 + \frac{1}{2}\xi\,(\xi-1)\,\,c_2 \end{aligned} \tag{3.5}$$

where $\xi \in [0,1]$ is the weight function between $c_0$ and $c_1$.

Another possible quadratic fit is the one through Cell-Centered values:

$$c_{CC}(\xi) = c_0\,\,(1-\xi) + c_1\,\xi + \frac{1}{2}(c_0 - 2c_1 + c_2)\,\left(\xi - \frac{1}{2}\right)^2 \tag{3.6}$$

$$= \left(1 - \xi + \frac{1}{2}\left(\xi - \frac{1}{2}\right)^2\right) c_0 + \left(\xi - \left(\xi - \frac{1}{2}\right)^2\right) c_1 + \left(\frac{1}{2}\left(\xi - \frac{1}{2}\right)^2\right) c_2 \tag{3.7}$$

The parabolic fit that gives neither underspecification nor overspecification of imposed values at boundaries is a combination of Equation (45) and (49). It is easy to show that that combination consists of $1/3$ times Equation (3.5) plus $2/3$ times Equation (3.8):

$$c_{opt}(\xi) = \left(\frac{13}{12} - \frac{3}{2}\xi - \frac{1}{2}\xi^2\right) c_0 + \left(-\frac{1}{6} + 2\xi - \xi^2\right) c_1 + \left(\frac{1}{12} - \frac{1}{2}\xi + \frac{1}{2}\xi^2\right) c_2 \tag{3.8}$$

Which lead to the following interpolations for $\xi = \frac{1}{2}$ and $\xi = 1$:

$$c_{opt}\left(\frac{1}{2}\right) = \frac{11}{24}c_0 + \frac{14}{24}c_1 - \frac{1}{24}c_2 \tag{3.9}$$

$$c_{opt}\left(1\right) = \frac{1}{12}c_0 + \frac{10}{12}c_1 + \frac{1}{12}c_2 \tag{3.10}$$

Where Equation (3.9) will be used for the natural boundary condition and Equation (3.10) will be used for the essential boundary condition.

Verification of its correctness by means of integration over the outermost finite volume:

$$\int_{\xi_{\frac{1}{2}}}^{\xi_{\frac{3}{2}}} c_{opt}(\xi)\, d\xi = \frac{1}{8}c_0 + \frac{3}{4}c_1 + \frac{1}{8}c_2 \tag{3.11}$$

In the right-hand side we see the weights of the mass matrix of the piecewise linear FVE method, i.e. averaged over the left outermost finite volume the quadratic function $c_{opt}$ equals the piecewise linear function used in the FVE scheme.

## 3.2 Regularization of given function

To get an error-minimizing method we need to regularize all data, as mentioned in chapter 2. Regularization of a given function is performed as described in Borsboom (1998) and Borsboom (2003). The function to regularize reads:

$$u(x) = u_{giv}(x). \tag{3.12}$$

The regularized function $\widetilde{u}$ reads (Borsboom, 1998, eq. 6):

$$\widetilde{u} - \frac{\partial}{\partial x}\Psi\frac{\partial \widetilde{u}}{\partial x} = u_{giv}, \qquad \Psi = c_\Psi \Delta x^2 E, \tag{3.13}$$

where

| | |
|---|---|
| $u_{giv}$ | Given function, ex. bathymetry, viscosity, ..., $[\,\cdot\,]$. |
| $\widetilde{u}$ | Regularized/smoothed function of $u_{giv}$, $[\,\cdot\,]$. |
| $\Psi$ | (Artificial) smoothing coefficient, [m$^2$] |
| $\Delta x$ | Space discretization, $\Delta x_{i+\frac{1}{2}} = x_{i+1} - x_i$, [m] |
| $c_\Psi$ | Smoothing factor (set by user), $[1/\,\cdot\,]$ |
| $E$ | Error function, $[\,\cdot\,]$ (see section 3.2.1) |

With some notation agreements:

| | |
|---|---|
| $u$ | Non-regularized/non-smoothed function, to be determined numerically. |

| | |
|---|---|
| $\widetilde{u}$ | Regularized/smoothed function, denoted by the wavy line. |
| $\overline{u}$ | Piecewise linear function, denoted by the bar. |
| $u_i$ | Value of the numerical value at point $x_i$, denoted by the subscript. |

First the discretization of Equation (3.13) is discussed and second the determination of the artificial smoothing coefficient $\Psi$.

The finite volume approach of Equation (3.13) reads:

$$\int_\Omega \widetilde{u}\, d\Omega - \int_\Omega \frac{\partial}{\partial x}\Psi\frac{\partial \widetilde{u}}{\partial x}\, d\Omega = \int_\Omega u_{giv}\, d\Omega \tag{3.14}$$

### *Integration of the first term*

Integration of the first term yields:

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \widetilde{u}\, dx \approx \frac{1}{2}\Delta x_{i-\frac{1}{2}}\left(\frac{1}{4}u_{i-1}+\frac{3}{4}u_i\right)+\frac{1}{2}\Delta x_{i+\frac{1}{2}}\left(\frac{3}{4}u_i+\frac{1}{4}u_{i+1}\right) \tag{3.15}$$

### *Integration of the second term*

Integration of the second term, with $\Psi_{i+\frac{1}{2}} = \frac{1}{2}(\Psi_i + \Psi_{i+1})$:

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial x}\Psi\frac{\partial \widetilde{u}}{\partial x}dx = \Psi\frac{\partial \widetilde{u}}{\partial x}\bigg|_{i+\frac{1}{2}} - \Psi\frac{\partial \widetilde{u}}{\partial x}\bigg|_{i-\frac{1}{2}} \approx \tag{3.16}$$

$$\approx \Psi_{i+\frac{1}{2}}\frac{u_{i+1}-u_i}{\Delta x_{i+\frac{1}{2}}} - \Psi_{i-\frac{1}{2}}\frac{u_i-u_{i-1}}{\Delta x_{i-\frac{1}{2}}} = \tag{3.17}$$

$$= \frac{\Psi_{i-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2}}}u_{i-1} - \left(\frac{\Psi_{i-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2}}}+\frac{\Psi_{i+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}\right)u_i + \frac{\Psi_{i+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}u_{i+1} \tag{3.18}$$

### *Integration of the right hand side*

For the integration of the right hand side we could use a smaller integration step size, to incorporate the sub-grid scale effects. If the function $u_{giv}$ is for example an analytic function this integral can be computed exact or data is given by a lot of measurement points per finite volume, see for example Figure 3.3. When integrating over a finite volume the sub-grid scale effects will be taken into account

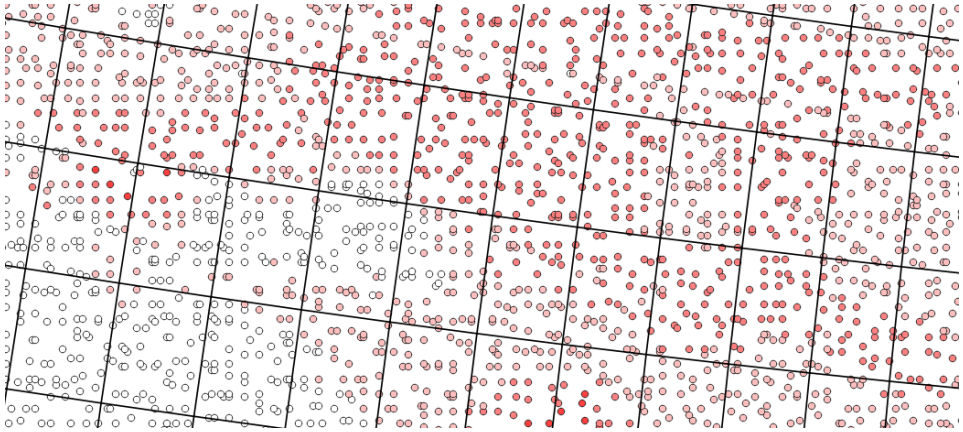


***Figure 3.3:*** *Two dimensional example of a lot of data points per grid cell. The data points are used to compute the integral at the righthand side of Equation (3.20).*

If the function $u_{giv}$ is for example an analytic function this integral can be computed exact.

### *Discretization of Equation (3.14)*

So the discretization of Equation (3.14) with $\alpha = \frac{1}{8}$, read (i.e. Borsboom (1998, eq. 7) and Borsboom (2003, eq. 6)):

$$\left(\frac{\Delta x_{i-\frac{1}{2}}}{8} - \frac{\Psi_{i-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2}}}\right) u_{i-1} + \left(\frac{3\Delta x_{i-\frac{1}{2}}}{8} + \frac{\Psi_{i-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2}}} + \frac{3\Delta x_{i+\frac{1}{2}}}{8} + \frac{\Psi_{i+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}\right) u_i + \tag{3.19}$$

$$+ \left(\frac{\Delta x_{i+\frac{1}{2}}}{8} - \frac{\Psi_{i+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}\right) u_{i+1} = \int_{x_{i-1/2}}^{x_{i+\frac{1}{2}}} u_{giv}\, dx \tag{3.20}$$

The boundary conditions to close the three diagonal system are $u_0 = u_{giv}(x_0)$ and $u_{I+1} = u_{giv}(x_{I+1})$.

### 3.2.1 Determination of artificial smoothing coefficient $\Psi$

The artificial smoothing coefficient $\Psi$ ( $= c_\psi \Delta x^2 E$, Equation (3.13)) is dependent on error $E$, the smoothing coefficent $c_E$ and the second derivative of the given function $u_{giv}$ (Borsboom, 1998, eq. 8). The error $E$ will be computed in computational space, meaning that a disturbance is spreaded over an equal number of cells before and after the location of the disturbance:

$$\left(\frac{\Delta\xi}{8} - \frac{c_E}{\Delta\xi}\right) E_{i-1} + \left(\frac{6\Delta\xi}{8} + \frac{2c_E}{\Delta\xi}\right) E_i + \left(\frac{\Delta\xi}{8} - \frac{c_E}{\Delta\xi}\right) E_{i+1} = \int_{i-\frac{1}{2}}^{i+\frac{1}{2}} |D_i| \; d\xi \tag{3.21}$$

with $\Delta\xi = 1$ it reads:

$$\left(\frac{1}{8} - c_E\right) E_{i-1} + \left(\frac{6}{8} + 2c_E\right) E_i + \left(\frac{1}{8} - c_E\right) E_{i+1} = |D_i| \tag{3.22}$$

**Choose $c_E$ equal to $c_\psi$** and take into account that $D_i$ is constant over a control volume

$$D_i = \Delta\xi^2 \frac{\partial^2 u_{giv}}{\partial \xi^2} \quad \text{(Borsboom, 1998, eq. 2)} \tag{3.23}$$

$$\approx u_{giv_{i-1}} - 2u_{giv_{i-1}} + u_{giv_{i+1}} \tag{3.24}$$

Now system the system of Equation (3.22) can be solved and where $\boldsymbol{\Psi}$ is set to:

$$\boldsymbol{\Psi} = c_\Psi \Delta x^2 \boldsymbol{E} \tag{3.25}$$

For an estimation of $c_\Psi$ see section A.2, in this document we use $c_\Psi = 4$.

The boundary conditions to close the three diagonal system are:

$$2E_0 - E_1 = |D_1| \quad \text{and} \tag{3.26}$$

$$-E_I + 2E_{I+1} = |D_I| \,. \tag{3.27}$$

### 3.2.2 Step function (Heaviside function)

For a full description of this example see Borsboom (2003, eq. 5). The function is initially defined as:

$$u_{given}(x) = \begin{cases} 0, & \text{if} \quad 0\ [m] \leq x \leq 1000\ [m], \\ 1, & \text{if} \quad 1000\ [m] < x \leq 2000\ [m], \end{cases} \tag{3.28}$$

For illustration, the step is chosen to be exactly on a grid point (in the middle of a control volume at $1000\ [m]$) and at the finite volume boundary, i.e. a half $\Delta x$ shifted to the right.

