

# Vertex-centered finite-volume discretization of the two-dimensional shallow-water equations

by

Deltares

Copyright © 2022 Deltares

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission of the author.



CONTENTS

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Scope</b>	<b>9</b>
<b>3</b>	<b>Governing equations</b>	<b>11</b>
3.1	Introduction . . . . .	11
3.2	The two-dimensional shallow-water equations . . . . .	12
3.2.1	Continuity . . . . .	12
3.2.2	Momentum equations . . . . .	13
3.3	Boundary conditions . . . . .	14
<b>4</b>	<b>Spatial discretization of the two-dimensional shallow-water equations</b>	<b>17</b>
4.1	Introduction . . . . .	17
4.2	A collocated vertex-centered finite-volume method . . . . .	17
4.2.1	Continuity . . . . .	19
4.2.2	Momentum equations . . . . .	27

4.2.3	Adding dissipation for stability . . . . .	32
4.2.4	Checking well-balancedness of the momentum equation . . . . .	32
4.2.5	Other open questions . . . . .	35
<b>5</b>	<b>Time integration of the two-dimensional shallow-water equations</b>	<b>37</b>
5.1	Introduction . . . . .	37
5.2	Solution strategies . . . . .	37
5.3	Time integration approaches . . . . .	38
5.3.1	Explicit methods . . . . .	38
5.3.2	(Semi-)Implicit methods . . . . .	38
<b>6</b>	<b>Boundary conditions</b>	<b>39</b>
6.1	Introduction . . . . .	39
6.2	Boundary positioning . . . . .	39
6.3	Boundary procedure for 1D linearized shallow water equations . . . . .	41
<b>7</b>	<b>Stability analysis</b>	<b>45</b>
<b>8</b>	<b>Implementation aspects</b>	<b>47</b>
8.1	Introduction . . . . .	47
8.2	Class hierarchy . . . . .	47
8.3	Parallelization . . . . .	48
<b>References</b>		<b>49</b>
<b>A</b>	<b>Grid insensitivity of the vertex-centered finite volume scheme</b>	<b>51</b>
A.1	Introduction . . . . .	51
A.2	Quadrilateral grids . . . . .	53
A.3	Triangular grids . . . . .	55

A.4	General formulation . . . . .	57
A.4.1	Sub-control volumes . . . . .	57
A.4.2	Sub-control volume faces . . . . .	58
<b>B</b>	<b>Application: Scalar 1D advection</b>	<b>59</b>
B.1	Introduction . . . . .	59
B.2	Equations . . . . .	59
B.3	Numerical method . . . . .	60
B.4	Numerical experiments . . . . .	63
B.4.1	Sinusoidal wave . . . . .	64
B.4.2	Gaussian disturbance . . . . .	64
B.5	Results . . . . .	65
B.5.1	Sinusoidal wave . . . . .	65
B.5.2	Gaussian disturbance . . . . .	66
B.6	Analysis . . . . .	68
B.7	Discussion . . . . .	71
B.8	Conclusions . . . . .	71
<b>C</b>	<b>Boundary conditions</b>	<b>73</b>
C.1	Introduction . . . . .	73
C.2	Equations . . . . .	73
C.3	Boundary handling . . . . .	74
C.3.1	Discretizing the model equation(s) at the boundary . . . . .	74
C.3.2	Applying the user-specified boundary conditions . . . . .	81
C.4	Numerical experiments . . . . .	82
C.5	Results . . . . .	82
C.6	Analysis . . . . .	82

C.7	Discussion . . . . .	83
C.8	Conclusions . . . . .	83
<b>D</b>	<b>Boundary conditions for 2D shallow water</b>	<b>85</b>
D.1	Introduction . . . . .	85
D.2	Equations . . . . .	85
D.3	Boundary handling . . . . .	87
D.3.1	Discretizing the model equation(s) at the boundary . . . . .	87
D.3.2	Applying the user-specified boundary conditions . . . . .	94
D.4	Solving loosely-coupled systems . . . . .	95
D.5	Numerical experiments . . . . .	97
D.6	Results . . . . .	97
D.7	Analysis . . . . .	97
D.8	Discussion . . . . .	97
D.9	Conclusions . . . . .	97

# CHAPTER 1

---

## INTRODUCTION

**IMPORTANT NOTE:** the present document is a draft and has a non-official status.

...

Some notes on the aim and context.

...



## CHAPTER 2

### SCOPE

- Shallow water
- Depth-averaged two-dimensional (2D) flow (and/or 1D?)
- Including the following physical processes/aspects:
  - non-stationary flow,
  - variable topography,
  - momentum advection,
  - bed friction,
  - wind friction,
  - mass sources/sinks (e.g. precipitation),
  - wetting and drying
  - viscous flux (possibly needed to include artificial viscosity)
- Including the following boundary condition types:
  - water level (time series),
  - (total) discharge (time series)

- Possible applications:
  - Rivers
  - 2D (coastal) flooding scenarios
  - Rainfall events

# CHAPTER 3

---

## GOVERNING EQUATIONS

### 3.1 Introduction

We consider the two-dimensional, depth-averaged shallow-water equations in conservative form. The state vector of primary variables is  $\vec{U} = (h, q, r)^T$ , where  $h$  is the total water depth and  $\vec{q} = (q, r)^T = (hu, hv)^T$  is the unit discharge vector, with  $q$  and  $r$  its components in  $x$ - and  $y$ -direction. The two-dimensional (2D) shallow-water equations consist of two conservation laws:

1. Conservation of mass
2. Conservation of momentum

The equation expressing the principle of conservation of mass is often referred to as the *continuity equation*. The equation expressing the principle of conservation of momentum is often referred to as the *momentum equation* and can be derived from the Navier-Stokes equations, by applying certain simplifications/assumptions.

## 3.2 The two-dimensional shallow-water equations

### 3.2.1 Continuity

$$\underbrace{\frac{\partial \vec{U}}{\partial t}}_{\text{Time derivative}} + \underbrace{\nabla \cdot \vec{F}}_{\text{Flux}} = \underbrace{\vec{S}}_{\text{Source}} \quad (3.1)$$

$$\underbrace{\vec{U}}_{\text{Solution}} = \underbrace{\nabla \vec{F}}_{\text{Gradient}} + \underbrace{\vec{S}}_{\text{Source}} \quad (3.2)$$

We consider the two-dimensional continuity equation:

$$\underbrace{\frac{\partial h}{\partial t}}_{\text{Time derivative}} + \underbrace{\nabla \cdot \vec{q}}_{\text{Mass flux}} = \underbrace{\tilde{S}}_{\text{Mass source}} \quad (3.3)$$

or written out in its vector components:

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} + \frac{\partial r}{\partial y} = \tilde{S} \quad (3.4)$$

where, in addition to the definitions before,  $\tilde{S}$  is a normalized source term (with unit  $\text{m}^3/\text{s} / \text{m}^2 = \text{m}/\text{s}$ ). As commonly accepted,  $t$  represents time and  $x, y$  represent the spatial coordinates in the horizontal plane.

The time derivative (or time rate of change) term accounts for the change in time of the water depth  $h$  in some flow region. The flux term accounts for the flow into or out of that region. The source term can contain sources/sinks of mass (of water), such as precipitation, irrigation, infiltration (into the soil), etc. in the region under consideration.

### 3.2.2 Momentum equations

Now, we consider the two-dimensional momentum equation, written in conservative form (for now without Coriolis term):

$$\underbrace{\frac{\partial \vec{q}}{\partial t}}_{\text{Time derivative}} + \underbrace{\nabla \cdot (\vec{q} \vec{q}/h)}_{\text{Momentum advection}} = -\underbrace{\nabla \left( \frac{1}{2} gh^2 \right)}_{\text{Pressure}} - \underbrace{gh \nabla z_b}_{\text{Bed-slope source}} \\ + \underbrace{\nabla \cdot \left( vh \left( \nabla (\vec{q}/h) + \nabla (\vec{q}/h)^T \right) \right)}_{\text{Viscous flux}} - \underbrace{\frac{c_{D_b} \vec{q} |\vec{q}|}{h^2}}_{\text{Bed shear stress}} - \underbrace{c_{D_w} \vec{U}_{10}^2}_{\text{Wind shear stress}} \quad (3.5)$$

where  $g$  is the gravitational acceleration,  $c_{D_b}$  and  $c_{D_w}$  are the bed-friction and wind-friction coefficients, respectively,  $v$  is the viscosity,  $|\vec{q}| = \sqrt{q^2 + r^2}$ , and  $\vec{U}_{10} = (U_{10}, V_{10})$  is the wind-velocity vector, commonly taken from wind measurements at approximately 10 m above the free-surface level.

As can be seen the momentum equation contains a time derivative (of the momentum), an advective term, a pressure-gradient term, a bed-slope (source or reaction) term, a viscous flux, a bed-shear stress and finally a wind-shear stress term.

Written out in two equations for the two vector components  $q$  and  $r$ , we obtain:

$$\frac{\partial q}{\partial t} + \nabla \cdot (\vec{q} q/h) = -\nabla \left( \frac{1}{2} gh^2 \right) + gh \nabla z_b \\ + \nabla \cdot \left( vh \left( 2 \frac{\partial (q/h)}{\partial x} + \frac{\partial (q/h)}{\partial y} + \frac{\partial (r/h)}{\partial x} \right) \right) - \frac{c_{D_b} q |\vec{q}|}{h^2} - c_{D_w} U_{10}^2 \quad (3.6)$$

and

$$\frac{\partial r}{\partial t} + \nabla \cdot (\vec{q} r/h) = -\nabla \left( \frac{1}{2} gh^2 \right) + gh \nabla z_b \\ + \nabla \cdot \left( vh \left( \frac{\partial (r/h)}{\partial x} + \frac{\partial (q/h)}{\partial y} + 2 \frac{\partial (r/h)}{\partial y} \right) \right) - \frac{c_{D_b} r |\vec{q}|}{h^2} - c_{D_w} V_{10}^2 \quad (3.7)$$

The momentum equation is, in fact, Newton's second law of motion  $F = ma$ , where the forces  $F$  and the acceleration  $a$  have been written out in more detail.

The forces in the equation are:

- the *gravitational body force*, pulling water 'down', resulting in a pressure term depending on the gradient in hydrostatic pressure ( $\frac{1}{2} gh^2$ ),

- the *bed-slope (reaction) term*, which in fact is the counter-reacting hydrostatic pressure on the bottom,
- the *viscous flux*, which is a shear stress acting between fast- and slow-moving fluid particles,
- the *bed shear stress*, which is the shear stress due to the frictional effect of a rough bed (and the accompanying boundary layer),
- the *wind shear stress*, which is the shear stress due to the frictional effect of wind above the free surface.

The acceleration terms are:

- the *local acceleration*, or the time derivative,
- the *advection acceleration*, or the advection term.