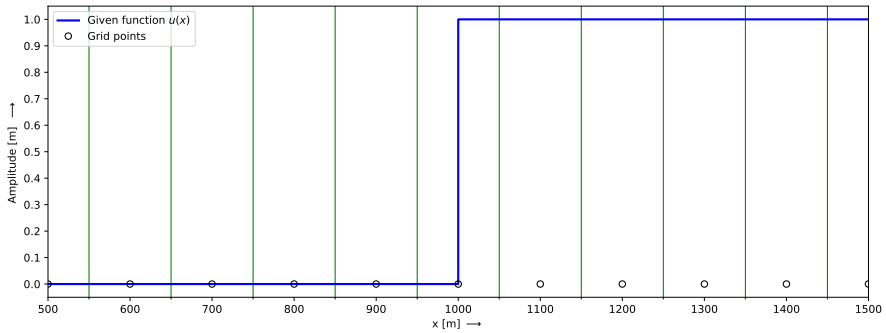
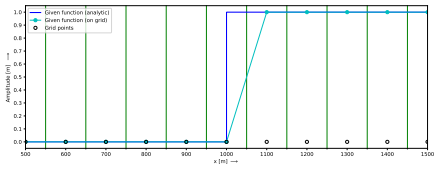
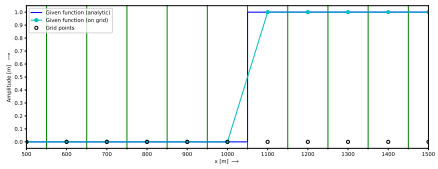



***Figure 3.4:*** *Step function to be estimated. The thin green vertical lines indicate the borders of the finite volumes.*

A straight forward piecewise approximation is shown in Figure 3.5. Both figures does show the same discretization function (cyan colored) but the step is at a different location.



***(a)*** *Step at grid node,* $x = 1000\ [m]$

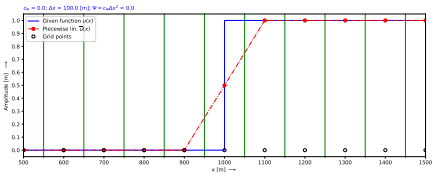


***(b)*** *Step at boundary of a finite volume,* $x = 1050\ [m]$

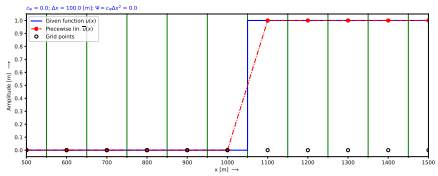
***Figure 3.5:*** *As seen from this figures the function defined on the grid nodes does not see the exact location of the step. Both function through the grid nodes have the same profile, but the step is at an other location.*

A piecewise approximation with a regularization coefficient of zero ($c_\Psi = 0$) is shown in Figure 3.6. Which looks quite well, but what is the value of the second derivative of the solution. For $\Delta x = 100\ [m]$ the absolute value of the second derivative is

***TODO 3.1:*** *Determine the value of the second derivative, Dx = 100 m and Dx = 50 m* ***TODO***



***(a)*** *Step at grid node,* $x = 500\ [m]$



***(b)*** *Step at boundary finite volume,* $x = 550\ [m]$

***Figure 3.6:*** *Step function approximated by a piecewise linear function (red line with dots at the nodes),* $c_\Psi = 0$, $\Delta x = 100\ [m]$ *and* $\Psi = 0$. *The thin green vertical lines indicate the borders of the finite volumes.*

As seen from Figure 3.6 the step is more taken into account as is presented in Figure 3.5. For $\Delta x = 100\,[\mathrm{m}]$ the absolute value of the second derivative is ...

***TODO 3.2:*** *Determine the value of the second derivative, Dx = 100 m and Dx = 50 m* ***TODO***

There are two options to estimate this step function by a piecewise linear smooth function with the same numerical accuracy:

1 Regularization with using a large grid size, the numerical solution is less close to the step function (see Figure 3.7)

2 Regularization with using a small grid size, the numerical solution is closer to the step function (see Figure 3.8),

Both options has the same value for $c_\Psi = 4$. Meaning that the step is represented by the same number of grid cells. How to estimate $c_\Psi$ can be read in Appendix A. It is up to the user which regularization is can be used for the numerical simulation. A better representation of the step function need a smaller grid size.

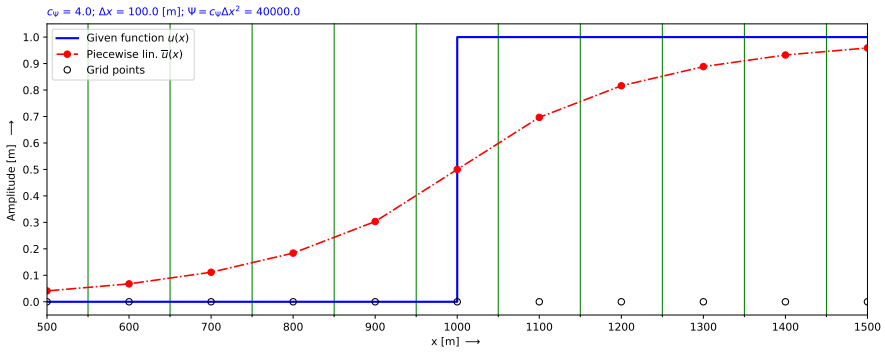



***Figure 3.7:*** *Step function approximated by a piecewise linear function (red line with dots at the nodes),* $c_\Psi = 4$, $\Delta x = 100\ [m]$ *and* $\Psi = 40000$. *The thin green vertical lines indicate the borders of the finite volumes.*

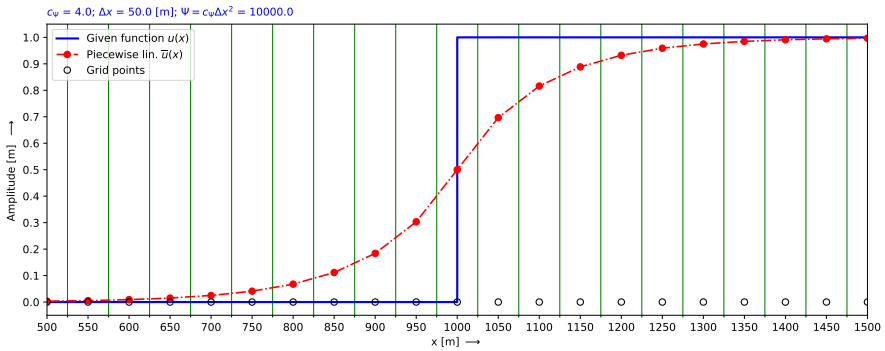



***Figure 3.8:*** *Step function approximated by a piecewise linear function (red line with dots at the nodes),* $c_\Psi = 4$, $\Delta x = 50\ [m]$ *and* $\Psi = 10000$. *he thin green vertical lines indicate the borders of the finite volumes.*

### 3.2.3 Small and a large gradient in the data set

To show the influence of regularization a more general data set is chosen, Borsboom (1998) (given function, here adjusted). A given function with a small (smooth) and large (steep) gradients in the data set is chosen. The function is defined by:

$$u_{given}(x) = \begin{cases} 10\left(\frac{1}{2} - \frac{1}{2}\tanh\left(20x/1000 - 6\right)\right), & \text{if} \quad 0\ [m] \leq x \leq 650\ [m], \\ 10, & \text{if} \quad 650\ [m] < x \geq 1000\ [m], \end{cases} \tag{3.29}$$

and shown in Figure 3.9:

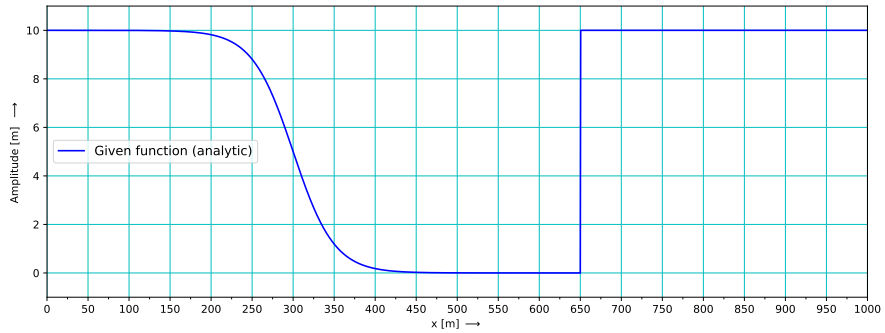
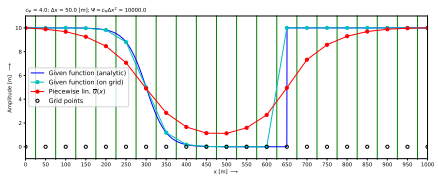
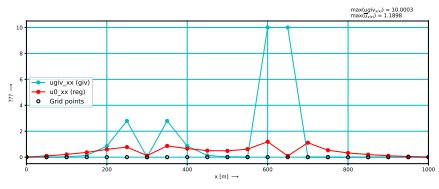



***Figure 3.9:*** *Large and small gradients in given function.*

First guess of regularization:




***(a)*** *Grid size* $\Delta x = 50\ [m]$




***(b)*** *Second derivatives, normalized grid size* $\Delta\xi = 1$.

***Figure 3.10:*** *Initial guess* ($c_\Psi = 4, \Delta x = 50\ [m]$)

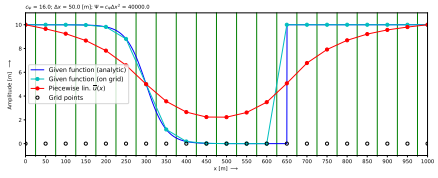
There are two options to adjust the data:

1 increase the regularization coefficient or

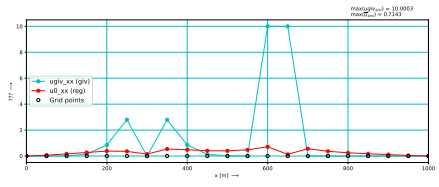
2 choose a smaller grid size.

## *Regularization coefficient increased*

Increasing the regularization coefficient $c_\Psi$:




***(a)*** *Grid size* $\Delta x = 50\ [m]$

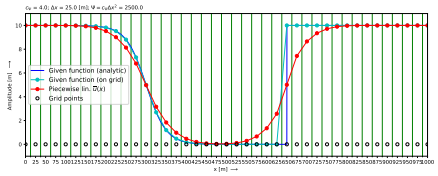



***(b)*** *Second derivatives, normalized grid size* $\Delta\xi = 1$.

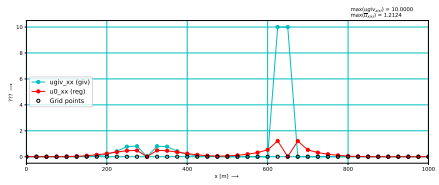
***Figure 3.11:*** *Regularization coefficient increased to 16* ($c_\Psi = 16, \Delta x = 50\ [m]$)

## *Grid size decreased*

Decrease the grid size $\Delta x$:




***(a)*** *Step at grid node,* $c_\Psi = 4, \Delta x = 25\ [m]$




***(b)*** *Second derivatives.*

***Figure 3.12:*** *Grid size decreased to* $25\ [m]$ ($c_\Psi = 4, \Delta x = 25\ [m]$)

# 4 Time integration scheme

To derive a time integration we start from the PDE:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{f}(\boldsymbol{u}) = 0 \tag{4.1}$$

For conservation types it can be written as:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{f}(\boldsymbol{u}) = \mathbf{S} \tag{4.2}$$

and when the finite volume approach is applied, we get

$$\int_\Omega \frac{\partial \boldsymbol{u}}{\partial t}\, d\Omega + \int_\Omega \nabla \cdot \boldsymbol{f}(\boldsymbol{u})\, d\Omega = \int_\Omega \mathbf{S}\, d\Omega, \tag{4.3}$$

and after Green's theorem

$$\int_\Omega \frac{\partial \boldsymbol{u}}{\partial t}\, d\Omega + \int_\Gamma \boldsymbol{f}(\boldsymbol{u}) \cdot \boldsymbol{n}\, d\Gamma = \int_\Omega \mathbf{S}\, d\Omega, \tag{4.4}$$

## 4.1 Fully implicit time integration by adding an iteration process

The system of Equations (4.1) can be written as, including the $\theta$ method:

$$\Delta t_{inv} \mathbf{M} \left( \boldsymbol{u}^{n+1} - \boldsymbol{u}^{n} \right) + \boldsymbol{f}\left( \boldsymbol{u}^{n+\theta} \right) = 0 \tag{4.5}$$

with $\Delta t_{inv} = 1/\Delta t$, $\mathbf{M}$ a mass-matrix and $0 \leq \theta \leq 1$.

To reach a fully implicit time integration an iteration process $p$ is added (Borsboom, 2019, eqs. 15/16):

$$\Delta t_{inv} \mathbf{M} \left( \boldsymbol{u}^{n+1,p+1} - \boldsymbol{u}^{n} \right) + \boldsymbol{f}\left( \boldsymbol{u}^{n+\theta,p+1} \right) = 0 \tag{4.6}$$

iterating from $p \rightarrow p+1$ until convergence.

The **first** term is split to get a so called "Delta" formulation, taking into account the previous iteration:

$$\Delta t_{inv} \mathbf{M} \left( \boldsymbol{u}^{n+1,p+1} - \boldsymbol{u}^{n+1,p} + \boldsymbol{u}^{n+1,p} - \boldsymbol{u}^{n} \right) + \cdots = 0 \tag{4.7}$$

$$\Delta t_{inv} \mathbf{M} \Delta \boldsymbol{u}^{n+1,p+1} + \cdots = -\Delta t_{inv} \mathbf{M} \left( \boldsymbol{u}^{n+1,p} - \boldsymbol{u}^{n} \right) \tag{4.8}$$

with $\Delta \boldsymbol{u}^{n+1,p+1} = \boldsymbol{u}^{n+1,p+1} - \boldsymbol{u}^{n+1,p}$. The right hand side is fully explicit (i.e. known at the previous iteration level $p$). And if the iteration process is converged, the term at the left hand side is zero (i.e. $\Delta \boldsymbol{u}^{n+1,p+1} = 0$) so the right handside represent the time derivative.

The **second** term of Equation (4.5)

$$\ldots + \boldsymbol{f}\left(\boldsymbol{u}^{n+\theta,p+1}\right) = 0 \tag{4.9}$$

will be linearized around the iteration step $p$ (Newton linearization) and yields

$$\boldsymbol{f}(\boldsymbol{u}^{n+\theta,p+1}) = \boldsymbol{f}(\boldsymbol{u}^{n+\theta,p}) + \frac{\partial \boldsymbol{f}(\boldsymbol{u}^{n+\theta,p})}{\partial \boldsymbol{u}^{n+1,p}}(\boldsymbol{u}^{n+\theta,p+1} - \boldsymbol{u}^{n+\theta,p}) \tag{4.10}$$

$$= \boldsymbol{f}(\boldsymbol{u}^{n+\theta,p}) + \frac{\partial \boldsymbol{f}(\boldsymbol{u}^{n+\theta,p})}{\partial \boldsymbol{u}^{n+1,p}}\Delta\boldsymbol{u}^{n+\theta,p+1} \tag{4.11}$$

with

$$\Delta\boldsymbol{u}^{n+\theta,p+1} = \boldsymbol{u}^{n+\theta,p+1} - \boldsymbol{u}^{n+\theta,p} \tag{4.12}$$

$$= \theta\boldsymbol{u}^{n+1,p+1} + (1-\theta)\boldsymbol{u}^{n} - \theta\boldsymbol{u}^{n+1,p} - (1-\theta)\boldsymbol{u}^{n} \tag{4.13}$$

$$= \theta\boldsymbol{u}^{n+1,p+1} - \theta\boldsymbol{u}^{n+1,p} \tag{4.14}$$

$$= \theta\Delta\boldsymbol{u}^{n+1,p+1} \tag{4.15}$$

After subtitution of Equation (4.15) into Equation (4.11) we get:

$$\boldsymbol{f}(\boldsymbol{u}^{n+\theta,p+1}) = \boldsymbol{f}(\boldsymbol{u}^{n+\theta,p}) + \theta\frac{\partial \boldsymbol{f}(\boldsymbol{u}^{n+\theta,p})}{\partial \boldsymbol{u}^{n+1,p}}\Delta\boldsymbol{u}^{n+1,p+1} \tag{4.16}$$

The Jacobian

$$\mathbf{J}^{n+1,p} = \frac{\partial \boldsymbol{f}(\boldsymbol{u}^{n+\theta,p})}{\partial \boldsymbol{u}^{n+1,p}} = \frac{\partial \boldsymbol{f}(\theta\boldsymbol{u}^{\boldsymbol{n+1,p}} + (1-\theta)\boldsymbol{u}^{n})}{\partial \boldsymbol{u}^{n+1,p}} \tag{4.17}$$

is the approximate linearization of $\boldsymbol{f}$ as a function of $\theta\boldsymbol{u}^{n+1} + (1-\theta)\boldsymbol{u}^{n}$ with respect to $\boldsymbol{u}^{n+1,p}$. The Jacobians needed for the shallow water equations are described in section 4.2.

The total time integration method read:

$$\boxed{\begin{aligned}&\left(\Delta t_{inv}\mathbf{M} + \theta\mathbf{J}^{n+1,p}\right)\Delta\mathbf{u}^{n+1,p+1} = \\ &\quad = -\left(\Delta t_{inv}\mathbf{M}\left(\boldsymbol{u}^{n+1,p} - \boldsymbol{u}^{n}\right) + \boldsymbol{f}\left(\theta\boldsymbol{u}^{n+1,p} + (1-\theta)\boldsymbol{u}^{n}\right)\right)\end{aligned}} \tag{4.18}$$

with $\boldsymbol{u}^{n+1,p+1} = \boldsymbol{u}^{n+1,p} + \Delta\boldsymbol{u}^{n+1,p+1}$ and right hand side is explicit w.r.t. the iterator $p$. In case the Newton iteration process converges, i.e.:

$$\lim_{p\to\infty}\left(\Delta\boldsymbol{u}^{n+1,p+1}\right) = \lim_{p\to\infty}\left(\boldsymbol{u}^{n+1,p+1} - \boldsymbol{u}^{n+1,p}\right) = 0. \tag{4.19}$$

then the left hand side of Equation (4.18) is equal to zero and thus it solves the original system of equations:

$$0 = \Delta t_{inv}\mathbf{M}\left(\boldsymbol{u}^{n+1,p} - \boldsymbol{u}^{n}\right) + \boldsymbol{f}\left(\theta\boldsymbol{u}^{n+1,p} + (1-\theta)\boldsymbol{u}^{n}\right). \tag{4.20}$$

Because in the previous part we consider only the first derivative (Jacobian) and assumed that the second derivative is nearly zero, which is not always through. Therefor we will extend the iteration process, see section 4.1.1. See also: Borsboom (1998) and Pulliam (2014)

### 4.1.1 Pseudo time stepping

In section 4.1 we assumed that only the Jacobian is relevant and the second derivative is negligible. But in some case it is not the cases we have to assure that the following inequality is true:

$$\left|\frac{1}{2}\frac{\partial^2 \boldsymbol{f}(\theta \boldsymbol{u}^{n+1,p} + (1-\theta)\boldsymbol{u}^n)}{(\partial \boldsymbol{u}^{n+1,p})^2}\Delta \boldsymbol{u}^{n+1,p+1}\right| < O\left(\left|\frac{\partial \boldsymbol{f}(\theta \boldsymbol{u}^{n+1,p} + (1-\theta)\boldsymbol{u}^n)}{\partial \boldsymbol{u}^{n+1,p}}\right|\right) \tag{4.21}$$

Therefor the time integration is extended with a (so called) pseudo timestep method, which read:

$$\boxed{\begin{aligned}&\left(\mathbf{M}_{pseu}\boldsymbol{T}_{pseu}^{n+1,p} + \Delta t_{inv}\mathbf{M} + \theta \mathbf{J}^{n+1,p}\right)\Delta \mathbf{u}^{n+1,p+1} = \\ &\quad = -\left(\Delta t_{inv}\mathbf{M}\left(\boldsymbol{u}^{n+1,p} - \boldsymbol{u}^n\right) + \boldsymbol{f}\left(\theta \boldsymbol{u}^{n+1,p} + (1-\theta)\boldsymbol{u}^n\right)\right)\end{aligned}} \tag{4.22}$$

where $\boldsymbol{T}_{pseu}^{n+1,p}$ is a vector containing the inverse of the pseudo timestep, which may vary for all grid nodes, and $\mathbf{M}_{pseu}$ a mass-matrix operating on the pseudo timestep vector. See for a more detailed description and how to choose the term $\mathbf{M}_{pseu}\boldsymbol{T}_{pseu}^{n+1,p}$ Borsboom (2019) and Buijs (2024).

## 4.2 Jacobians

As seen in section 4.1 Jacobians need to be computed. These Jacobians does not contain only derivatives to the major variables but also to place derivatives, which need special attention (section 4.2.3). For example for the two dimensional convection flux ($\boldsymbol{q}\boldsymbol{q}^T$) and presure term $gh\nabla\zeta$, where $\boldsymbol{q} = (q, r)^T$ and $\zeta$ the water level.

As example we take the integral form of the two dimensional non-linear wave equation. This equation reads:

$$\int_\Omega \frac{\partial \boldsymbol{u}}{\partial t}\, d\Omega + \int_\Omega \nabla \bullet \boldsymbol{F}\, d\Omega = 0 \Leftrightarrow \int_\Omega \frac{\partial \boldsymbol{u}}{\partial t}\, d\Omega + \oint_\Gamma \boldsymbol{F} \bullet \boldsymbol{n}\, d\Gamma = 0 \tag{4.23}$$

with vector $\boldsymbol{u} = (h, q, r)^T$ the Jacobian read. With

| | |
|---|---|
| $h$ | The total water depth, [m] |
| $q$ | The water flux in $x$-direction, [$\mathrm{m^2\,s^{-1}}$] |
| $r$ | The water flux in $y$-direction, [$\mathrm{m^2\,s^{-1}}$] |

The Jacobian of the function $F(h, q, r)$ as used in the two dimensional shallow

water equations reads:

$$\mathbf{J} = \begin{pmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{pmatrix} = \begin{pmatrix} \frac{\partial F_1}{\partial h} & \frac{\partial F_1}{\partial q} & \frac{\partial F_1}{\partial r} \\ \frac{\partial F_2}{\partial h} & \frac{\partial F_2}{\partial q} & \frac{\partial F_2}{\partial r} \\ \frac{\partial F_3}{\partial h} & \frac{\partial F_3}{\partial q} & \frac{\partial F_3}{\partial r} \end{pmatrix} \tag{4.24}$$

### 4.2.1 Non-linear term, product

If the Jacobian contains non-linear terms, each variable of $\boldsymbol{u}$ is linearized before using it in the non-linear term. As an example a product of two quantities is taken, say $q$ and $r$ (like the convection term in two dimensional shallow water equations) and $\Delta q^{n+1,p+1} = \Delta q$ and $\Delta r^{n+1,p+1} = \Delta r$:

$$(qr)\big|^{n+\theta,p+1} = \left(q^{n+\theta,p} + \theta\Delta q\right)\left(r^{n+\theta,p} + \theta\Delta r\right) \approx \tag{4.25}$$

$$\approx q^{n+\theta,p}r^{n+\theta,p} + \theta q^{n+\theta,p}\Delta r + \theta r^{n+\theta,p}\Delta q \tag{4.26}$$

omitting the quadratic term $O((\Delta q)^2, \Delta q\Delta r, (\Delta r)^2)$.

*Jacobian*

When the Jacobian notation is used for the function $F(q,r) = qr$

$$\mathbf{J} = \begin{pmatrix} J_{11} & J_{12} \end{pmatrix} = \begin{pmatrix} \frac{\partial F}{\partial q} & \frac{\partial F}{\partial r} \end{pmatrix} = \begin{pmatrix} \frac{\partial qr}{\partial q} & \frac{\partial qr}{\partial r} \end{pmatrix} = \begin{pmatrix} r & q \end{pmatrix} \tag{4.27}$$

it leads to following approximation

$$(qr)\big|^{n+\theta,p+1} \approx q^{n+\theta,p}r^{n+\theta,p} + \theta J_{11}^{n+\theta,p}\Delta q + \theta J_{12}^{n+\theta,p}\Delta r = \tag{4.28}$$

$$= q^{n+\theta,p}r^{n+\theta,p} + \theta r^{n+\theta,p}\Delta q + q^{n+\theta,p}\Delta r \tag{4.29}$$

### 4.2.2 Non-linear term, quotient

If the Jacobian contains non-linear terms, each variable of $\boldsymbol{u}$ is linearized before using it in the non-linear term. in this example a quotient of two quantities is taken, say $q$ and $h$ (representing the velocity in the two dimensional shallow water equations) and $\Delta q^{n+1,p+1} = \Delta q$ and $\Delta h^{n+1,p+1} = \Delta h$:

$$\left(\frac{q}{h}\right)\Big|^{n+\theta,p+1} = \frac{q^{n+\theta,p} + \theta\Delta q}{h^{n+\theta,p} + \theta\Delta h} \approx \tag{4.30}$$

$$\approx \frac{q^{n+\theta,p} + \theta\Delta q}{h^{n+\theta,p}}\left(1 - \frac{\theta}{h^{n+\theta,p}}\Delta h + O\left((\Delta h)^2\right)\right) \approx \tag{4.31}$$

$$\approx \left(\frac{q^{n+\theta,p}}{h^{n+\theta,p}} + \frac{\theta}{h^{n+\theta,p}}\Delta q\right)\left(1 - \frac{\theta}{h^{n+\theta,p}}\Delta h + O\left((\Delta h)^2\right)\right) \approx \tag{4.32}$$

$$\approx \frac{q^{n+\theta,p}}{h^{n+\theta,p}} - \theta\frac{q^{n+\theta,p}}{(h^{n+\theta,p})^2}\Delta h + \theta\frac{1}{h^{n+\theta,p}}\Delta q \tag{4.33}$$

omitting the quadratic term $O((\Delta q)^2, \Delta q\Delta h, (\Delta h)^2)$.