### 3.3 Boundary conditions

We only solve the governing equations in the interior of the domain. To adequately represent the connection of our modelling domain to the outside world, we need to introduce boundary conditions. Mathematically speaking, the boundary conditions are also needed to close the system of equations that needs to be solved. The number of unknown variables (for 2D shallow water, these are the variables in our state vector  $\vec{U}$ , i.e.  $h$ ,  $q$  and  $r$  and the flow state at the boundary (sub-critical or super-critical flow)) determine the number of boundary conditions we need to prescribe at a boundary, to obtain a so-called well-posed system of equations. For typical sub-critical flow conditions, only one variable needs to be given a boundary condition at a boundary, e.g. at an upstream boundary. The other variables then obtain their boundary conditions at other (e.g. downstream) boundaries. For super-critical flow, no signals can travel upstream, and boundary conditions for at least two (formally for all) variables need to be prescribed at the upstream (super-critical flow) boundary. In this case, no boundary condition is needed (nor physically relevant) at the downstream boundary. Handling such situations in hydrodynamic models – in particular the transitions from and to super-critical flow – is a challenge. This will be further elaborated on in Chapter 6.

The type of boundary conditions typically depends on the application. For oceanic / estuarine applications, tidal signals in the form of time-varying water levels are often prescribed. For river applications, an upstream river discharge is a common boundary

condition. Possible lateral inflows of smaller contributory rivers or streams are often also included as discharge boundary conditions. The simplest way of prescribing such boundary conditions is using Dirichlet-type conditions, where the quantity (i.e. water level, water depth or discharge) is given a value at the boundary. Such a value can, however, be prescribed varying in space along the boundary, or varying in time.

Other types of somewhat less common boundary conditions are Neumann boundary conditions, where the gradient of a certain quantity is prescribed. As a possible application, one could consider the prescription of the water level slope at an open boundary, particularly for river applications.

Combinations of Dirichlet and Neumann boundary conditions are also possible. These are Robins-type boundary conditions. Here a (weighted) combination of a value and a gradient is prescribed.

The final – and least common – type of boundary condition is a Cauchy-type boundary condition, where *both* a Dirichlet *and* a Neumann condition are prescribed for a certain quantity.



## CHAPTER 4

---

### SPATIAL DISCRETIZATION OF THE TWO-DIMENSIONAL SHALLOW-WATER EQUATIONS

#### 4.1 Introduction

This chapter describes the spatial discretization of the governing equations described in chapter 3. Many different discretization methods can be applied. In the next section the presently-(to-be)-implemented method is described.

#### 4.2 A collocated vertex-centered finite-volume method

In this section we describe a vertex-centered finite-volume discretization of the shallow-water equations, as described in chapter 3.

...CONTEXT:

[Busto et al. \(2021\)](#)

[McBride et al. \(2007\)](#)

more references to be added

...

The primary variables  $h$ ,  $q$  and  $r$  (and the bed level  $z_b$ ) will be stored on the grid vertices, i.e. a collocated storage is applied. Control volumes around grid vertices will be constructed by assembling sub-control volumes around these vertices, one sub-control volume for each cell connected to the vertex. The sub-control volumes are constructed by dividing each computational cell in  $N_s$  sub-volumes, by introducing sub-control volume boundaries between the grid cell centroid and the grid cell face midpoints (see Figure 4.1). These sub-control volumes will always be quadrilateral (not the control volume itself!), irrespective of the grid structure. For quadrilateral grids (including rectangular grids)  $N_s = 4$ , whereas for triangular grids  $N_s = 3$ . Similarly, for possible pentagonal grids  $N_s = 5$  and for hexagonal grids  $N_s = 6$ , as can be seen in the figure.

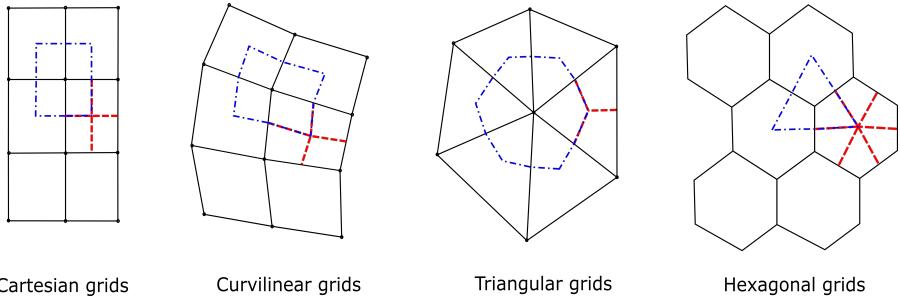


Figure 4.1: Illustration of the control-volume (dash-dotted blue lines) and sub-control-volume construction (dashed red lines) on (two-dimensional) rectangular, curvilinear, triangular and hexagonal grids (black lines).

For brevity, we introduce the following abbreviations for the control volumes:

- cv = control volume
- scv = sub-control volume
- scvf = sub-control volume face

In the following sections we focus on the spatial discretization of the equations. The time integration method will be addressed in chapter 5. We do not consider the system of equations as a whole (in vector form), because we want to retain the possibility

to apply different time integration methods or solution strategies for the different equations in the system. As an example, one may want to employ a semi-implicit discretization and substitute the momentum equations in the continuity equation, as is done in D-Flow FM (and e.g. UnTRIM). Such an approach would not be possible when considering the system of equations in vector form. Therefore, we first consider the discretization of the continuity equation in section 4.2.1 and later we consider the (rotationally-invariant) momentum equation in section 4.2.2.

### 4.2.1 Continuity

We consider the two-dimensional continuity equation:

$$\underbrace{\frac{\partial h}{\partial t}}_{\text{Time derivative}} + \underbrace{\nabla \cdot \vec{q}}_{\text{Mass flux}} = \underbrace{\tilde{S}}_{\text{Mass source}} \quad (4.1)$$

or written out in its vector components:

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} + \frac{\partial r}{\partial y} = \tilde{S} \quad (4.2)$$

where, in addition to the definitions before,  $\tilde{S}$  is a normalized source term (with unit  $\text{m}^3/\text{s} / \text{m}^2 = \text{m}/\text{s}$ ). As commonly accepted,  $t$  represents time and  $x, y$  represent the spatial coordinates in the horizontal plane.

This differential equation describes the variation in space and time of the water depth  $h$ . To resolve the derivatives in the equation, we need to integrate in space and time. The integration in space is an integration over the model domain, which in turn becomes an integration over the computational grid.

Integrating equation 4.2 over a control volume  $\Omega_{i,j}$ , we obtain:

$$\int_{\Omega_{i,j}} \frac{\partial h}{\partial t} dS + \int_{\Omega_{i,j}} \frac{\partial q}{\partial x} dS + \int_{\Omega_{i,j}} \frac{\partial r}{\partial y} dS = \int_{\Omega_{i,j}} \tilde{S} dS \quad (4.3)$$

The two cell integrals of the fluxes in the continuity equation can be rewritten to boundary integrals using Gauss' divergence theorem:

$$\int_{\Omega_{i,j}} \frac{\partial h}{\partial t} dS + \int_{\partial\Omega_{i,j}} (\vec{q} \cdot \vec{n}) dl = \int_{\Omega_{i,j}} \tilde{S} dS \quad (4.4)$$

Now we introduce the flux  $F_j$  over the edge of a sub-control volume:

$$F_j = \int_{\Gamma_j} (\vec{q}_j \cdot \vec{n}_j) dl_j \quad (4.5)$$

where  $\Gamma_j$  is the edge with index  $j$  of a sub-volume of  $\Omega_{i,j}$ . Note, therefore, that this is now a nested integral: an integral along a sub-edge  $\Gamma_j$ , as part of an integral along the control-volume boundary  $\partial\Omega$ . Substituting this in 4.4, we obtain:

$$\underbrace{\int_{\Omega_{i,j}} \frac{\partial \vec{U}}{\partial t} dS}_{\text{Time derivative}} + \underbrace{\sum_{j \in \partial\Omega_{i,j}} \vec{F}_j}_{\text{Flux term}} = \underbrace{\int_{\Omega_{i,j}} \vec{S} dS}_{\text{Source term}} \quad (4.6)$$

$$\underbrace{\int_{\Omega_{i,j}} \frac{\partial h}{\partial t} dS}_{\text{Time derivative}} + \underbrace{\sum_{j \in \partial\Omega_{i,j}} F_j}_{\text{Flux term}} = \underbrace{\int_{\Omega_{i,j}} \tilde{S} dS}_{\text{Source term}} \quad (4.7)$$

where  $\vec{n}_j$  is the unit outward normal at boundary sub-edge  $j$ .

In Equation 4.7 there are three components that require computation: the time derivative of  $h_{i,j}$  (for control volume  $\Omega_{i,j}$ ), the flux  $F_j$  and the source term  $S_{i,j}$ . All three of these computations involve integration over some geometric entity. For the time derivative and the source term, this involves integration over a control volume, whereas for the flux it involves integration over the control volume boundary. The computation of these integrals involve integrals over the sub-volumes (or sub-edges of sub-volumes) that are connected to the vertex  $(i, j)$ . We discuss each of these individual computations/integrations in the following sections.

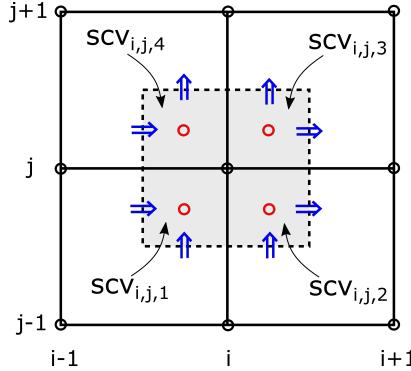


Figure 4.2: Illustration of the placement of variables in vertices and of the control volume in a rectangular 2d grid. The red circles show the quadrature points of the one-point Gauss integration rule for the four sub-volumes of control volume  $\Omega_{i,j}$ . The blue arrows correspond to the fluxes across the (dashed) boundary  $\partial\Omega$  of the control volume.

#### 4.2.1.1 Integration of the time derivative

The integral of the depth over a volume (actually a control volume area) will result in a representation of the actual water volume  $V$  in the control volume. As noted before, the integral will be composed of four integrals over the four "sub-volumes" of  $\Omega_{i,j}$ . In each of these four sub-volumes the actual water volume contribution can be computed by performing some integration rule over the volume. This can vary from a simple one-point-Gauss integration, to a more accurate integration involving more quadrature points. This flexibility needs to be available in the software code. In turn, the value(s) in the quadrature point(s) need(s) to be approximated/interpolated between the cell vertices where the values of the different quantities are known (i.e.  $h$  (water depth) and  $q, r$  as fluxes). This interpolation can be done with different orders of accuracy. Classical finite volume methods assume constant values within cells. Assuming (bi-)linear variation in cells will increase the accuracy of the approximation of the values in the quadrature points and with that the accuracy of the integration over the (sub-)volumes. Further improvement of the accuracy could be achieved by assuming (piece-wise) (bi-)quadratic or (bi-)cubic variation of variables within cells. Also this modelling flexibility needs to be available in the software code. In summary, the (sub-)cell integration function may require different implementations for different orders of *interpolation* and different order of *integration* (and number of quadrature

points involved in the integration).