*Jacobian*

When the Jacobian notation is used for the function $F(h,q) = q/h$

$$\mathbf{J} = \begin{pmatrix} J_{11} & J_{12} \end{pmatrix} = \begin{pmatrix} \frac{\partial F}{\partial h} & \frac{\partial F}{\partial q} \end{pmatrix} = \begin{pmatrix} \frac{\partial q/h}{\partial h} & \frac{\partial q/h}{\partial q} \end{pmatrix} = \begin{pmatrix} -\frac{q}{h^2} & \frac{1}{h} \end{pmatrix} \tag{4.34}$$

it leads to following approximation

$$(qr)|^{n+\theta,p+1} \approx \frac{q^{n+\theta,p}}{r^{n+\theta,p}} + \theta J_{11}^{n+\theta,p}\Delta h + \theta J_{12}^{n+\theta,p}\Delta q = \tag{4.35}$$

$$= q^{n+\theta,p}r^{n+\theta,p} - \theta\frac{q^{n+\theta,p}}{(h^{n+\theta,p})^2}\Delta h + \theta\frac{1}{h^{n+\theta,p}}\Delta q \tag{4.36}$$

### 4.2.3 Terms with an operator

The Jacobian of an operator is also applied to the argument of the operator. As an example the pressure term of the shallow water equations is taken, where $\Delta h^{n+1,p+1} = \Delta h$ and $\Delta\zeta^{n+1,p+1} = \Delta\zeta$:

$$gh\frac{\partial\zeta}{\partial x}\bigg|^{n+\theta,p+1} \approx g\left(h^{n+\theta,p} + \theta\Delta h\right)\frac{\partial}{\partial x}\left(\zeta^{n+\theta,p} + \theta\Delta\zeta\right) \approx \tag{4.37}$$

$$\approx gh^{n+\theta,p}\frac{\partial\zeta^{n+\theta,p}}{\partial x} + \theta gh^{n+\theta,p}\frac{\partial\Delta\zeta}{\partial x} + \theta g\frac{\partial\zeta^{n+\theta,p}}{\partial x}\Delta h \tag{4.38}$$

omitting the quadratic term $O(\Delta h\ \partial\Delta\zeta/\partial x)$. The term $\partial\Delta\zeta/\partial x$ is not always small and therefor a psuedo time step method is introduced, see section 4.1.1. In discrete form it reads on location $x_{i+\frac{1}{2}}$:

$$gh\frac{\partial\zeta}{\partial x}\bigg|_{i+\frac{1}{2}}^{n+\theta,p+1} \approx gh_{i+\frac{1}{2}}^{n+\theta,p}\frac{\zeta_{i+1}^{n+\theta,p} - \zeta_i^{n+\theta,p}}{\Delta x_i} + \theta gh_{n+\frac{1}{2}}^{n+\theta,p}\frac{\Delta\zeta_{i+1} - \Delta\zeta_i}{\Delta x_i} + \tag{4.39}$$

$$+ \theta g\frac{\zeta_{i+1}^{n+\theta,p} - \zeta_i^{n+\theta,p}}{\Delta x_i}\Delta h_{i+\frac{1}{2}} \tag{4.40}$$

Remember that the gradient over a grid cell $\Delta x_i$ is constant, due to the piece-wise linear approximation between two nodes.

*Jacobian*

When the Jacobian notation is used for the function $F(q,h) = gh\,\partial\zeta/\partial x$

$$\mathbf{J} = \begin{pmatrix} J_{11} & J_{12} \end{pmatrix} = \begin{pmatrix} \frac{\partial F}{\partial h} & \frac{\partial F}{\partial \zeta} \end{pmatrix} = \begin{pmatrix} \frac{\partial (gh\,\partial\zeta/\partial x)}{\partial h} & \frac{\partial (gh\,\partial\zeta/\partial x)}{\partial \zeta} \end{pmatrix} = \begin{pmatrix} g\,\frac{\partial \zeta}{\partial x} & gh\,\frac{\partial}{\partial x} \end{pmatrix} \tag{4.41}$$

it leads to following approximation

$$gh\frac{\partial \zeta}{\partial x}\bigg|_{i+\frac{1}{2}}^{n+\theta,p+1} \approx gh^{n+\theta,p}\frac{\partial \zeta^{n+\theta,p}}{\partial x} + \theta J_{11}\Delta h + \theta J_{12}\frac{\partial}{\partial x}(\Delta\zeta) \approx \tag{4.42}$$

$$\approx gh^{n+\theta,p}\frac{\zeta_{i+1}^{n+\theta,p} - \zeta_i^{n+\theta,p}}{\Delta x_i} + \theta g\frac{\zeta_{i+1}^{n+\theta,p} - \zeta_i^{n+\theta,p}}{\Delta x_i}\Delta h_{i+\frac{1}{2}} + \tag{4.43}$$

$$+ \theta g h_{n+\frac{1}{2}}^{n+\theta,p}\frac{\Delta\zeta_{i+1} - \Delta\zeta_i}{\Delta x_i} \tag{4.44}$$

# 5 Towards the shallow water equations

In this document we will end up with an implementation description of the 1D shallow water equation. We start a zero-dimensional implementation of a source term, representing the source and sink of external influences, like a power plant. Here we will show the results of a Brusselator (Ault and Holmgreen, 2003) and of a air pollution model (Hundsdorfer and Verwer, 2003, ex. 1.1 pg. 7). Then we continue with the one dimensional advection/transport equation than a one dimensional wave equation without convection, then with convection and at last with a bottom friction term. in this sequence we are missing the viscosity term, that term will be investigated by the advection-diffusion equation.

Because we will handle the shallow water equations in the variables $h$ and $q$ and not in $\zeta$ and $u$ the equations does have always a non-linear behaviour, only for very small amplitude the behaviour is like a linear system. For linear wave equations the behaviour is always linear, even for large amplitudes, which is not the case for the equations we consider.

## 5.1 0-D Source/sink term

In this section a zero-dimensional model is implementation of the source term, representing the source and sink of external influences, like a power plant. The main (simple) equation will look like:

$$\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{f}(\boldsymbol{u}, t) \tag{5.1}$$

Here we will show the results of an air pollution model, section 5.1.1 (Hundsdorfer and Verwer, 2003, ex. 1.1 pg. 7) and a Brusselator, section 5.1.2 (Ault and Holmgreen, 2003).

### 5.1.1 Air pollution

### Analytic description

We illustrate the mass action law by the following three reactions between oxygen $O_2$, atomic oxygen $O$, nitrogen oxide $NO$, and nitrogen dioxide $NO_2$ (Hundsdorfer and Verwer, 2003, eq. 1.1, page 7):

$$NO_2 + h\nu \xrightarrow{k_1} NO + O, \tag{5.2}$$

$$O + O_2 \xrightarrow{k_2} O_3, \tag{5.3}$$

$$NO + O_3 \xrightarrow{k_3} O_2 + NO_2. \tag{5.4}$$

The corresponding ODE system reads:

$$\frac{\partial u_1}{\partial t} = k_1 u_3 - k_2 u_1 \tag{5.5}$$

$$\frac{\partial u_2}{\partial t} = k_1 u_3 - k_3 u_2 u_4 + \sigma_2 \tag{5.6}$$

$$\frac{\partial u_3}{\partial t} = k_3 u_2 u_4 - k_1 u_3 \tag{5.7}$$

$$\frac{\partial u_4}{\partial t} = k_2 u_1 - k_3 u_2 u_4 \tag{5.8}$$

with $\boldsymbol{u}(0) = (0.0, 2.0 \times 10^{-1}, 2.0 \times 10^{-3}, 2.0 \times 10^{-1})^T$ and $\sigma_2 = 10^{-7}$, and the coefficients $k$ are defined as (the given conditions are different from the conditions as defined in Hundsdorfer and Verwer (2003, pg. 8):

$$k_1 = \begin{cases} 10^{-5} \exp\left(7\ g(t)\right) \\ 10^{-40}, \qquad \text{during night} \end{cases} \tag{5.9}$$

$$k_2 = 2.0 \times 10^{-2} \tag{5.10}$$

$$k_3 = 1.0 \times 10^{-3} \tag{5.11}$$

with

$$g(t) = \left(\sin\left(\frac{\pi}{16}(t_h - 4)\right)\right)^{0.2}, \qquad t_h = \frac{t}{3600}; \tag{5.12}$$

where $t_h$ is the time in hours. How these equations are discretized is given in Equation (5.1.1).

## Numerical discretization

The discretization in $\Delta$-formulation reads:

$$\frac{1}{\Delta t}\Delta u_1^{n+1,p+1} = -\frac{1}{\Delta t}(u_1^{n+1,p} - u_1^n) + k_1(u_3^{n+\theta,p+1}) - k_2(u_1^{n+\theta,p+1}) \tag{5.13}$$

$$\frac{1}{\Delta t}\Delta u_2^{n+1,p+1} = -\frac{1}{\Delta t}(u_2^{n+1,p} - u_2^n) + k_1(u_3^{n+\theta,p+1}) - k_3(u_2^{n+\theta,p+1})(u_4^{n+\theta,p+1}) + \sigma_2 \tag{5.14}$$

$$\frac{1}{\Delta t}\Delta u_3^{n+1,p+1} = -\frac{1}{\Delta t}(u_3^{n+1,p} - u_3^n) + k_3(u_2^{n+\theta,p+1})(u_4^{n+\theta,p+1}) - k_1(u_3^{n+\theta,p+1}) \tag{5.15}$$

$$\frac{1}{\Delta t}\Delta u_4^{n+1,p+1} = -\frac{1}{\Delta t}(u_4^{n+1,p} - u_4^n) + k_2(u_1^{n+\theta,p+1}) - k_3(u_2^{n+\theta,p+1})(u_4^{n+\theta,p+1}) \tag{5.16}$$

with linearization of $\boldsymbol{u}^{n+\theta,p+1}$ yields:

$$\begin{aligned}\frac{1}{\Delta t}\Delta u_1^{n+1,p+1} = &-\frac{1}{\Delta t}(u_1^{n+1,p} - u_1^n) + \\ &+ k_1(u_3^{n+\theta,p} + \theta\Delta u_3^{n+1,p+1}) - k_2(u_1^{n+\theta,p} + \theta\Delta u_1^{n+1,p+1})\end{aligned} \tag{5.17}$$

$$\begin{aligned}\frac{1}{\Delta t}\Delta u_2^{n+1,p+1} = &-\frac{1}{\Delta t}(u_2^{n+1,p} - u_2^n) + k_1(u_3^{n+\theta,p} + \theta\Delta u_3^{n+1,p+1}) + \\ &- k_3(u_2^{n+\theta,p} + \theta\Delta u_2^{n+1,p+1})(u_4^{n+\theta,p} + \theta\Delta u_4^{n+1,p+1}) + \sigma_2\end{aligned} \tag{5.18}$$

$$\begin{aligned}\frac{1}{\Delta t}\Delta u_3^{n+1,p+1} = &-\frac{1}{\Delta t}(u_3^{n+1,p} - u_3^n) + k_3(u_2^{n+\theta,p} + \theta\Delta u_2^{n+1,p+1})(u_4^{n+\theta,p} + \\ &+ \theta\Delta u_4^{n+1,p+1}) - k_1(u_3^{n+\theta,p} + \theta\Delta u_3^{n+1,p+1})\end{aligned} \tag{5.19}$$

$$\begin{aligned}\frac{1}{\Delta t}\Delta u_4^{n+1,p+1} = &-\frac{1}{\Delta t}(u_4^{n+1,p} - u_4^n) + k_2(u_1^{n+\theta,p} + \theta\Delta u_1^{n+1,p+1}) + \\ &- k_3(u_2^{n+\theta,p} + \theta\Delta u_2^{n+1,p+1})(u_4^{n+\theta,p} + \theta\Delta u_4^{n+1,p+1})\end{aligned} \tag{5.20}$$

and rearrange the system of equations to $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, yields

$$\begin{aligned}\frac{1}{\Delta t}\Delta u_1^{n+1,p+1} - k_1\theta\Delta u_3^{n+1,p+1} + k_2\theta\Delta u_1^{n+1,p+1} = \\ = -\frac{1}{\Delta t}(u_1^{n+1,p} - u_1^n) + k_1 u_3^{n+\theta,p} - k_2 u_1^{n+\theta,p}\end{aligned} \tag{5.21}$$

$$\begin{aligned}\frac{1}{\Delta t}\Delta u_2^{n+1,p+1} - k_1\theta\Delta u_3^{n+1,p+1} + k_3\theta u_4^{n+1,p}\Delta u_2^{n+1,p+1} + k_3\theta u_2^{n+1,p}\Delta u_4^{n+1,p+1} = \\ = -\frac{1}{\Delta t}(u_2^{n+1,p} - u_2^n) + k_1 u_3^{n+\theta,p} - k_3 u_2^{n+\theta,p}u_4^{n+\theta,p} + \sigma_2\end{aligned} \tag{5.22}$$

$$\begin{aligned}\frac{1}{\Delta t}\Delta u_3^{n+1,p+1} - k_3 u_2^{n+\theta,p}\theta\Delta u_4^{n+1,p+1} - k_3 u_4^{n+\theta,p}\theta\Delta u_2^{n+1,p+1} + k_1\theta\Delta u_3^{n+1,p+1} = \\ = -\frac{1}{\Delta t}(u_3^{n+1,p} - u_3^n) + k_3 u_2^{n+\theta,p}u_4^{n+\theta,p} - k_1 u_3^{n+\theta,p}\end{aligned} \tag{5.23}$$

$$\frac{1}{\Delta t}\Delta u_4^{n+1,p+1} - k_2\theta\Delta u_1^{n+1,p+1} + k_3 u_2^{n+\theta,p}\theta\Delta u_4^{n+1,p+1} + k_3 u_4^{n+\theta,p}\theta\Delta u_2^{n+1,p+1} =$$
$$= -\frac{1}{\Delta t}(u_4^{n+1,p} - u_4^n) + k_2 u_1^{n+\theta,p} - k_3 u_2^{n+\theta,p} u_4^{n+\theta,p} \tag{5.24}$$

### 5.1.2 Brusselator

#### Analytic description

The ODE system for the Brusselator reads Ault and Holmgreen (2003, eq. 14, 15):

$$\frac{\partial u_1}{\partial t} = 1 - (k_2 + 1)u_1 + k_1 u_1^2 u_2, \tag{5.25}$$
$$\frac{\partial u_2}{\partial t} = k_2 u_1 - k_1 u_1^2 u_2 \tag{5.26}$$

with $k_1 = 1$ and $k_2 = 2.5$ and initial values $u_1(0) = 0$ and $u_2(0) = 0$.

#### Numerical discretization

The ODE system for the brusselator reads (Ault and Holmgreen, 2003, eq. 14, 15):

$$\frac{\partial u_1}{\partial t} = 1 - (k_2 + 1)u_1 + k_1 u_1^2 u_2, \tag{5.27}$$
$$\frac{\partial u_2}{\partial t} = k_2 u_1 - k_1 u_1^2 u_2 \tag{5.28}$$

The discretization in $\Delta$-formulation reads:

$$\frac{1}{\Delta t}\Delta u_1^{n+1,p+1} = -\frac{1}{\Delta t}(u_1^{n+1,p} - u_1^n) + 1 - (k_2 + 1)u_1^{n+\theta,p+1} +$$
$$+ k_1 \left(u_1^{n+\theta,p+1}\right)^2 u_2^{n+\theta,p+1} \tag{5.29}$$
$$\frac{1}{\Delta t}\Delta u_2^{n+1,p+1} = -\frac{1}{\Delta t}(u_2^{n+1,p} - u_2^n) + k_2(u_1^{n+\theta,p+1}) +$$
$$- k_1 \left(u_1^{n+\theta,p+1}\right)^2 u_2^{n+\theta,p+1} \tag{5.30}$$

with linearization of $\boldsymbol{u}^{n+\theta,p+1}$ yields:

$$\frac{1}{\Delta t}\Delta u_1^{n+1,p+1} = -\frac{1}{\Delta t}(u_1^{n+1,p} - u_1^n) + 1 - (k_2 + 1)\left(u_1^{n+\theta,p} + \theta\Delta u_1^{n+1,p+1}\right) +$$
$$+ k_1 \left(u_1^{n+\theta,p} + \Delta u_1^{n+1,p+1}\right)^2 \left(u_2^{n+\theta,p} + \Delta u_1^{n+1,p+1}\right) \tag{5.31}$$
$$\frac{1}{\Delta t}\Delta u_2^{n+1,p+1} = -\frac{1}{\Delta t}(u_2^{n+1,p} - u_2^n) + k_2(u_1^{n+\theta,p} + \theta\Delta u_1^{n+1,p+1}) +$$
$$- k_1 \left(u_1^{n+\theta,p} + \Delta u_1^{n+1,p+1}\right)^2 \left(u_2^{n+\theta,p} + \Delta u_1^{n+1,p+1}\right) \tag{5.32}$$

and rearrange the system of equations to $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ and omitting the second order terms, yields

$$\frac{1}{\Delta t}\Delta u_1^{n+1,p+1} - \theta\left((k_2+1) + 2k_1 u_1^{n+\theta,p} u_2^{n+\theta,p}\right)\Delta u_1^{n+1,p+1} - \theta k_1 \left(u_1^{n+1,p+1}\right)^2 \Delta u_2^{n+1,p+1} =$$
$$= -\frac{1}{\Delta t}(u_1^{n+1,p} - u_1^n) + 1 - (k_2+1)u_1^{n+\theta,p} + k_1\left(u_1^{n+\theta,p}\right)^2 u_2^{n+\theta,p} \qquad (5.33)$$

$$\frac{1}{\Delta t}\Delta u_2^{n+1,p+1} + \theta\left(k_2 + 2k_1 u_1^{n+\theta,p} u_2^{n+\theta,p}\right)\Delta u_1^{n+1,p+1} + \theta k_1 \left(u_1^{n+1,p+1}\right)^2 \Delta u_2^{n+1,p+1} =$$
$$= -\frac{1}{\Delta t}(u_2^{n+1,p} - u_2^n) + k_2 u_1^{n+\theta,p} - k_1\left(u_1^{n+\theta,p}\right)^2 u_2^{n+\theta,p} \qquad (5.34)$$

This system can be implemented and solved, some results are presented in section 5.1.2.

## 5.2 1-D Advection equation

The considered advection equation reads:

$$\frac{\partial c}{\partial t} + \frac{\partial uc}{\partial x} = 0, \qquad u > 0. \qquad (5.35)$$

A constituent $c$ is transported from the left to the right with velocity $u\,[\mathrm{m/s}]$. Which is discretised on the grid

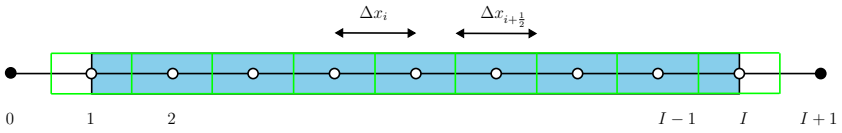



***Figure 5.1:*** *Water body (blue area), finite volumes (green boxes), computational points (open dots), virtual computational points (black dots), boundary points are at $x_1$ (inflow/west boundary) and $x_{I+\frac{1}{2}}$ (outflow/east boundary)*

An **essential** boundary condition at left side an inflow boundary is needed. And, at the right side an outflow boundary 'condition' is required for numerical reasons (called a **natural** boundary condition), i.e. a discretization of the model equation at the outflow boundary. The natural boundary conditions is fully determined by the outgoing signal and therefor we use Equation (B.10) for the outgoing signal. For the advection equation it reads:

$$\frac{\partial c}{\partial t} + \frac{\partial\, cu}{\partial x} = 0, \qquad u > 0. \qquad (5.36)$$

The **essential** boundary condition at the inflow boundary reads:

$$c(0,t) = c_0(t), \quad t > 0 \qquad \text{(essential boundary)} \qquad (5.37)$$

The essential boundary condition is supplied at $x_1$ with the following discretization (Equation (3.10))

$$\frac{1}{12}\Delta c_0^{n+1,p+1} + \frac{10}{12}\Delta c_1^{n+1,p+1} + \frac{1}{12}\Delta c_2^{n+1,p+1} = \tag{5.38}$$

$$= c_0(t) - \left(\frac{1}{12}c_0^{n+1,p} + \frac{10}{12}c_1^{n+1,p} + \frac{1}{12}c_2^{n+1,p}\right) \tag{5.39}$$

The **natural** is chosen in that way that as less as possible left going spurious numerical waves are generated at the outflow boundary, i.e. reflection. The natural boundary condition is supplied at $x_I$ with the discretization constants as determined by Equation (3.9) and boundary condition Equation (5.36) which yields:

$$\left(\frac{1+\alpha_{bc}}{\Delta t} + \theta\frac{u}{\Delta x}\right)\Delta c_{I+1}^{n+1,p+1} + \left(\frac{1-2\alpha_{bc}}{\Delta t} - \theta\frac{u}{\Delta x}\right)\Delta c_I^{n+1,p+1} + \frac{\alpha_{bc}}{\Delta t}\Delta c_{I-1}^{n+1,p+1} = \tag{5.40}$$

$$= -\left\{\frac{1+\alpha_{bc}}{\Delta t}\left(c_{I+1}^{n+1,p} - c_{I+1}^{n}\right) + \frac{1-2\alpha_{bc}}{\Delta t}\left(c_I^{n+1,p} - c_I^{n}\right) + \frac{\alpha_{bc}}{\Delta t}\left(c_{I-1}^{n+1,p} - c_{I-1}^{n}\right) + \right. \tag{5.41}$$

$$\left. + \frac{u}{\Delta x}\left(c_{I+1}^{n+\theta,p} - c_I^{n+\theta,p}\right)\right\} \tag{5.42}$$

where $\alpha_{bc} = 2\alpha - \frac{1}{2}$ ($\alpha_{bc} = 2\frac{1}{8} - \frac{1}{2} = -\frac{1}{4}$)

## 5.3 Diagonalise 1-D wave equation

The one dimensional shallow water equations with convection for flat bottom ($\frac{1}{2}g\,\partial h^2/\partial x = gh\,\partial h/\partial x$ and $\partial z_b/\partial x = 0$), reads

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0 \qquad \textit{continuity eq.} \tag{5.43}$$

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q^2}{h}\right) + gh\frac{\partial h}{\partial x} = 0 \qquad \textit{momentum eq.} \tag{5.44}$$

These one dimensional shallow water equations can be written in matrix and vector notation as:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \mathbf{A}\frac{\partial \boldsymbol{u}}{\partial x} = 0 \tag{5.45}$$

To find the characteristic equations this set of equations should be written in a set of equation representing left and right going waves. The diagonalisation is performed as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \mathbf{P}\underbrace{\mathbf{P^{-1}AP}}_{\mathbf{\Lambda}}\mathbf{P^{-1}}\frac{\partial \boldsymbol{u}}{\partial x} = 0 \tag{5.46}$$

multiply this with $\mathbf{P^{-1}}$

$$\mathbf{P^{-1}}\frac{\partial \boldsymbol{u}}{\partial t} + \mathbf{\Lambda P^{-1}}\frac{\partial \boldsymbol{u}}{\partial x} = 0 \tag{5.47}$$

with $\mathbf{\Lambda}$ a diagonal matrix and thus the left and right going signals are independent. For the one dimensional shallow water equations the two independent convection equations read:

$$\begin{pmatrix} \sqrt{gh} + \frac{q}{h} & -1 \\ \sqrt{gh} - \frac{q}{h} & 1 \end{pmatrix} \begin{pmatrix} \textit{continuity eq.} \\ \textit{momentum eq.} \end{pmatrix} = 0 \tag{5.48}$$

See for a derivation Appendix B.