As an example we shall work out the integration for the choice of (bi-)linear variation within cells with one-point Gauss quadrature per sub-volume. The one-point Gauss integration (within sub-volumes) then simply looks as follows:

$$\int_{\Omega_{i,j}} \frac{\partial h}{\partial t} dS = \sum_{sc \in \Omega_{i,j}} \frac{\partial h_{sc}}{\partial t} A_{sc} \quad (4.8)$$

where we for now simply assume that the temporal discretization of  $\partial h_{sc}/\partial t$  will be some function of  $h_{sc}$  that does not involve any further spatial coupling. The computation of the integral in (4.8) then comes down to finding an approximation to  $h_{sc}$ . For simplicity we now assume that the depth  $h_{sc}$  in the four sub-control volumes (Figure 4.3a) is bi-linearly interpolated as:

$$sc_{i,j,1} : \quad h_{sc} = \frac{9h_{i,j} + 3h_{i-1,j} + 3h_{i,j-1} + h_{i-1,j-1}}{16} \quad (4.9)$$

$$sc_{i,j,2} : \quad h_{sc} = \frac{9h_{i,j} + 3h_{i+1,j} + 3h_{i,j-1} + h_{i+1,j-1}}{16} \quad (4.10)$$

$$sc_{i,j,3} : \quad h_{sc} = \frac{9h_{i,j} + 3h_{i-1,j} + 3h_{i,j+1} + h_{i-1,j+1}}{16} \quad (4.11)$$

$$sc_{i,j,4} : \quad h_{sc} = \frac{9h_{i,j} + 3h_{i+1,j} + 3h_{i,j+1} + h_{i+1,j+1}}{16} \quad (4.12)$$

**NOTE:** Perhaps, the formulation in (4.8) should be generalized to involve some kind of a weighted sum over all quadrature points involved in the integration of the sub-volume.

The interpolations of the values in the sub-control volume centres can best be written in terms of a matrix of interpolation coefficients that is then multiplied by a vector of vertex values (in this case for the water depth  $h$ , but a formulation in terms of coefficients is applicable to any quantity that needs to be interpolated). Written out for cell  $(i, j)$ , this becomes:

$$h_{i,j,sc} = \begin{bmatrix} 9 & 3 & 3 & 1 \\ 3 & 9 & 3 & 1 \\ 1 & 3 & 9 & 3 \\ 3 & 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} h_{i-1,j-1} \\ h_{i,j-1} \\ h_{i,j} \\ h_{i-1,j} \end{bmatrix} \quad (4.13)$$

where  $sc \in 1, 2, 3, 4$ . Note the chosen counterclockwise ordering of the vertices of the cell (in the right-hand-side vector and corresponding to Figure 4.3a).

The time derivative term for  $h$  will typically contain two or more instances of  $h$ , depending on the type and the order of the chose time integration approach. For instance, for a simple (forward or backward) Euler time integration, the time-derivative would be approximated as:

$$\frac{\partial h}{\partial t} = \frac{h^{n+1} - h^n}{\Delta t} \quad (4.14)$$

or again integrated over the control volume  $\Omega_{i,j}$ :

$$\int_{\Omega_{i,j}} \frac{\partial h}{\partial t} dS = \int_{\Omega_{i,j}} \frac{h^{n+1} - h^n}{\Delta t} dS \quad (4.15)$$

$$= \frac{\int_{\Omega_{i,j}} h^{n+1} dS}{\Delta t} - \frac{\int_{\Omega_{i,j}} h^n dS}{\Delta t} \quad (4.16)$$

In other words, it requires the integration of  $h$  both at the current and at the new time level. However, as we have seen above, the integration involves bilinear interpolation in the surrounding vertex values. This thus requires applying the matrix-vector product from 4.13 for both time levels. For the current time level it simply results in an explicit right-hand-side contribution containing the matrix-vector product. For the new time level, this product can not yet be computed (as the new  $h^{n+1}$  is not yet known) and results in a contribution to the (mass) matrix:

$$h_{i,j,sc}^{n+1} = \begin{bmatrix} 9 & 3 & 3 & 1 \\ 3 & 9 & 1 & 3 \\ 1 & 3 & 9 & 3 \\ 1 & 3 & 3 & 9 \end{bmatrix} \begin{bmatrix} h_{i-1,j-1}^{n+1} \\ h_{i,j-1}^{n+1} \\ h_{i-1,j}^{n+1} \\ h_{i,j}^{n+1} \end{bmatrix} \quad (4.17)$$

which means the contribution to the mass matrix is just the coefficient matrix:

$$\begin{bmatrix} 9 & 3 & 3 & 1 \\ 3 & 9 & 1 & 3 \\ 1 & 3 & 9 & 3 \\ 1 & 3 & 3 & 9 \end{bmatrix} \quad (4.18)$$

This is the local mass matrix contribution for any variable that needs to be integrated over a (sub-)control volume.

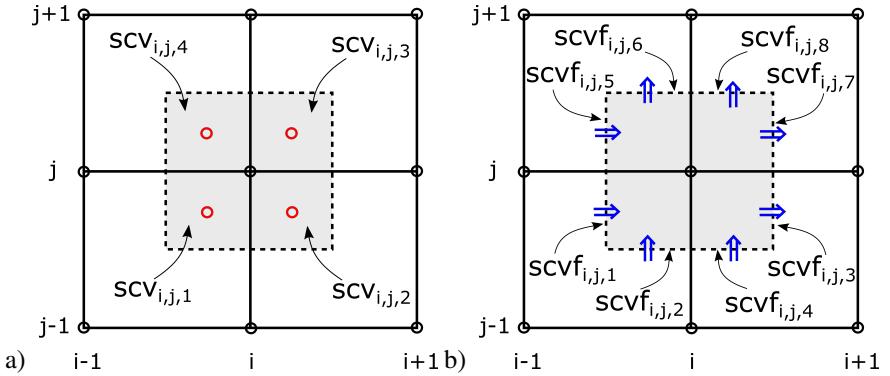


Figure 4.3: a) Illustration of the subvolumes of the control volume in a rectangular 2d grid. b) Illustration of the sub-volume edges of the control volume in a rectangular 2d grid.

The red circles show the quadrature points of the one-point Gauss integration rule for the four sub-volumes of control volume  $\Omega_{i,j}$ . The blue arrows correspond to the fluxes across the (dashed) boundary  $\partial\Omega$  of the control volume.

#### 4.2.1.2 Integration of the flux term

Similarly, the integration of the fluxes is performed. Starting again from equation 4.7, we now take a closer look on how to compute the fluxes over all sub-edges of the control volume  $\Omega_{i,j}$  around vertex  $(i, j)$ . On quadrilateral grids (and thus also on rectangular grids), a control volume boundary will consist of eight sub-edges.

The recipe for the computation of the fluxes over these sub-edges is similar to that of the volume of the sub-volumes. We apply a one-point Gauss integration over each sub-edge, for which we need the value of the flux  $F_j = (\vec{q}_j \cdot \vec{n}_j)l_j$ . To compute this value we need to interpolate the flux  $\vec{q}_j = (q, r)_j$  to the sub-control volume edges. This interpolation can again be done in several ways. For simplicity, we again limit ourselves to a bi-linear interpolation. The sub-edges are numbered as illustrated in figure 4.3b.

**NOTE:** This could be implemented as  $\vec{q}_j \cdot \vec{n}_j$  for each sub-volume edge, which is most general and can be applied later on to more general quadrilateral grids directly. Or it could be implemented in its components  $(q, r)$ , which is possibly more efficient, but not generally applicable for other grid types. What to implement in the first prototype?

**NOTE:** How to number the sub-volume edges? Will we loop over nodes or cells?

Should we "store" the sub-volume topology, i.e. which sub-volumes/sub-edges belong to which vertex?

The resulting interpolation formulas for the eight sub-edges are:

$$se_{i,j,1} : \quad q_{sc} = \frac{3q_{i,j} + 3q_{i-1,j} + q_{i,j-1} + q_{i-1,j-1}}{8} \quad (4.19)$$

$$se_{i,j,2} : \quad r_{sc} = \frac{3r_{i,j} + 3r_{i,j-1} + r_{i-1,j} + r_{i-1,j-1}}{8} \quad (4.20)$$

$$se_{i,j,3} : \quad q_{sc} = \frac{3q_{i,j} + 3q_{i+1,j} + q_{i,j-1} + q_{i+1,j-1}}{8} \quad (4.21)$$

$$se_{i,j,4} : \quad r_{sc} = \frac{3r_{i,j} + 3r_{i,j-1} + r_{i+1,j} + r_{i+1,j-1}}{8} \quad (4.22)$$

$$se_{i,j,5} : \quad q_{sc} = \frac{3q_{i,j} + 3q_{i-1,j} + q_{i,j+1} + q_{i-1,j+1}}{8} \quad (4.23)$$

$$se_{i,j,6} : \quad r_{sc} = \frac{3r_{i,j} + 3r_{i,j+1} + r_{i-1,j} + r_{i-1,j+1}}{8} \quad (4.24)$$

$$se_{i,j,7} : \quad q_{sc} = \frac{3q_{i,j} + 3q_{i+1,j} + q_{i,j+1} + q_{i+1,j+1}}{8} \quad (4.25)$$

$$se_{i,j,8} : \quad r_{sc} = \frac{3r_{i,j} + 3r_{i,j+1} + r_{i+1,j} + r_{i+1,j+1}}{8} \quad (4.26)$$

Again, the interpolations of the values over the sub-control volume faces can be written in terms of a matrix-vector product (in this case for the flux  $q$ , but a formulation in terms of coefficients is applicable to any quantity that needs to be interpolated). Written out for face  $(i, j)$ , this becomes:

$$h_{i,j,se} = \begin{bmatrix} 3 & 3 & 1 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} h_{i-1,j-1} \\ h_{i,j-1} \\ h_{i,j} \\ h_{i-1,j} \end{bmatrix} \quad (4.27)$$

where  $se \in 1, 2, 3, 4$ . Note the chosen counterclockwise ordering of the vertices of the cell (in the right-hand-side vector and corresponding to Figure 4.3a).

How this contribution is included in the discrete system of equations, depends on the chosen time integration for the flux term. For instance, for a simple (explicit) forward Euler time integration, the flux would be approximated as:

$$F_j^n = (\vec{q}_j^n \cdot \vec{n}_j) l_j \quad (4.28)$$

or integrated over the control volume boundary  $\partial\Omega_{i,j}$ :

$$\int_{\partial\Omega_{i,j}} (\vec{q}_j^n \cdot \vec{n}_j) l_j dl \quad (4.29)$$

$$= \frac{\int_{\Omega_{i,j}} h^{n+1} dS}{\Delta t} - \frac{\int_{\Omega_{i,j}} h^n dS}{\Delta t} \quad (4.30)$$

In other words it requires the integration of  $h$  both at the current and at the new time level. However, as we have seen above, the integration involves bilinear interpolation in the surrounding vertex values. This thus requires applying the matrix-vector product from 4.13 for both time levels. For the current time level it simply results in an explicit right-hand-side contribution containing the matrix-vector product. For the new time level, this product can not yet be computed (as the new  $h^{n+1}$  is not yet known) and results in a contribution to the (mass) matrix:

$$h_{i,j,sc}^{n+1} = \begin{bmatrix} 3 & 3 & 1 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} h_{i-1,j-1}^{n+1} \\ h_{i,j-1}^{n+1} \\ h_{i-1,j}^{n+1} \\ h_{i,j}^{n+1} \end{bmatrix} \quad (4.31)$$

which means the contribution to the mass matrix is just the coefficient matrix:

$$\begin{bmatrix} 3 & 3 & 1 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 1 & 3 \end{bmatrix} \quad (4.32)$$

This is the local mass matrix contribution for any variable that needs to be integrated over a (sub-)control volume face.

#### 4.2.1.3 Integration of the source term

The integration of the source term follows the same approach as the integration of the time derivative in the continuity equation. It should be noted that this may result in additional contributions to surrounding vertices, if the source term depends on one of the primary variables (most likely on  $h$  (possibly through dependency on the water level  $\zeta$ )). If the source term is simply an explicit contribution of a volume of water, the integration only results in contributions to the corresponding 9 entries in the right-hand-side vector.

#### 4.2.1.4 Summary

In summary, it can be noticed that the continuity equation involves a nine-point stencil (around vertex  $(i, j)$ ), both from the volume approximation and from the flux (volume boundary) approximation.

**Question:** Does the size of this stencil depend on the order of the interpolation and integration formulas? If so, can we / should we generalize the formulations such that we can have this flexibility in our code? Can we make sure the stencil stays local by introducing extra interior quadrature points as in a Finite Element Model?

For implicit time-integration methods, this would result in a nine-diagonal matrix (on regular structured – rectangular/curvilinear – grids) or in a block-banded matrix, with nine nonzero entries per row, on irregularly-numbered unstructured (quadrilateral) grids. For unstructured triangular grids, the stencil will involve all vertices that are part of the "N-tagon" around the vertex in question. If this is a pentagon, it will involve a 6-point stencil. If it is a hexagon, it will be a 7-point stencil, etc. This will result in a block-banded matrix with a variable number of off-diagonals for each row in the matrix.

For an explicit time-integration scheme, this spatial discretization still results in a matrix system that needs to be solved. The reason for this is the spatial discretization of the time derivative term for the volume in (4.7). After substitution of the sub-volume contributions from (4.9), this time derivative involves (in this case) nine  $V^{n+1}$  contributions, for all  $V^{n+1}$  values in the stencil, resulting in a so-called "mass matrix" as is common in Finite Element methods.

## 4.2.2 Momentum equations

Now, we consider the two-dimensional momentum equation, written in conservative form (for now without Coriolis term):

$$\underbrace{\frac{\partial \vec{q}}{\partial t}}_{\text{Time derivative}} + \underbrace{\nabla \cdot (\vec{q} \vec{q}/h)}_{\text{Momentum advection}} = -\nabla \left( \underbrace{\frac{1}{2} gh^2}_{\text{Pressure}} \right) - \underbrace{gh \nabla z_b}_{\text{Bed-slope source}} + \underbrace{\nabla \cdot \left( vh \left( \nabla(\vec{q}/h) + \nabla(\vec{q}/h)^T \right) \right)}_{\text{Viscous flux}} - \underbrace{\frac{c_{D_b} \vec{q} |\vec{q}|}{h^2}}_{\text{Bed shear stress}} - \underbrace{c_{D_w} \vec{U}_{10}^2}_{\text{Wind shear stress}} \quad (4.33)$$

where  $g$  is the gravitational acceleration,  $c_{D_b}$  and  $c_{D_w}$  are the bed-friction and wind-friction coefficients, respectively,  $v$  is the viscosity,  $|\vec{q}| = \sqrt{q^2 + r^2}$ , and  $\vec{U}_{10} =$

$(U_{10}, V_{10})$  is the wind-velocity vector, commonly taken from wind measurements at approximately 10 m above the free-surface level.

As can be seen the momentum equation contains a time derivative (of the momentum), an advective term, a pressure-gradient term, a bed-slope (source or reaction) term, a viscous flux, a bed-shear stress and finally a wind-shear stress term.

Written out in two equations for the two vector components  $q$  and  $r$ , we obtain:

$$\begin{aligned} \frac{\partial q}{\partial t} + \nabla \cdot (\vec{q}q/h) &= -\nabla \left( \frac{1}{2}gh^2 \right) - gh\nabla z_b \\ + \nabla \cdot \left( vh \left( 2\frac{\partial(q/h)}{\partial x} + \frac{\partial(q/h)}{\partial y} + \frac{\partial(r/h)}{\partial x} \right) \right) - \frac{c_{D_b}q|\vec{q}|}{h^2} - c_{D_w}U_{10}^2 \end{aligned} \quad (4.34)$$

and

$$\begin{aligned} \frac{\partial r}{\partial t} + \nabla \cdot (\vec{q}r/h) &= -\nabla \left( \frac{1}{2}gh^2 \right) - gh\nabla z_b \\ + \nabla \cdot \left( vh \left( \frac{\partial(r/h)}{\partial x} + \frac{\partial(q/h)}{\partial y} + 2\frac{\partial(r/h)}{\partial y} \right) \right) - \frac{c_{D_b}r|\vec{q}|}{h^2} - c_{D_w}V_{10}^2 \end{aligned} \quad (4.35)$$

Just as for the continuity equation, we need to integrate the momentum equations over a control volume  $\Omega$  (as constructed in Section 4.1). For brevity, we only show this for the  $q$ -equation:

$$\begin{aligned} \int_{\Omega_{i,j}} \frac{\partial q}{\partial t} dS + \int_{\Omega_{i,j}} \nabla \cdot (\vec{q}q/h) dS &= - \int_{\Omega_{i,j}} \nabla \left( \frac{1}{2}gh^2 \right) dS - \int_{\Omega_{i,j}} gh\nabla z_b dS \\ + \int_{\Omega_{i,j}} \nabla \cdot \left( vh \left( 2\frac{\partial(q/h)}{\partial x} + \frac{\partial(q/h)}{\partial y} + \frac{\partial(r/h)}{\partial x} \right) \right) dS \\ - \int_{\Omega_{i,j}} \frac{c_{D_b}q|\vec{q}|}{h^2} dS - \int_{\Omega_{i,j}} c_{D_w}U_{10}^2 dS \end{aligned} \quad (4.36)$$

Again we rewrite the volume integrals of the flux terms into an integral along the control-volume boundary. For the momentum equation these flux terms are the advective flux, the pressure flux and the viscous flux:

$$\begin{aligned} \int_{\Omega_{i,j}} \frac{\partial q}{\partial t} dS + \int_{\partial\Omega_{i,j}} (\vec{q}q/h) \cdot \vec{n} dl &= - \int_{\partial\Omega_{i,j}} \frac{1}{2}gh^2 \vec{n} dl - \int_{\Omega_{i,j}} gh\nabla z_b dS \\ + \int_{\partial\Omega_{i,j}} vh \left( 2\frac{\partial(q/h)}{\partial x} + \frac{\partial(q/h)}{\partial y} + \frac{\partial(r/h)}{\partial x} \right) \vec{n} dl \\ - \int_{\Omega_{i,j}} \frac{c_{D_b}q|\vec{q}|}{h^2} dS - \int_{\Omega_{i,j}} c_{D_w}U_{10}^2 dS \end{aligned} \quad (4.37)$$

Two terms in (4.37) require integration of a spatial derivative: the bed-slope source term and the viscous flux. For the bed-slope source term, this involves a simple derivative of the bed level  $z_b$ . The derivative in the viscous flux, however, involves the quotients  $q/h$  and  $r/h$ . The integration will involve interpolation of the variables to the faces of the sub-volumes. To make sure we do not interpolate such combined variables, but rather interpolate the single variables directly, we rewrite the integrand of the viscous flux as:

$$\begin{aligned} vh \left( 2 \frac{\partial(q/h)}{\partial x} + \frac{\partial(q/h)}{\partial y} + \frac{\partial(r/h)}{\partial x} \right) = \\ v \left( 2 \frac{\partial q}{\partial x} - 2 \frac{q}{h} \frac{\partial h}{\partial x} + \frac{\partial q}{\partial y} - \frac{q}{h} \frac{\partial h}{\partial y} + \frac{\partial r}{\partial x} - \frac{r}{h} \frac{\partial r}{\partial x} \right) \end{aligned} \quad (4.38)$$

With this, the  $q$ -momentum equation becomes:

$$\begin{aligned} \int_{\Omega_{i,j}} \frac{\partial q}{\partial t} dS + \int_{\partial\Omega_{i,j}} (\vec{q}q/h) \cdot \vec{n} dl = - \int_{\partial\Omega_{i,j}} \frac{1}{2} gh^2 \vec{n} dl - \int_{\Omega_{i,j}} gh \nabla z_b dS \\ + \int_{\partial\Omega_{i,j}} v \left( 2 \frac{\partial q}{\partial x} - 2 \frac{q}{h} \frac{\partial h}{\partial x} + \frac{\partial q}{\partial y} - \frac{q}{h} \frac{\partial h}{\partial y} + \frac{\partial r}{\partial x} - \frac{r}{h} \frac{\partial r}{\partial x} \right) \vec{n} dl \\ - \int_{\Omega_{i,j}} \frac{c_{D_b} q |\vec{q}|}{h^2} dS - \int_{\Omega_{i,j}} c_{D_w} U_{l0}^2 dS \end{aligned} \quad (4.39)$$

The momentum equation (in the present form), therefore, contains three volume-integration terms and three volume-boundary-integration terms. To perform the integrations, we again need to decide upon an integration formula, e.g. one-point Gauss, and decide upon an interpolation formula to obtain the values in the quadrature points (i.e. for one-point Gauss: the value in the integration point), just as we did for the continuity equation. For simplicity, we make the same choices here, i.e. we apply a one-point Gauss integration rule and bi-linear interpolation within cells.

For each of the terms in (4.39) we now list the different variables needed to perform the integration and the positions where these variables are needed.

#### 4.2.2.1 The time derivative

The time-derivative term  $\int_{\Omega_{i,j}} \frac{\partial q}{\partial t} dS$  is integrated using a one-point Gauss rule for each of the sub-control volumes. For this purpose, the value of  $q$  needs to be interpolated to the centres of the sub-volumes (see Figure 4.3). This, again, involves the nine-point stencil given by the coefficients of (4.9) (applied to  $q$  instead of  $h$ ).