Now we have split the wave equation into a right and left going signal now we are able to apply the **natural** boundary conditions as described in section 5.2 for each of the signals. The **essential** boundaries condition is chosen to be absorbing boundaries, so no reflections at the boundaries will appear.

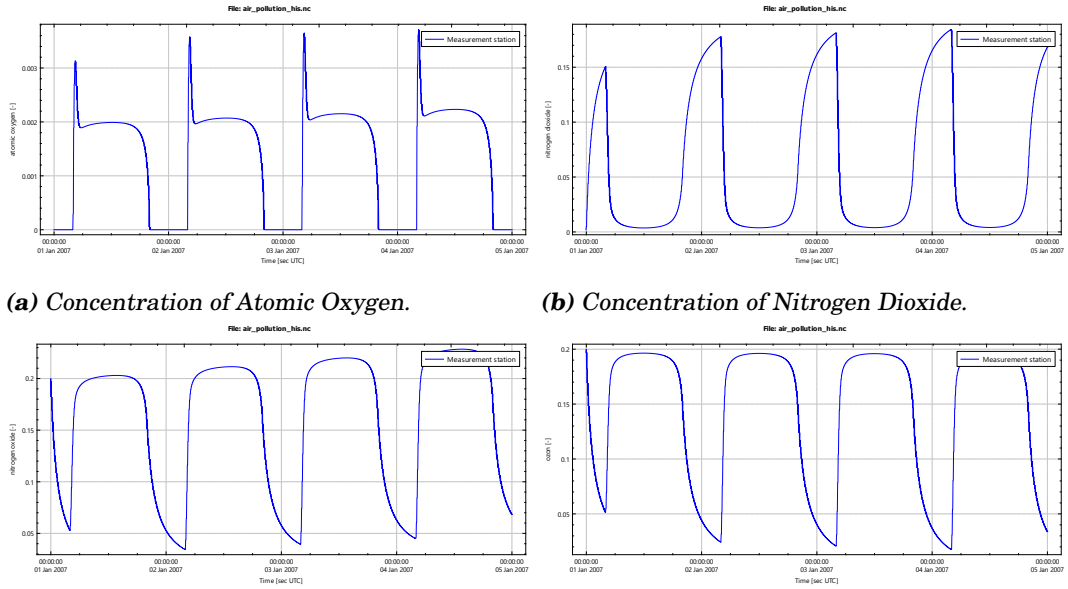
# 6 Numerical experiments

## 6.1 0-D, sources and sinks

In this section the time discretization is given for the air pollution example as described in section 5.1.1 and for the Brusselator, as described in section 5.1.2.

### 6.1.1 Air pollution

Some numerical results of the air pollution example (section 5.1.1) is:




***(a)*** *Concentration of Atomic Oxygen.*

***(b)*** *Concentration of Nitrogen Dioxide.*

***(c)*** *Concentration of Nitrogen Oxide.*

***(d)*** *Concentration of Ozon.*

***Figure 6.1:*** *Result plots of the different constituents, compute with the fully implicit time integration method with a timestep of 0.5 [s].*

#### Differen time integrators

Numerical stability for different values of $\Delta t$ are studied for Euler-explicit, Runga-Kutta-4 and the fully implicit $\Delta$-formulation.

***Table 6.1:*** *Stability for different time integrators*

| | Time step [s] | Euler explicit | Runge-Kutta 4 | Fully Implicit $\Delta$-formulation |
|---|---|---|---|---|
| **1** | 0.5 | - | ✓ | ✓ |
| **2** | 60 | ✓ | ✓ | ✓ |

| | Time step [s] | Euler Explicit | Runge-Kutta 4 | Fully Implicit $\Delta$-formulation |
|---|---|---|---|---|
| **3** | 120 | Unstable | ✓ | ✓ |
| **4** | 180 | - | Unstable | ✓ |
| **5** | 240 | - | - | ✓ |
| **6** | 300 | - | - | ✓ |
| **7** | 900 | - | - | ✓ |
| **8** | 1800 | - | - | ✓ |
| **9** | 3600 | - | - | ✓ |

### 6.1.2 Brusselator

Some numerical results of the brusselator example (section 5.1.2) is:

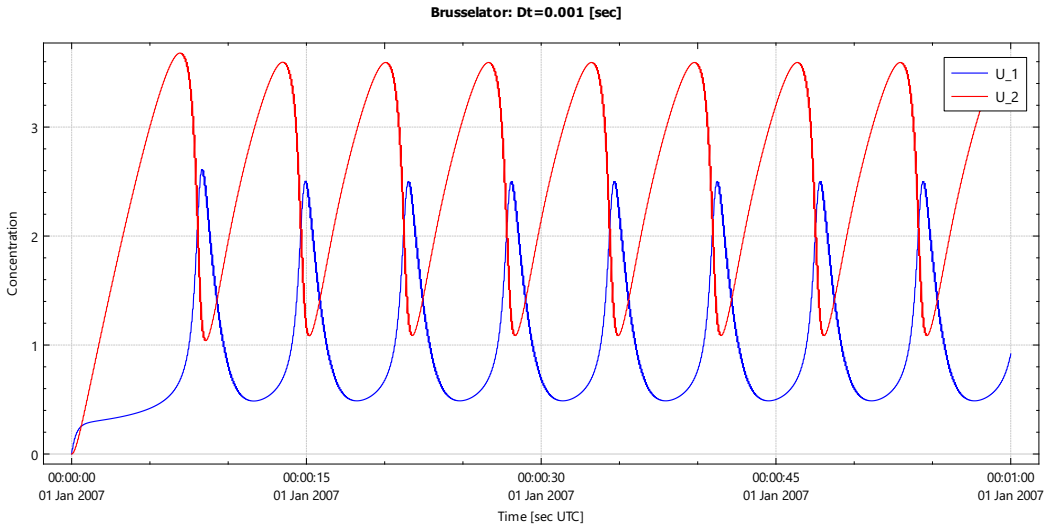
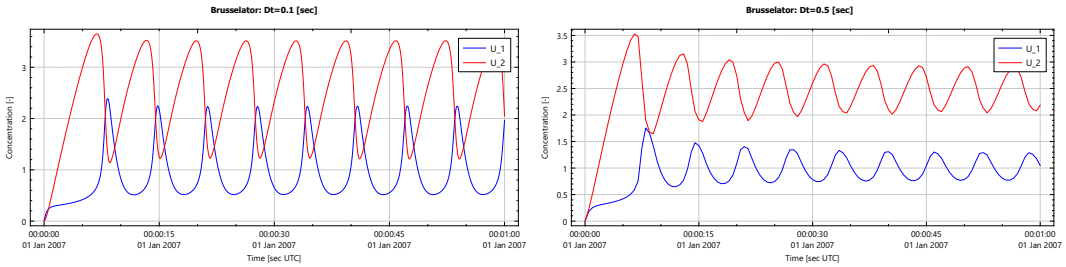



***Figure 6.2:*** *Fully Implicit:* $\Delta t = 0.001$, $k_1 = 1$, $k_2 = 2.5$.

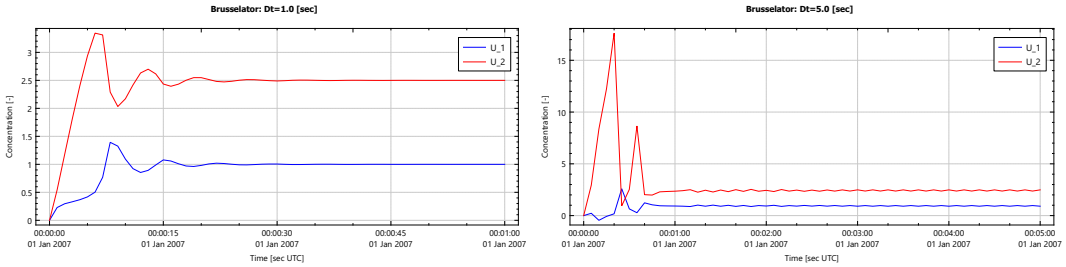
The solution presented in Figure 6.2 is assumed to be the reference (analytic) solution of Equation (5.28).




***(a)*** *Fully Implicit:* $\Delta t = 0.1$, $k_1 = 1$, $k_2 = 2.5$ ***(b)*** *Fully Implicit:* $\Delta t = 0.5$, $k_1 = 1$, $k_2 = 2.5$

***Figure 6.3:*** *Result plots for constant value of* $k_1 = 1$ *and* $k_2 = 2.5$*, computed with a fully implicit (*$\Delta$ *formulation) time integration method for different time steps* $\Delta t = 0.001, 0.1, 0.5$.

Extra attention is needed for the Fully Implicit time integration with larger time step:




***(a)*** *Fully Implicit:* $\Delta t = 1.0$, $k_1 = 1$, $k_2 = 2.5$ ***(b)*** *Fully Implicit:* $\Delta t = 5.0$, $k_1 = 1$, $k_2 = 2.5$

***Figure 6.4:*** *Result plots for constant value of* $k_1 = 1$ *and* $k_2 = 2.5$*, computed with a fully implicit (*$\Delta$ *formulation) time integration method for different time steps* $\Delta t = 1.0, 5.0$.

Figure 6.4a converge to the equilibrium state $(u_1, u_2) = (1.0, 2.5)$ and Figure 6.4b looks to converge to the equilibrium state $(u_1, u_2) = (1.0, 2.5)$ but is still wiggling after 5 min of simulation time (even after one day — not presented here). How these equations are discretized is given in Equation (5.1.2).

## Different time integrators

Numerical stability for different values of $\Delta t$ are studied for the Runga-Kutta-4 and fully implicit $\Delta$-formulation.

***Table 6.2:*** *Stability of different time integrators for the Brusselator.*

| | Time step [s] | Runge-Kutta 4 | Fully Implicit $\Delta$-formulation |
|---|---|---|---|
| **1** | 0.1 | ✓ | ✓ |

| | Time step [s] | Runge-Kutta 4 | Fully Implicit $\Delta$-formulation |
|---|---|---|---|
| **2** | 0.2 | ✓ | ✓ |
| **3** | 0.5 | ✓ | ✓ |
| **4** | 1.0 | Unstable | ✓ |
| **5** | 2.0 | | ✓ |
| **6** | 5.0 | | ✓ |

## 6.2 1-D Advection equation

The considered advection equation reads:

$$\frac{\partial c}{\partial t} + \frac{\partial uc}{\partial x} = 0, \tag{6.1}$$

With initial condition for the velocity $u_{given} = 10\,\mathrm{m\,s^{-1}}$, which coincide with the wave celerity in the next numerical experiments.

$$u(x,0) = u_{given} \tag{6.2}$$

and with a prescribed boundary condition at the left side of the domain

$$c(0,t) = c_{given} \begin{cases} \frac{1}{2}\cos\left(\pi\left(\frac{t_{reg}-t}{t_{reg}}\right)+1\right) & \text{if } t < t_{reg}, \\ 1 & \text{if } t \geq t_{reg}, \end{cases} \tag{6.3}$$

where

$t_{reg}$ The regularization time for the given time-series, [s]

### Numerical experiment

The numerical experiment is performed with the following parameters:

- Length of the domain, $12\,000\,\mathrm{m}$
- Grid size, $\Delta x = 10\,\mathrm{m}$
- Start time, $t_{start} = 0\,\mathrm{s}$
- End time, $t_{stop} = 3600\,\mathrm{s}$
- Timestep, $\Delta t = 5\,\mathrm{s}$
- Regularization time for time-series, $t_{reg} = 300\,\mathrm{s}$
- Prescribed constant velocity, $u_{given}(x,t) = 10\,\mathrm{m\,s^{-1}}$
- Prescribed initial value of the constituent, $c(x,0) = 0\,[-]$.
- Prescribed constant constituent on boundary, $c_{given}(0,t) = 1\,[-]$

### 6.2.1 1-D Advection equation + $\exp(\phi)$

Due to numerical discretization the value $c$ could become negative, in certain applications a positive value is reguired. To ensure the positivity of the constituent $c$ we will write the equation with variable $\phi$, where $\phi$ is defined as:

$$\phi = \ln(c) \tag{6.4}$$

The considered advection equation now reads:

$$\frac{\partial \phi}{\partial t} + \frac{\partial u\phi}{\partial x} = 0, \tag{6.5}$$

With initial condition for the velocity $u_{given} = 10\,\mathrm{m\,s^{-1}}$, which coincide with the wave celerity in the next numerical experiments.