#### 4.2.2.2 Momentum advection

The momentum advection term  $\int_{\partial\Omega_{i,j}} (\vec{q}q/h) \cdot \vec{n} dl$  is a nonlinear term involving the full vector  $\vec{q}$ , the vector component  $q$  and the water depth  $h$ . The term is integrated as a flux term along the boundary of the control volume, by summing the contributions of the sub-volume faces (multiplied by the respective unit outward normal vector to obtain the correct direction/sign of the flux contribution). The variables  $q$  and  $r$  were also required on the control volume faces for the continuity equation. The interpolation of  $h$  on the volume faces is now also required.

For explicit time-integration methods, this term simply results in a contribution to the right-hand side of the momentum equation. However, for implicit methods, this term needs to be linearized, such that the linearization coefficients can be taken up in the coefficient matrix of the momentum equation. This is discussed in more detail in Chapter 5.

#### 4.2.2.3 Pressure gradient

The pressure gradient is a flux term. It requires integration over the sub-control volume faces. For this purpose, we need to interpolate the values of the water depth  $h$  at the volume faces. This has already been realized for the momentum advection term.

Similar as for the momentum advection term, also the pressure term is nonlinear and requires a linearization in case of an implicit time integration. This is again discussed in Chapter 5.

#### 4.2.2.4 Bed-slope source term

The bed-slope source term  $\int_{\Omega_{i,j}} gh \nabla z_b dS$  is integrated over the control volume. To compute the integral, we need the water depth  $h$  at the quadrature point, i.e. for the one-point Gauss rule, we need  $h$  at the sub-control volume centre. This is already known from the other terms.

Additionally, we need to compute the bed slopes (gradients) in  $x$ - and  $y$ -direction and then interpolate these gradients to the quadrature points. The computation of the

gradients is done as follows (using  $z_b = z$  for brevity):

$$\frac{\partial z_b}{\partial x}(\xi, \eta) = \frac{z_{i+1,j} - z_{i,j}}{\Delta x} \frac{\Delta y - \eta}{\Delta y} + \frac{z_{i+1,j+1} - z_{i,j+1}}{\Delta x} \frac{\eta}{\Delta y} \quad (4.40)$$

$$\frac{\partial z_b}{\partial y}(\xi, \eta) = \frac{z_{i,j+1} - z_{i,j}}{\Delta y} \frac{\Delta x - \xi}{\Delta x} + \frac{z_{i+1,j+1} - z_{i+1,j}}{\Delta y} \frac{\xi}{\Delta x} \quad (4.41)$$

where  $\xi$  and  $\eta$  are local coordinates inside a cell, where  $\xi = \eta = 0$  at the cell centre and  $\xi = -1/2\Delta x$  and  $\eta = -1/2\Delta y$  at index  $(i,j)$  and  $\xi = 1/2\Delta x$  and  $\eta = 1/2\Delta y$  at index  $(i+1,j+1)$ . Probably we need this gradient term again as a set of interpolation coefficients in the stencil, in this case for the computation of the gradient in the sub-control volumes. Applying the sub-volume numbering as for the continuity equation, this would result in:

$$sc_{i,j,1} : \quad \frac{\partial z_b}{\partial x}_{sc} = \frac{1}{\Delta x} \left( \frac{3}{4}z_{i+1,j} - \frac{3}{4}z_{i,j} + \frac{1}{4}z_{i+1,j+1} - \frac{1}{4}z_{i,j+1} \right) \quad (4.42)$$

$$\frac{\partial z_b}{\partial y}_{sc} = \frac{1}{\Delta y} \left( \frac{3}{4}z_{i,j+1} - \frac{3}{4}z_{i,j} + \frac{1}{4}z_{i+1,j+1} - \frac{1}{4}z_{i+1,j} \right) \quad (4.43)$$

$$sc_{i,j,2} : \quad \frac{\partial z_b}{\partial x}_{sc} = \frac{1}{\Delta x} \left( \frac{3}{4}z_{i+1,j} - \frac{3}{4}z_{i,j} + \frac{1}{4}z_{i+1,j+1} - \frac{1}{4}z_{i,j+1} \right) \quad (4.44)$$

$$\frac{\partial z_b}{\partial y}_{sc} = \frac{1}{\Delta y} \left( \frac{1}{4}z_{i,j+1} - \frac{1}{4}z_{i,j} + \frac{3}{4}z_{i+1,j+1} - \frac{3}{4}z_{i+1,j} \right) \quad (4.45)$$

$$sc_{i,j,3} : \quad \frac{\partial z_b}{\partial x}_{sc} = \frac{1}{\Delta x} \left( \frac{1}{4}z_{i+1,j} - \frac{1}{4}z_{i,j} + \frac{3}{4}z_{i+1,j+1} - \frac{3}{4}z_{i,j+1} \right) \quad (4.46)$$

$$\frac{\partial z_b}{\partial y}_{sc} = \frac{1}{\Delta y} \left( \frac{3}{4}z_{i,j+1} - \frac{3}{4}z_{i,j} + \frac{1}{4}z_{i+1,j+1} - \frac{1}{4}z_{i+1,j} \right) \quad (4.47)$$

$$sc_{i,j,4} : \quad \frac{\partial z_b}{\partial x}_{sc} = \frac{1}{\Delta x} \left( \frac{1}{4}z_{i+1,j} - \frac{1}{4}z_{i,j} + \frac{3}{4}z_{i+1,j+1} - \frac{3}{4}z_{i,j+1} \right) \quad (4.48)$$

$$\frac{\partial z_b}{\partial y}_{sc} = \frac{1}{\Delta y} \left( \frac{1}{4}z_{i,j+1} - \frac{1}{4}z_{i,j} + \frac{3}{4}z_{i+1,j+1} - \frac{3}{4}z_{i+1,j} \right) \quad (4.49)$$

where it can be noted that the gradients in  $x$  direction in sub-control volumes 1 and 2 are identical, as well as those in sub-control volumes 3 and 4. Similarly, the  $y$ -gradients are identical in sub-control volumes 1 and 3 and sub-control volumes 2 and 4, respectively. For flexibility and simplicity, however, we will leave the computations in the generic form above.

Additionally, we again require the (coefficients for the) water depth  $h$  at the quadrature points. These have already been computed.

#### 4.2.2.5 Viscous flux

The viscous flux involves an integration over the (sub-)control volume faces. For this term, we need again our primary variables  $h$ ,  $q$  and  $r$  on the volume faces. These are already known from the other terms. However, we also require the gradients of all three quantities.

For both the  $q$ - and the  $r$ -momentum equation, we require  $\partial h / \partial x$  and  $\partial h / \partial y$ .

For the  $q$ -momentum equation, we additionally require  $\partial q / \partial x$ ,  $\partial q / \partial y$  and  $\partial r / \partial x$ .

For the  $r$ -momentum equation, we additionally require  $\partial r / \partial x$ ,  $\partial r / \partial y$  and  $\partial q / \partial y$ .

In total, this thus involves the full Jacobian of our state vector  $\vec{U} = (h, q, r)^T$ :

$$J = \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} \\ \frac{\partial q}{\partial x} & \frac{\partial q}{\partial y} \\ \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \end{bmatrix} \quad (4.50)$$

#### 4.2.2.6 Bed shear stress

The bed shear stress also involves a nonlinear term. This term is integrated over the control volume, again by summing contributions from the sub-volumes. For this purpose, three quantities are needed in the quadrature point(s). For the one-point Gauss integration rule, the quadrature point is again the sub-volume centre. This means the fluxes  $q$  and  $r$  need to be interpolated in the sub-control volume centres.

#### 4.2.2.7 Wind shear stress

The wind shear stress term is also integrated over the control volume. For this purpose, the wind shear stress coefficient and the wind velocity vector need to be interpolated to the sub-control volume centres.

### 4.2.3 Adding dissipation for stability

### 4.2.4 Checking well-balancedness of the momentum equation

For accurate (quasi-)steady flow predictions over variable topography an important property of a numerical scheme for the shallow-water equation is that it is *well-*

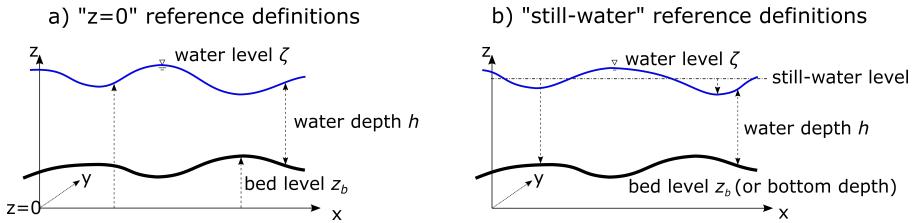


Figure 4.4: Definition of the bed level  $z_b$ , the water depth  $h$  and the water level  $\zeta$ .  
 a) "z=0" reference, where the quantities are defined with respect to  $z = 0$ ; or  
 b) "still-water" reference, where the quantities are all defined with respect to the still-water level.

*balanced*, meaning that the flux terms and the source terms need to balance each other in the momentum equations, see e.g. [Noelle et al. \(2007\)](#); [Caleffi & Valiani \(2009\)](#); [Ricchiuto \(2015\)](#); [Caleffi et al. \(2016\)](#) for some recent papers on this topic. Most research on well-balancedness in shallow-water models is conducted using Godunov methods with Riemann solvers, see [Toro & Garcia-Navarro \(2007\)](#) for an overview. These approaches are particularly suited for capturing of shocks or discontinuities, e.g. in dambreak flows or hydraulic jumps. Numerical modelling systems like Delft3D-FLOW, D-Flow Flexible Mesh, UnTRIM or TELEMAC, do not suffer from the well-balancedness issue, because of the chosen form of the pressure gradient term. The pressure (gradient) term can be written in conservative or non-conservative form. The conservative form results in two terms (including a non-conservative product for the bed-slope reaction term, in case of variable topography):

- Pressure gradient in conservative form:  $\nabla \left( \frac{gh^2}{2} \right) + gh\nabla z_b$
- Pressure gradient in non-conservative form:  $gh\nabla\zeta$

with  $g$ , the gravitational acceleration,  $h$  the water depth,  $\zeta$  the water level,  $z_b$  the bed level, see Figure 4.4. The figure also shows that several definitions are possible. The "z=0" reference option is typically used in river (or other more inland) applications, whereas the "still-water" reference option is more common in ocean or estuarine applications. Both approaches/definitions have their own advantages/disadvantages when it comes to round-off errors in the computations.

The two pressure-gradient form above are equivalent in the continuous formulation, when all variables vary smoothly. However, to obtain a well-balanced scheme, these terms also need to be equivalent at a discrete level. The question therefore is:

Do we have  $gh\nabla\zeta - gh\nabla z_b = gh\nabla h = g\nabla(h^2/2)$  at discrete level?

To verify this, we consider the two formulations in one dimension:

$$\int_{\Omega} \frac{\partial}{\partial x} \left( \frac{gh^2}{2} \right) dV + \int_{\Omega} gh \frac{\partial z_b}{\partial x} dV = \quad (4.51)$$

$$\oint_{\partial\Omega} \left( \frac{gh^2}{2} \right) (\vec{n}_{scvf}, \vec{n}_x) dS + \int_{\Omega} gh \frac{\partial z_b}{\partial x} dV = \quad (4.52)$$

$$\sum_{scvf \in \partial\Omega} \frac{g}{2} h_{scvf}^2 (\vec{n}_{scvf}, \vec{n}_x) + \sum_{scv \in \Omega} A_{scv} gh_{scv} \frac{\partial z_b}{\partial x} = \quad (4.53)$$

$$\begin{aligned} & \frac{g}{2} (h_{scvf,R}^2 - h_{scvf,L}^2) + \\ & \frac{g\Delta x}{8} (h_1 + 3h_2) \frac{(z_b)_2 - (z_b)_1}{\Delta x} \frac{g\Delta x}{8} (3h_2 + h_3) \frac{(z_b)_3 - (z_b)_2}{\Delta x} = \end{aligned} \quad (4.54)$$

$$\begin{aligned} & \frac{g}{2} \left( \frac{h_2 + h_3}{2} \right)^2 - \frac{g}{2} \left( \frac{h_1 + h_2}{2} \right)^2 + \\ & \frac{g}{8} (h_1 + 3h_2) ((z_b)_2 - (z_b)_1) + \frac{g}{8} (3h_2 + h_3) ((z_b)_3 - (z_b)_2) = \end{aligned} \quad (4.55)$$

$$\begin{aligned} & \frac{g}{8} (h_2^2 + 2h_2h_3 - 2h_1h_2 - h_1^2) + \\ & \frac{g}{8} (h_1 + 3h_2) ((z_b)_2 - (z_b)_1) + \frac{g}{8} (3h_2 + h_3) ((z_b)_3 - (z_b)_2) \end{aligned} \quad (4.56)$$

And similarly:

$$\int_{\Omega} gh \frac{\partial \zeta}{\partial x} dV = \quad (4.57)$$

$$\sum_{scv \in \Omega} A_{scv} gh_{scv} \frac{\partial \zeta}{\partial x} = \quad (4.58)$$

$$\begin{aligned} & \frac{g\Delta x}{8} (h_1 + 3h_2) \frac{h_2 + (z_b)_2 - h_1 - (z_b)_1}{\Delta x} + \\ & \frac{g\Delta x}{8} (3h_2 + h_3) \frac{h_3 + (z_b)_3 - h_2 - (z_b)_2}{\Delta x} = \end{aligned} \quad (4.59)$$

$$\begin{aligned} & \frac{g}{8} (h_1 + 3h_2) (h_2 + (z_b)_2 - h_1 - (z_b)_1) + \\ & \frac{g}{8} (3h_2 + h_3) (h_3 + (z_b)_3 - h_2 - (z_b)_2) \end{aligned} \quad (4.60)$$

Now the question is: is (4.56) equal to (4.60)? When comparing these two equations, one can see that the terms involving  $z_b$  cancel on both sides. What remains is to check whether:

$$\frac{g}{8} (h_3^2 + 2h_2h_3 - 2h_1h_2 - h_1^2)$$

equals

$$\frac{g}{8} (h_1 + 3h_2) (h_2 - h_1) + \frac{g}{8} (3h_2 + h_3) (h_3 - h_2)$$

Rewriting the second expression (from (4.60)), we obtain:

$$\begin{aligned} & \frac{g}{8} (h_1 + 3h_2) (h_2 - h_1) + \frac{g}{8} (3h_2 + h_3) (h_3 - h_2) = \\ & \frac{g}{8} (3h_2^2 - 2h_1h_2 - h_1^2) + \frac{g}{8} (h_3^2 + 2h_2h_3 - 3h_2^2) = \\ & \frac{g}{8} (h_3^2 + 2h_2h_3 - 2h_1h_2 - h_1^2) \end{aligned} \quad (4.61)$$

(4.61) equals the expression from (4.56), i.e. the conservative and non-conservative pressure-gradient formulations are equal also on discrete level. This is due to the assumed linearity of all variables within cells. Due to the fact that the non-conservative formulation only contains one term (involving the water level gradient), there is no "well-balancing" problem. Due to the equivalence, the conservative formulation is now – per definition – well-balanced.

NOTE: a similar, yet more involved, derivation can be done for demonstrating well-balancedness of the scheme in 2D.

## 4.2.5 Other open questions



## CHAPTER 5

### TIME INTEGRATION OF THE TWO-DIMENSIONAL SHALLOW-WATER EQUATIONS

#### 5.1 Introduction

#### 5.2 Solution strategies

Predictor-corrector

Semi-implicit discretization with substitution of the momentum equation in the continuity equation and solving for the pressure (conform D-Flow FM and UnTRIM).

ADI

Linear Multistep methods (e.g. Runge-Kutta)

Non-linear (and linearization) approaches, i.e. Newton/Picard.

## 5.3 Time integration approaches

### 5.3.1 Explicit methods

#### 5.3.1.1 Explicit time integration for the collocated vertex-centered finite-volume method

### 5.3.2 (Semi-)Implicit methods

#### 5.3.2.1 Implicit time integration for the collocated vertex-centered finite-volume method

# CHAPTER 6

---

## BOUNDARY CONDITIONS

### 6.1 Introduction

In this section we describe the treatment of open and closed boundaries. We consider the different types of boundaries and the integration of the boundary conditions in the solution algorithm.

### 6.2 Boundary positioning

An important choice is the positioning of the boundaries. For modelers it is important that the boundary positioning is intuitive and that the resulting values of variables on the boundaries can be linked to the boundary input in a logical and preferably simple way.

The domain boundaries can be chosen to coincide either with the grid lines or with the control volume boundaries. The latter choice would mean that the domain boundary lies in the interior of computational cells and that boundary cells (containing the domain boundaries) have both interior nodes (either one or two) and virtual nodes (either three or two), see Figure 6.1 for a comparison of both approaches.

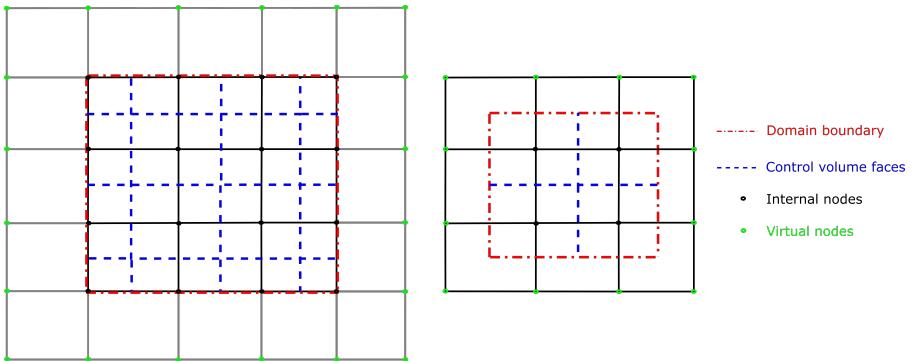


Figure 6.1: Illustration of the position of the domain boundary (red dash-dotted line), with respect to the computational grid (black lines) and to the control volumes (blue dashed lines) in a rectangular 2d grid.

In the first prototype, the virtual vertices/nodes are defined as part of the main grid numbering, i.e. they have numbering between zero and the number of vertices/nodes. The distinction between internal and virtual nodes is then made based on masking of the nodes.

In Figure 6.1 shows a more realistic grid. The grid can consist of:

- Inner nodes: the control volumes for these nodes are fully included in the computational domain.
- Virtual nodes: the control volumes for these nodes are not fully included in the computational domain, but are still part of the numerical solution. These volumes are called boundary volumes.
- Inactive nodes = the control volumes for these grid points are not defined nor are part of the numerical solution.

Boundary conditions can be applied in 3 steps:

- Loop over all internal control volumes (step 1): In this step, the contributions of the internal control volumes to the system matrix and the right-hand side are computed (for the mass matrix and the spatial discretization matrix)
- Loop over the boundary volumes (step 2): In this step, the contributions of the boundary volumes to the system matrix and the right hand side are computed

- Impose boundary conditions (step 3): In this step, the boundary conditions are imposed. Dirichlet boundary conditions values are imposed at the virtual nodes. Neumann boundary conditions values are imposed by relating inner nodes to virtual nodes. For example, a zero concentration gradient boundary condition is imposed by setting the difference between inner and virtual nodes to 0.

Modelers typically use lines to define boundary conditions with time-varying values defined at the line vertices. A possible approach for assigning values to virtual nodes can be: (1) projecting the virtual nodes on the boundary line, (2) computing the interpolated values on the projected points, and (3) assigning the interpolated values to the virtual nodes.

... TODO: Describe the mart's cut-cell method ...

## 6.3 Boundary procedure for 1D linearized shallow water equations

Here we detail the boundary procedure for the linearized shallow water equations:

$$\frac{\partial \zeta}{\partial t} + h \frac{\partial u}{\partial x} = 0 \quad (6.1)$$

$$\frac{\partial u}{\partial t} + g \frac{\partial \zeta}{\partial x} = 0 \quad (6.2)$$

Where  $\zeta$  is the water level (or free-surface level) and  $h$  is the water depth and  $u$  is the flow velocity in  $x$ -direction. The water depth  $h$  is related to the water level through the bed level  $z_b$  as:  $h = \zeta - z_b$ . In the linearized shallow water equations  $h$  is assumed constant.

In step 2 of the procedure described above, we establish a relation between the virtual node 0 and the inner node 1 by discretizing the equations at the boundary face. Assuming a uniform space discretization  $\Delta x$  and an implicit time discretization, the following equations are obtained:

$$\frac{1}{2} \left( \frac{\zeta_0^{n+1} - \zeta_0^n}{\Delta t} + \frac{\zeta_1^{n+1} - \zeta_1^n}{\Delta t} \right) + h \frac{u_1^{n+1} - u_0^{n+1}}{\Delta x} = 0 \quad (6.3)$$

$$\frac{1}{2} \left( \frac{u_0^{n+1} - u_0^n}{\Delta t} + \frac{u_1^{n+1} - u_1^n}{\Delta t} \right) + g \frac{\zeta_1^{n+1} - \zeta_0^{n+1}}{\Delta x} = 0 \quad (6.4)$$

For clarity, the contribution to the system matrix is here separated into the time contribution (the mass matrix) and the spatial contribution. The contributions of the virtual node and the first inner node end up in the first and second row of the mass matrix, respectively:

$$\begin{bmatrix} \frac{1}{2\Delta t} & 0 & \frac{1}{2\Delta t} & 0 & 0 & 0 \\ 0 & \frac{1}{2\Delta t} & 0 & \frac{1}{2\Delta t} & 0 & 0 \\ \frac{1}{8\Delta t} & 0 & \frac{3}{4\Delta t} & 0 & \frac{1}{8\Delta t} & 0 \\ 0 & \frac{1}{8\Delta t} & 0 & \frac{3}{4\Delta t} & 0 & \frac{1}{8\Delta t} \\ 0 & 0 & \frac{1}{8\Delta t} & 0 & \frac{3}{4\Delta t} & 0 \\ 0 & 0 & 0 & \frac{1}{8\Delta t} & 0 & \frac{3}{4\Delta t} \dots \end{bmatrix} \begin{bmatrix} \zeta_0^{n+1} \\ u_0^{n+1} \\ \zeta_1^{n+1} \\ u_1^{n+1} \\ \zeta_2^{n+1} \\ u_2^{n+1} \end{bmatrix} \quad (6.5)$$

The spatial discretization matrix is the following:

$$\begin{bmatrix} 0 & -\frac{h}{\Delta x} & 0 & \frac{h}{\Delta x} & 0 & 0 \\ -\frac{g}{\Delta x} & 0 & \frac{g}{\Delta x} & 0 & 0 & 0 \\ 0 & -\frac{h}{2\Delta x} & 0 & 0 & 0 & \frac{h}{2\Delta x} \\ 0 & 0 & -\frac{g}{2\Delta x} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{h}{2\Delta x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{g}{2\Delta x} & 0 \dots \end{bmatrix} \begin{bmatrix} \zeta_0^{n+1} \\ u_0^{n+1} \\ \zeta_1^{n+1} \\ u_1^{n+1} \\ \zeta_2^{n+1} \\ u_2^{n+1} \end{bmatrix} \quad (6.6)$$

The discretization of the equations at node 0 gives two equations for the two unknowns  $\zeta_0^{n+1}$ ,  $u_0^{n+1}$ . However, we must impose boundary conditions to include the effects outside of the model domain.