$$u(x,0) = u_{given} \tag{6.6}$$

and with a prescribed boundary condition for the constituent $c$ at the left side of the domain

$$c(0,t) = c_{given} \begin{cases} \frac{1}{2}\cos\left(\pi\left(\frac{t_{reg}-t}{t_{reg}}\right)+1\right) & \text{if } t < t_{reg}, \\ 1 & \text{if } t \geq t_{reg}, \end{cases} \tag{6.7}$$

where

$t_{reg}$ The regularization time for the given time-series, [s]

A value of $c(0,t) = 0$ is estimated by, so:

$$\phi(0,t) = \ln(c(0,t)) \gtrsim -50 \tag{6.8}$$

#### Numerical experiment

The numerical experiment is performed with the following parameters:

- Length of the domain, $12\,000\,\mathrm{m}$
- Grid size, $\Delta x = 10\,\mathrm{m}$
- Start time, $t_{start} = 0\,\mathrm{s}$
- End time, $t_{stop} = 3600\,\mathrm{s}$
- Timestep, $\Delta t = 5\,\mathrm{s}$
- Regularization time for time-series, $t_{reg} = 300\,\mathrm{s}$
- Prescribed constant velocity, $u_{given}(x,t) = 10\,\mathrm{m\,s^{-1}}$
- Prescribed initial value of the constituent, $c(x,0) = 1 \times 10^{-20}$, so $\phi = -50\,[-]$.
- Prescribed constant constituent on boundary $c_{given}(0,t) = 1$, so $\phi = \ln(c_{given}(0,t)) = \ln(1) = 0\,[-]$

## 6.3 1-D Wave equation

The considered 1-D wave equation reads:

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0 \qquad \textit{continuity eq.} \tag{6.9}$$

$$\frac{\partial q}{\partial t} + gh\frac{\partial h}{\partial x} = 0 \qquad \textit{momentum eq.} \tag{6.10}$$

With initial conditions

$$h(x,0) = \zeta(x,0) - z_b(x), \tag{6.11}$$

$$q(x,0) = 0 \tag{6.12}$$

for the the water level a Gaussian hump is prescribed

$$\zeta(x) = 2a_0 \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right), \quad [\mathrm{m}] \tag{6.13}$$

At the boundaries no ingoing signals are prescribed, so outgoing signals are leaving the domain unhampered, which means no reflections will be present other then numerical reflections.

The numerical experiment is performed with the following parameters:

- Length of the domain, $12\,000\,\mathrm{m}$, ranging from $-6000\,\mathrm{m}$ to $6000\,\mathrm{m}$.
- Bed level, $z_b = -10\,\mathrm{m}$.
- Grid size, $\Delta x = 10\,\mathrm{m}$.
- Start time, $t_{start} = 0\,\mathrm{s}$.
- End time, $t_{stop} = 3600\,\mathrm{s}$.
- Timestep, $\Delta t = 10\,\mathrm{s}$.
- Regularization time for time-series, $t_{reg} = 300\,\mathrm{s}$.
- Amplitude of the Gaussian hump at the boundary, $a_0 = 0.01\,\mathrm{m}$.
- Centre of the Gaussian hump, $\mu = 3000\,\mathrm{m}$.
- Spreading of the Gaussian hump, $\sigma = 700\,\mathrm{m}$.

And an experiment with given boundary values:

- Prescribed discharge boundary at $-6000\,\mathrm{m}$: $q(0,t) = 0.05\,\mathrm{m^2\,s^{-1}}$.
- Prescribed water level boundary value at $6000\,\mathrm{m}$: $\zeta(0,t) = 0.02\,\mathrm{m}$.

Because the boundary values are constant in time the solution is time-independent. Therefore two simulation should be performed:

1 A stationary computation (with $\Delta t = 0\,\mathrm{s}$) and
2 an temporal computation (with $t_{stop} = 10\,800\,\mathrm{s}$)

# 7 2-D Shallow water equations

## 7.1 Space discretisation

### 7.1.1 Space discretisation, structured

### 7.1.2 Space discretisation, unstructured

# References


Ault, Shaun and Erik Holmgreen (2003). *Dynamics of the Brusselator*. URL: https://mate.unipv.it/~boffi/teaching/download/Brusselator.pdf.

Borsboom, M. (1998). "Development of an error-minimizing adative grid method". In: *Applied numerical mathematical* 26, pp. 13–21.

Borsboom, M. (2001). "Development of a 1-D Error-Minimizing Moving Adaptive Grid Method". In: ed. by B.Schiesser A. van de Wouwer P. Saucez. CRC Press, to be published. Chap. Adaptive Method of Lines. DOI: https://dx.doi.org/10.1201/9781420035612-8.

Borsboom, M. (2003). *To smooth or not to smooth*. Memo WL | Delft Hydraulics / Deltares. A note on two-step numerical modeling.

Borsboom, M. (2019). *Numerical design of a fast, robust and accurate 1D shallow-water solver for pipe flows with largetime scales – proposal (WORK IN PROGRESS)*.

Borsboom, M. (2024). *Export from MapleSoft: "funcfitsmoothing_Fouriermodeanalysis.pdf"*.

Buijs, J. (2024). *Pseudo-Transient Continuation for the Rapid Convergence of Shallow Water Equation Solvers*. Delft University of Technology, Deltares internship report.

Hundsdorfer, W. and J. G. Verwer (2003). *Numerical solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer.

Kan, J. van, A. Segal, and F. Vermolen (2008). *Numerical Methods in Scientific Computing*. VSSD. ISBN: 9789071301506. URL: https://books.google.nl/books?id=fCARnQAACAAJ.

Logan, J. David. (1987). *Applied Mathemathics A contemporary approach*. Johan Wiley and Sons, Inc. ISBN: 0-471-85083-7.

Pulliam, T. H. (2014). *Time accuracy and the use of implicit methods*. Tech. rep. NASA Ames Research Center. URL: https://overflow.larc.nasa.gov/wp-content/uploads/sites/54/2014/06/Time_Accurate_Methods_V8.pdf.

# A Error estimation

In this section a Fourier mode analysis for constant $\Delta x$ and $\Psi$ of Equation (3.14) is given. For convenience the equation is repeated here:

$$\widetilde{u} - \frac{\partial}{\partial x}\Psi\frac{\partial \widetilde{u}}{\partial x} = u_{giv} \tag{A.1}$$

The Fourier-mode transform of Equation (3.14) reads, when $\Delta x = constant$ and $\Psi = c_\Psi \Delta x^2 = constant$:

$$\widetilde{u}\exp(\mathrm{i}kx)\left(1 + \Psi k^2\right) = \mathcal{F}(u_{giv}) \tag{A.2}$$

multiply $\Psi$ by $\Delta x^2/\Delta x^2 = 1$ yields

$$\widetilde{u}\exp(\mathrm{i}kx)\left(1 + \frac{\Psi}{\Delta x^2}(k\Delta x)^2\right) = \mathcal{F}(u_{giv}) \tag{A.3}$$

which is easier to use in the analysis due to the term $k\Delta x$.

The discretisation of Equation (3.20) with constant $\Delta x$ and $\Psi$ yields:

$$\left(\frac{1}{8} - c_\Psi\right)u_{i-1} + \left(\frac{6}{8} + 2c_\Psi\right)u_i + \left(\frac{1}{8} - c_\Psi\right)u_{i+1} = \tag{A.4}$$

$$= \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+\frac{1}{2}}} u_{giv}\,dx \tag{A.5}$$

Substitution of the Fourier modes

$$\overline{u}_k = \overline{u}_k \exp(\mathrm{i}k\Delta x) \quad \text{and} \quad u_{giv} = u_{giv,k}\exp(\mathrm{i}k\Delta x) \tag{A.6}$$

in this equation yields:

$$\overline{u}_k\left(\left(\frac{1}{8} - c_\Psi\right)\exp(-\mathrm{i}k\Delta x) + \left(\frac{6}{8} + 2c_\Psi\right) + \left(\frac{1}{8} - c_\Psi\right)\exp(-\mathrm{i}k\Delta x)\right) = \tag{A.7}$$

$$= \frac{u_{giv,k}}{\Delta x}\int_{x_{i-1/2}}^{x_{i+\frac{1}{2}}} \exp(\mathrm{i}k\Delta x)\,dx \tag{A.8}$$

$\Rightarrow$

$$\overline{u}_k\left(\frac{6}{8} + 2c_\Psi + \left(\frac{1}{8} - c_\Psi\right)2\cos(k\Delta x)\right) = \tag{A.9}$$

$$= u_{giv,k}\frac{-\mathrm{i}(\exp(\frac{\mathrm{i}}{2}k\Delta x) - \exp(\frac{\mathrm{i}}{2}k\Delta x))}{k\Delta x}$$

$\Leftrightarrow$

$$\overline{u}_k \left( \frac{6}{8} + 2c_\Psi + \left( \frac{1}{8} - c_\Psi \right) 2\cos(k\Delta x) \right) = u_{giv,k} \frac{2\sin(\frac{1}{2}k\Delta x)}{k\Delta x} \tag{A.10}$$

The ratio between the regularized function $\widetilde{u}$ (Equation (A.3)) and given function reads (see Figure A.1):

$$\widetilde{r} = \left| \frac{\widetilde{u}_k}{u_{giv,k}} \right| = \left| \frac{1}{1 + c_\Psi (k\Delta x)^2} \right| \tag{A.11}$$

and the ratio between the piecewise linear function $\overline{u}$ (Equation (A.10)) and given function reads (see Figure A.1):

$$\overline{r} = \left| \frac{\overline{u}_k}{u_{giv,k}} \right| = \left| \frac{2\sin(\frac{1}{2}k\Delta x)}{k\Delta x \left( \frac{6}{8} + 2c_\Psi + \left( \frac{1}{8} - c_\Psi \right) 2\cos(k\Delta x) \right)} \right| \tag{A.12}$$

$\Leftrightarrow$

$$\overline{r} = \left| \frac{\overline{u}_k}{u_{giv,k}} \right| = \left| \frac{8\sin(\frac{1}{2}k\Delta x)}{k\Delta x \left( 3 + \cos(k\Delta x) + 8c_\Psi (1 - \cos(k\Delta x)) \right)} \right| \tag{A.13}$$

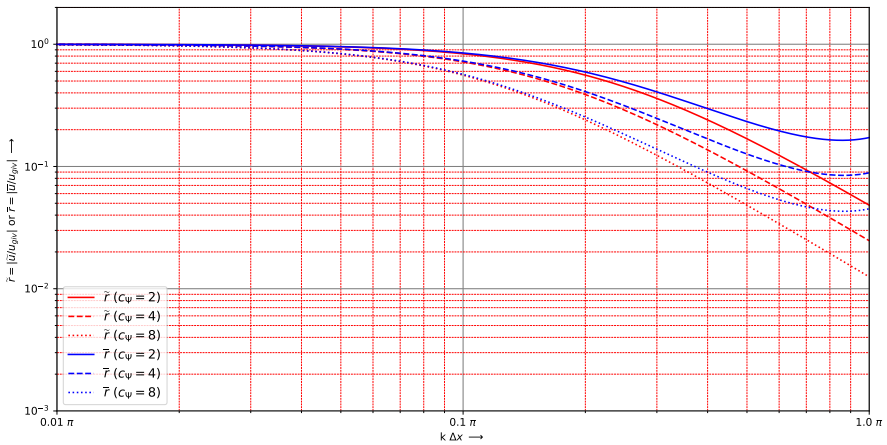



***Figure A.1:*** *Ration between* $\widetilde{r} = |\widetilde{u}_k/u_{giv,k}|$ *, and* $\overline{r} = |\overline{u}_k/u_{giv,k}|$*. For different values of* $c_\Psi = \Psi/\Delta x^2$*.*

## A.1 Numerical error

Discretization error values are determined when $\Delta x = constant$ and $\Psi = 0$, Equation (3.20) reduces to:

$$\frac{1}{8}u_{i-1} + \frac{6}{8}u_i + \frac{1}{8}u_{i+1} = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+\frac{1}{2}}} u_{giv}\, dx \tag{A.14}$$

Numerical error as discretized between grid pints

$$\textbf{errorFVEgridpoint}(k\Delta x) = \left| 1 - \frac{8\sin(\frac{1}{2}k\Delta x)}{k\Delta x(3+\cos(k\Delta x))} \right| \tag{A.15}$$

Numerical error as discretized between cell centres (numerical Fourier-mode through the cell centres)

$$\textbf{errorFVEcellcentre}(k\Delta x) = \left| 1 - \frac{8\sin(\frac{1}{2}k\Delta x)\cos(\frac{1}{2}k\Delta x)}{k\Delta x(3+\cos(k\Delta x))} \right| \tag{A.16}$$

With lowest order estimation of (see Borsboom (2024)):

$$-\frac{1}{24}(k\Delta x)^2 - \frac{7}{960}(k\Delta x)^4 + O(\Delta x^6) \tag{A.17}$$

So if the numerical error needs to be smaller then 1 %, the number of grid cells per wave length ($\lambda = N\Delta x$) is computed as

$$\frac{1}{24}(k\Delta x)^2 < 0.01 \Rightarrow k\Delta x < 0.5 \Rightarrow \tag{A.18}$$

$$\left(\frac{2\pi}{N\Delta x}\right)\Delta x < 0.5 \Rightarrow N > 4\pi \approx 13 \tag{A.19}$$

so per wave detail $\approx 7$ grid cells.