Dirichlet boundary condition for  $u$  can be imposed by means of polylines with time-varying values at the polyline vertices. The polyline might lie between node 0 and 1 at a non-dimensional distance  $\alpha$  from node 1. To match the value  $u_b$  defined on the polyline, the following equation must hold for  $u$  at node 0 and 1 for every time  $t$ :

$$\alpha u_0 + (1 - \alpha) u_1 = u_b \quad (6.7)$$

Replacing one of the discretized model equations at the boundary will introduce some loss of the physical behavior modeled by the equations. A possibly better approach is to appropriately combine the two discretized model equations at the boundary/virtual node to mimic this physical behavior. For the linearized shallow-water equations, such a formulation can be based on the following relation, which represents an outgoing characteristic at the left boundary:

$$\sqrt{gh}\zeta = hu \quad (6.8)$$

For imposing the latter physical relation, the first row of the matrices above is multiplied by  $\sqrt{gh}$  and added to the second row multiplied by  $-h$ . As a result, the first row can now be nullified and used for imposing the dirichlet boundary condition that relates  $u_0$  and  $u_1$ .

The mass matrix contributions after imposing the boundary conditions become:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\sqrt{gh}}{2\Delta t} & -\frac{h}{2\Delta t} & \frac{\sqrt{gh}}{2\Delta t} & -\frac{h}{2\Delta t} & 0 & 0 \\ \frac{1}{8\Delta t} & 0 & \frac{3}{4\Delta t} & 0 & \frac{1}{8\Delta t} & 0 \\ 0 & \frac{1}{8\Delta t} & 0 & \frac{3}{4\Delta t} & 0 & \frac{1}{8\Delta t} \\ 0 & 0 & \frac{1}{8\Delta t} & 0 & \frac{3}{4\Delta t} & 0 \\ 0 & 0 & 0 & \frac{1}{8\Delta t} & 0 & \frac{3}{4\Delta t} \dots \end{bmatrix} \begin{bmatrix} \xi_0^{t+1} \\ u_0^{t+1} \\ \xi_1^{t+1} \\ u_1^{t+1} \\ \xi_2^{t+1} \\ u_2^{t+1} \end{bmatrix} \quad (6.9)$$

The spatial discretization contributions after imposing the boundary conditions become:

$$\begin{bmatrix} 0 & \alpha & 0 & 1-\alpha & 0 & 0 \\ h\frac{g}{\Delta x} & -\sqrt{gh}\frac{h}{\Delta x} & -h\frac{g}{\Delta x} & \sqrt{gh}\frac{h}{\Delta x} & 0 & 0 \\ 0 & -\frac{h}{2\Delta x} & 0 & 0 & 0 & \frac{h}{2\Delta x} \\ 0 & 0 & -\frac{g}{2\Delta x} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{h}{2\Delta x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{g}{2\Delta x} & 0 \dots \end{bmatrix} \begin{bmatrix} \xi_0^{t+1} \\ u_0^{t+1} \\ \xi_1^{t+1} \\ u_1^{t+1} \\ \xi_2^{t+1} \\ u_2^{t+1} \end{bmatrix} \quad (6.10)$$



## CHAPTER 7

---

### STABILITY ANALYSIS

Von Neumann Stability analysis

Fourier analysis

Amplification factor / amplitude errors (damping)

Phase errors.



# CHAPTER 8

---

## IMPLEMENTATION ASPECTS

### 8.1 Introduction

### 8.2 Class hierarchy

The following hierarchy of classes is probably needed:

```
Problem
  EquationSystem
  SolutionStrategy
    Equation
      Term
        Variables      //($h, q, r, \nabla z_b, \nu$, etc.)
        IntegralType   //((sub-volume/sub-volume face)
        IntegrationRule //((e.g. one-point Gauss, including number
                           // and position of quadrature points)
        InterpolationType //((e.g. bi-linear interpolation)
TimeIntegration //((explicit or (semi-)implicit)
BoundaryConditions //((should this be part of the system,
                      // or of a single equation?))
```

We start from an `EquationSystem` object that contains a vector of equation objects, one for each equation in the system of equations. Additionally, the `EquationSystem` object has knowledge about *how* to solve the system of equations, i.e. it has reference to a `SolutionStrategy`. The `SolutionStrategy` contains information about the order in which the equations are to be solved: in a particular order or simultaneously, and about possible dependencies between consecutive steps.

A call to `EquationsSystem.solve()` will then result in the execution of the procedure described in the `SolutionStrategy`, involving one or more calls to:

`Equation.assemble()`

for each of the equations in the `EquationSystem`.

The procedure in the `assemble()` function of the `Equation` class is then as follows:

```
Loop over all internal computational // (i.e. excluding virtual/ghost cells
For each cell:
  Loop over the $N$ sub-control volumes // (for quadrilaterals, $N=4, for triangles)
    For each sub-volume:
      Loop over each term in the equation:
        For each term:
          term.integrate(term.IntegrationRule, term.IntegralType) // which calls
            local\_coef\_matrix = term.interpolate(term.InterpolationType)
```

where the `interpolate` function returns the local interpolation coefficients for the surrounding  $N$  vertices (4 for quads, 3 for triangles), from which the variables in the term, required in the quadrature points in the cell, can be computed by multiplying these coefficients with the corresponding nodal values of the variables in the term (or combinations thereof).

## 8.3 Parallelization

Notes on MPI and domain decomposition

## REFERENCES

- Busto, S., Dumbser, M., & Río-Martín, L. (2021). Staggered semi-implicit hybrid finite volume/finite element schemes for turbulent and non-newtonian flows. *Mathematics*, 9(22).
- Caleffi, V. & Valiani, A. (2009). Well-Balanced Bottom Discontinuities Treatment for High-Order Shallow Water Equations WENO Scheme. *Journal of Engineering Mechanics*, 135(7), 684–696.
- Caleffi, V., Valiani, A., & Li, G. (2016). A comparison between bottom-discontinuity numerical treatments in the DG framework. *Applied Mathematical Modelling*, 40, 7516–7531.
- McBride, D., Croft, N., & Cross, M. (2007). Finite volume method for the solution of flow on distorted meshes. *International Journal of Numerical Methods for Heat & Fluid Flow*, 17(2), 213–239.
- Noelle, S., Xing, Y., & Shu, C.-W. (2007). High-order well-balanced finite volume WENO schemes for shallow water equation with moving water. *Journal of Computational Physics*, 226(1), 29–58.
- Ricchiuto, M. (2015). An explicit residual based approach for shallow water flows. *Journal of Computational Physics*, 280, 306–344.
- Toro, E. F. & Garcia-Navarro, P. (2007). Godunov-type methods for free-surface shallow flows: A review. *Journal of Hydraulic Research*, 45(6), 736–751.



## APPENDIX A

### GRID INSENSITIVITY OF THE VERTEX-CENTERED FINITE VOLUME SCHEME

#### A.1 Introduction

The vertex-centred finite volume scheme requires integration of all terms in the equations over the control volumes around vertices. The control volume around a vertex consists of sub-control volumes that are part of the cells surrounding the vertex. Depending on the grid type/structure the number of connected cells can vary. To simplify the handling of grids of different cell types (quadrilateral, triangular, hexagonal, etc.) and possible hybrid grids, the integration over control volumes is replaced by integration over sub-control volumes, where the latter step is handled on a per-cell basis. Each cell contains  $N$  sub-control volumes, where  $N$  is the number of vertices of that cell, i.e. each vertex has as corresponding sub-control volume (scv). Additionally, the connections between the sub-control volumes (inside a cell) form the sub-control volume faces (scvf's).

The integration over a scv or scvf contained in a cell, results in contributions that are expressed in terms of the values of the corresponding vertices of that cell. These contributions can either be explicit as coefficients times the known values in the vertices, or they can be implicit as coefficients that need to be assembled in the global system of equations that needs to be solved for all degrees of freedom (the values of

the primary variables in the cell vertices). These (explicit and implicit) contributions are then added to the corresponding matrix and/or right-hand side entries for the corresponding (control volumes of the) vertices of that cell (for each term in the equations). After which the system of equations can be solved.

For the basic one-point Gauss integration rule (per scv or scvf), the integration involves interpolating the values in the quadrature points, being either the scv centres (for the volume-type integrals of Section 4.2.1) or the scvf midpoints (for the flux-type integrals of Section 4.2.1). The interpolations involve the surrounding vertices of the cell under consideration. The interpolation then results in a local coefficient matrix expressing the interpolation weights of the surrounding vertices for each of the quadrature points. It turns out that the local coefficient matrix can be expressed in a generic way that holds for all grid cell shapes. This is due to the fact that all the supporting points building the scv's are based on simple averaging (central approximations). This is illustrated below for general quadrilateral and triangular grids, for both scv's and scvf's, after which the general formulations are provided.