Discretization error values are given when $\Psi = c_\Psi \Delta x^2 = 0$ (see Figure A.2):

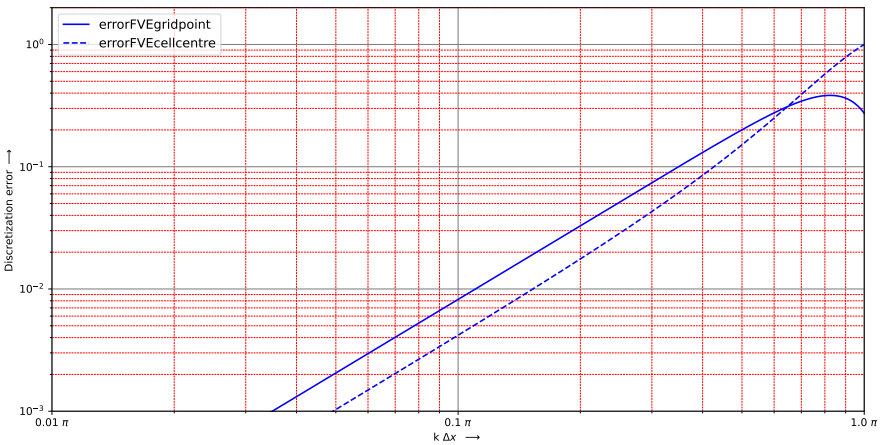



***Figure A.2:*** *Error function for value* $\Psi = c_\Psi \Delta x^2 = 0$.

## A.2 Determining the factor $c_\Psi$

In the limit of $k\Delta x \to \infty$ Equation (A.11) behaves as:

$$\lim_{k\Delta x\to\infty} \frac{1}{1+c_\Psi(k\Delta x)^2} = \frac{1}{c_\Psi(k\Delta x)^2} \tag{A.20}$$

The cut-off frequency is determined by $1 = 1/(c_\Psi(k\Delta x)^2)$, hence filter parameter $c_\Psi = 1/(k\Delta x)^2$ gives cut-off frequency $k$, or $k\Delta x = \sqrt{1/c_\Psi}$ is obtained with filter parameter $c_\Psi$. Given the wave length from the required numerical error (1 %, $N = 13$), the coefficient $c_\Psi$ reads:

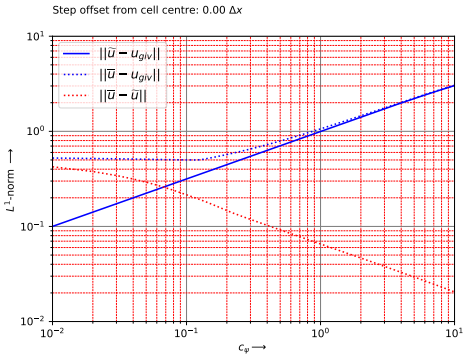
$$c_\Psi = \frac{1}{(k\Delta x)^2} \quad \Rightarrow \quad c_\Psi = \frac{N^2}{(2\pi)^2} \approx 4 \tag{A.21}$$

***Table A.1:*** *Several typical values for numerical accuracy, $c_\psi$ and number of nodes per wavelength ($N$), the highligthed line is set as default.*

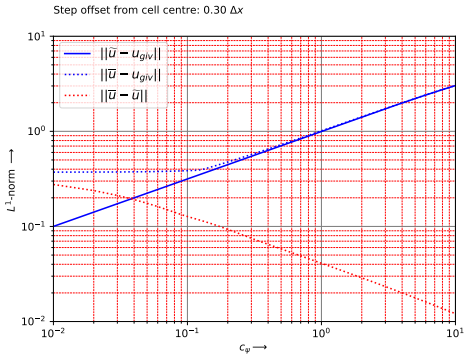
| | Accuracy | $c_\Psi$ | N |
|---|---|---|---|
| **1** | 5 % | 0.5 | 4.5 |
| **2** | 2 % | 2 | 8.9 |
| **3** | 1 % | 4 | 12.8 |
| **4** | 0.5 % | 10 | 18.1 |

## A.3 $L^1$-norm for the functions $\widetilde{u}$, $\overline{u}$ and $u_{given}$

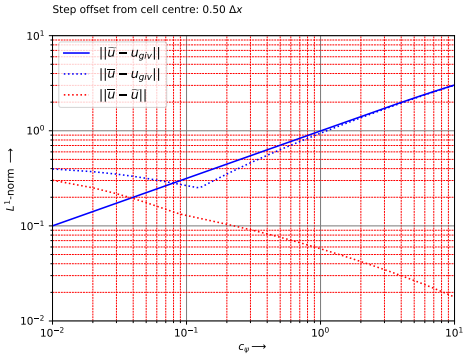
The next figures show the $L^1$-norm for the functions $||\widetilde{u} - u_{giv}||_1$, $||\overline{u} - u_{giv}||_1$ and $||\overline{u} - \widetilde{u}||_1$




***(a)*** *Step defined at cell centre.*




***(b)*** *Step defined with an of set of 0.3 $\Delta x$ from cell centre.*

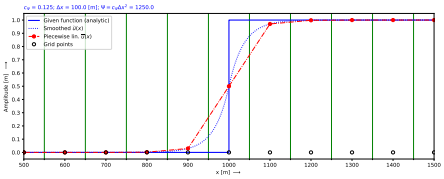



***(c)*** *Step defined with an of set of 0.5 $\Delta x$ from cell centre, thus defined at cell face.*

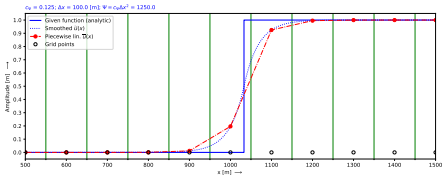
***Figure A.3:*** *Several plots of $L^1$-norm for different locations of the step.*

As seen from these figures. When choosing $c_\Psi > 4$ then the $L^1$-norm of $||\overline{u} - \widetilde{u}||_1$ is for all locations of the step smaller then $2 \times 10^{-2}$ (red dotted lines in the plots). Also the blue and blue dotted line are close together for $c_\Psi > 4$. That is the region in which we want to have the discetisation because there is the difference between the piecewise linear numerical solution $\overline{u}$ and the regularized solution $\widetilde{u}$ is independent of the location of the step, given by the function $u_{giv}$.

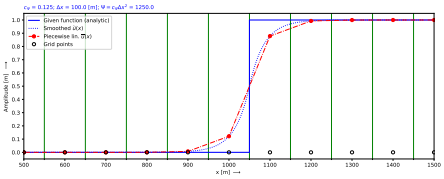
We also see that the dotted blue line is completely above the blue line if the step is located at the cell centre and with an offset of $0.3\Delta x$ (see Figure A.3a and Figure A.3b) and it is partly below the blue line if the offset of the step is located at $0.5\Delta x$ (Figure A.3c). The dotted blue line has also a sharp bend at the location where $c_\Psi = 0.125$. These approximations are presented in the Figure A.4a, Figure A.4b and Figure A.4c. As seen from these plot the piecewise linear approximation shown in Figure A.4c is closer to the Heaviside function as the other two approximations.



***(a)*** *Step defined at cell centre.*



***(b)*** *Step defined with an of set of 0.3 $\Delta x$ from cell centre.*



***(c)*** *Step defined with an of set of 0.5 $\Delta x$ from cell centre, thus defined at cell face.*

***Figure A.4:*** *Several plots of the piecewise linear approximation ($\overline{u}$) of the Heaviside function compared to the regularized function ($\widetilde{u}$).*

# B Diagonalise 1D wave equation with convection

The one dimensional shallow water equations with convection for flat bottom ($\frac{1}{2}g\,\partial h^2/\partial x = gh\,\partial h/\partial x$ and $\partial z_b/\partial x = 0$), reads

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{B.1}$$

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q^2}{h}\right) + gh\frac{\partial h}{\partial x} = 0 \tag{B.2}$$

The convection term can rewritten in the linear form for the derivatives as:

$$\frac{\partial}{\partial x}\left(\frac{q^2}{h}\right) = \frac{2q}{h}\frac{\partial q}{\partial x} - \frac{q^2}{h^2}\frac{\partial h}{\partial x} \tag{B.3}$$

In matrix notation it reads:

$$\begin{pmatrix} h \\ q \end{pmatrix}_t + \begin{pmatrix} 0 & 1 \\ gh - \frac{q^2}{h^2} & \frac{2q}{h} \end{pmatrix} \begin{pmatrix} h \\ q \end{pmatrix}_x = 0 \tag{B.4}$$

To make the system of equations diagonal, we have to find the eigen values and the eigen vectors. The eigenvalues of the matrix are (using **Maplesoft**)

$$\lambda_1 = \frac{q}{h} + \sqrt{gh} \qquad \text{and} \qquad \lambda_2 = \frac{q}{h} - \sqrt{gh} \tag{B.5}$$

The eigenvectors are

$$\begin{pmatrix} 1 \\ \frac{q}{h} + \sqrt{gh} \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} 1 \\ \frac{q}{h} - \sqrt{gh} \end{pmatrix} \tag{B.6}$$

The diagonalised SWE with convection read (after multiplying by $2\sqrt{gh}$, using **Maplesoft**)

$$\begin{pmatrix} \sqrt{gh} - \frac{q}{h} & 1 \\ \sqrt{gh} + \frac{q}{h} & -1 \end{pmatrix} \begin{pmatrix} h \\ q \end{pmatrix}_t + \begin{pmatrix} \frac{q}{h} + \sqrt{gh} & 0 \\ 0 & \frac{q}{h} - \sqrt{gh} \end{pmatrix} \begin{pmatrix} \sqrt{gh} - \frac{q}{h} & 1 \\ \sqrt{gh} + \frac{q}{h} & -1 \end{pmatrix} \begin{pmatrix} h \\ q \end{pmatrix}_x = 0 \tag{B.7}$$

Written in two separate equations

$$\left(\sqrt{gh} - \frac{q}{h}\right)\frac{\partial h}{\partial t} + \frac{\partial q}{\partial t} + \left(\frac{q}{h} + \sqrt{gh}\right)\left(\left(\sqrt{gh} - \frac{q}{h}\right)\frac{\partial h}{\partial x} + \frac{\partial q}{\partial x}\right) = 0 \quad \text{right going} \tag{B.8}$$

$$\left(\sqrt{gh} + \frac{q}{h}\right)\frac{\partial h}{\partial t} - \frac{\partial q}{\partial t} + \left(\frac{q}{h} - \sqrt{gh}\right)\left(\left(\sqrt{gh} + \frac{q}{h}\right)\frac{\partial h}{\partial x} - \frac{\partial q}{\partial x}\right) = 0 \quad \text{left going} \tag{B.9}$$

Which can be written as a combination of the continuity and momentum equation (Borsboom, 2001, eq. 4), (keep in mind that also Equation (B.3) is used)

$$\begin{matrix}\text{right going}\\ \text{left going}\end{matrix} \quad \begin{pmatrix} \sqrt{gh} - \frac{q}{h} & 1 \\ \sqrt{gh} + \frac{q}{h} & -1 \end{pmatrix} \begin{pmatrix} \textit{continuity eq.} \\ \textit{momentum eq.} \end{pmatrix} = 0 \tag{B.10}$$

or, after multiplying with $h$

$$\begin{matrix}\text{right going}\\ \text{left going}\end{matrix} \quad \begin{pmatrix} h\sqrt{gh} - q & h \\ h\sqrt{gh} + q & -h \end{pmatrix} \begin{pmatrix} \textit{continuity eq.} \\ \textit{momentum eq.} \end{pmatrix} = 0 \tag{B.11}$$

Email: jan.mooiman@outlook.com
GitHub Mooiman