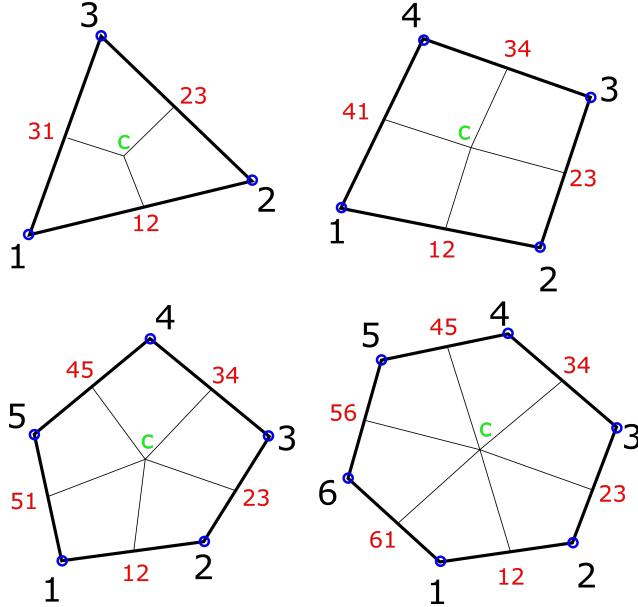


Figure A.1: Illustration of the points involved in the interpolation over sub-control volumes in different two-dimensional grids. a) triangular grids; b) quadrilateral grids; c) pentagonal grids; d) hexagonal grids. The thick black lines form the primary computational grid. The blue circles are the primary grid vertices, numbered 1-N, where N is the number of vertices in the cell. Point "c" represents the centre of the cell. Cell-face midpoints (e.g. 12) are labeled by the corresponding connected vertices for that face (in this case vertices 1 and 2). The sub-control volumes are then always built up from four support points: a primary vertex, the cell centre and two adjacent cell-face midpoints.

## A.2 Quadrilateral grids

Figure A.1b) shows a quadrilateral cell and the constructed scv's and scvf's. Moreover, the cell centre  $c$  is shown and the scvf midpoints, number by their connected vertices are shown. The formulas for both the coordinates of the points, but also for interpolating values for a generic value  $v$  of some quantity in these points is given as:

$$v_c = \frac{1}{4} (v_1 + v_2 + v_3 + v_4) \quad (\text{A.1})$$

where  $v_1, v_2, v_3$  and  $v_4$  are the values in the four vertices, and

$$v_{12} = \frac{1}{2} (v_1 + v_2) \quad (\text{A.2})$$

with similar expressions for the other primary grid cell face midpoints.

Using these expressions, we can now derive interpolation formulas both for the scv centres and for the scvf midpoints.

### Sub-control volumes

The value in a scv center is then computed as (for brevity we only show the formulation for  $scv_1$ , connected to vertex 1):

$$v_{scv_1} = \frac{1}{4} (v_1 + v_c + v_{12} + v_{41}) \quad (\text{A.3})$$

which – by inserting the expressions for  $v_c, v_{12}$  and  $v_{41}$  – becomes:

$$v_{scv_1} = \frac{1}{4} \left( v_1 + \frac{1}{4} (v_1 + v_2 + v_3 + v_4) + \frac{1}{2} (v_1 + v_2) + \frac{1}{2} (v_1 + v_4) \right) \quad (\text{A.4})$$

$$v_{scv_1} = \frac{1}{16} (9v_1 + 3v_2 + 1v_3 + 3v_4) \quad (\text{A.5})$$

This exercise can be repeated for the other three vertices in the cell to obtain their coefficients, which of course turn out to be identical, just in a different order. The resulting local interpolation coefficients can be written in matrix-vector form (see also Section 4.2.1):

$$\vec{v}_{scv} = \frac{1}{16} \begin{bmatrix} 9 & 3 & 1 & 3 \\ 3 & 9 & 3 & 1 \\ 1 & 3 & 9 & 3 \\ 3 & 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (\text{A.6})$$

### Sub-control volume faces

Similarly, we can obtain the interpolation formulas in the sub-control volume face midpoints. For the scvf connecting points 12 and C (let's call it " $scvf_{12-c}$ " in Figure A.1b), we obtain that the centre value is computed as:

$$v_{scvf_{12-c}} = \frac{1}{2} (v_{12} + v_c) \quad (\text{A.7})$$

Substituting again the expressions for  $v_{12}$  and  $v_c$ , we obtain:

$$v_{scvf_{12-c}} = \frac{1}{2} \left( \frac{1}{2} (v_1 + v_2) + \frac{1}{4} (v_1 + v_2 + v_3 + v_4) \right) \quad (\text{A.8})$$

$$v_{scvf_{12-c}} = \frac{1}{8} (3v_1 + 3v_2 + 1v_3 + 1v_4) \quad (\text{A.9})$$

Just as for the scv's, this exercise can be repeated for the other three vertices in the cell to obtain their coefficients, which of course turn out to be identical, just in a different order. The resulting local interpolation coefficients can be written in matrix-vector form (see also Section 4.2.1):

$$\vec{v}_{scvf} = \frac{1}{8} \begin{bmatrix} 3 & 3 & 1 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (\text{A.10})$$

Here it should be noted that the order of the scvf's is by increasing indices, i.e.  $v_{scvf_{12-c}}$ ,  $v_{scvf_{23-c}}$ ,  $v_{scvf_{34-c}}$ ,  $v_{scvf_{41-c}}$ .

## A.3 Triangular grids

A similar derivation can be done for triangular grids (see Figure A.1a). Now the cell has only three vertices and the formulations read as follows:

$$v_c = \frac{1}{3} (v_1 + v_2 + v_3) \quad (\text{A.11})$$

and

$$v_{12} = \frac{1}{2} (v_1 + v_2) \quad (\text{A.12})$$

with similar expressions for the other primary grid cell face midpoints.

Using these expressions, we can now derive interpolation formulas both for the scv centres and for the scvf midpoints.

### Sub-control volumes

The value in a scv center is then computed as (for brevity we only show the formulation for scv<sub>1</sub>, connected to vertex 1):

$$v_{scv_1} = \frac{1}{4} (v_1 + v_c + v_{12} + v_{31}) \quad (\text{A.13})$$

which – by inserting the expressions for  $v_c$ ,  $v_{12}$  and  $v_{31}$  – becomes:

$$v_{scv_1} = \frac{1}{4} \left( v_1 + \frac{1}{3} (v_1 + v_2 + v_3) + \frac{1}{2} (v_1 + v_2) + \frac{1}{2} (v_1 + v_3) \right) \quad (\text{A.14})$$

$$v_{scv_1} = \frac{1}{24} (14v_1 + 5v_2 + 5v_3) \quad (\text{A.15})$$

This exercise can be repeated for the other three vertices in the cell to obtain their coefficients, which of course turn out to be identical, just in a different order. The resulting local interpolation coefficients can be written in matrix-vector form:

$$\vec{v}_{scv} = \frac{1}{24} \begin{bmatrix} 14 & 5 & 5 \\ 5 & 14 & 5 \\ 5 & 5 & 14 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (\text{A.16})$$

### Sub-control volume faces

Similarly, we can obtain the interpolation formulas in the sub-control volume face midpoints. For the scvf connecting points 12 and  $C$  (let's call it " $scvf_{12-c}$ " in Figure A.1a), we obtain that the centre value is computed as:

$$v_{scvf_{12-c}} = \frac{1}{2} (v_{12} + v_c) \quad (\text{A.17})$$

Substituting again the expressions for  $v_{12}$  and  $v_c$ , we obtain:

$$v_{scvf_{12-c}} = \frac{1}{2} \left( \frac{1}{2} (v_1 + v_2) + \frac{1}{3} (v_1 + v_2 + v_3) \right) \quad (\text{A.18})$$

$$v_{scvf_{12-c}} = \frac{1}{12} (5v_1 + 5v_2 + 2v_3) \quad (\text{A.19})$$

Just as for the scv's, this exercise can be repeated for the other three scvf centres in the cell to obtain their coefficients, which of course turn out to be identical, just in a different order. The resulting local interpolation coefficients can be written in matrix-vector form:

$$\vec{v}_{scv} = \frac{1}{12} \begin{bmatrix} 5 & 5 & 2 \\ 2 & 5 & 5 \\ 5 & 2 & 5 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (\text{A.20})$$

Here it should be noted that the order of the scvf's is by increasing indices, i.e.  $v_{scvf_{12-c}}$ ,  $v_{scvf_{23-c}}$ ,  $v_{scvf_{31-c}}$ .

## A.4 General formulation

It can be seen that the resulting interpolation coefficients for quadrilateral and triangular grids are independent of the local grid coordinates. This is due to the fact that all approximations of the support points of the scv involve simple averages (i.e. 1/4, 1/2, 1/3) of surrounding values. Under the assumption that the vertices of a cell are numbered in counter-clockwise orientation, one can derive a general formulation for these coefficients, written as a function of the number of vertices  $N$  of a cell. This formulation can be derived in a similar way as the coefficients were constructed above. For this we note the following:

Property 1 A sub-control volume always has four corners, independent of the primary grid cell type.

Property 2 The interpolation of a sub-control volume face midpoint always involves a central average (1/2).

Property 3 The interpolation of the cell centre value involves a division by  $1/N$ .

Property 4 Vertices of a cell are numbered in counter-clockwise orientation.

Using these properties one can derive coefficient matrices for a general cell with  $N$  vertices, both for the scv and for the scvf.

### A.4.1 Sub-control volumes

For sub-control volumes, there are three possible values for the coefficients, independent of the grid cell type:

1. The value for the vertex corresponding to the scv:  $c_{scv,1}$
2. The value for the two direct neighbouring vertices:  $c_{scv,2}$
3. The value for all other (possible) vertices:  $c_{scv,3}$

These can be computed as:

$$c_{scv,1} = \frac{1}{2} + \frac{1}{N}$$

$$c_{scv,2} = \frac{1}{8} + \frac{1}{N}$$

$$c_{scv,3} = \frac{1}{N}$$

By substituting  $N = 3$  or  $N = 4$ , it can be verified that the coefficients derived for triangular and quadrilateral grids (for the scv interpolations) are obtained, respectively.

### A.4.2 Sub-control volume faces

Similarly, for sub-control volume faces, there are two possible values for the coefficients, independent of the grid cell type:

1. The value for the two direct neighbour vertices of the scvf  $c_{scvf,1}$
2. The value for all other (possible) vertices  $c_{scvf,2}$

These can be computed as:

$$c_{scvf,1} = \frac{1}{4} + \frac{1}{2N}$$

$$c_{scvf,2} = \frac{1}{2N}$$

By substituting  $N = 3$  or  $N = 4$ , it can be verified that the coefficients derived for triangular and quadrilateral grids (for the scvf interpolations) are obtained, respectively.

## APPENDIX B

### APPLICATION: SCALAR 1D ADVECTION

#### B.1 Introduction

To allow a simpler illustration of the collocated, vertex-centered, finite-volume scheme, we consider a simple test model for scalar advection in one dimension. Even though the problem is strictly one-dimensional, we consider the generalization of the method for two-dimensional grids. For simplicity, we focus only on the interior domain and leave out the effect of boundary conditions.

#### B.2 Equations

We consider the advection of a scalar quantity  $c$  in one dimension (the  $x$ -direction) with a constant flow velocity  $a$ . The governing equation reads:

$$\frac{\partial c}{\partial t} + a \frac{\partial c}{\partial x} = 0 \quad (\text{B.1})$$

with  $a > 0$ , indicating flow in positive  $x$ -direction.

## B.3 Numerical method

We solve the equation on a simple two-dimensional Cartesian grid with uniform grid sizes ( $\Delta x, \Delta y$ ). The concentration  $c$  is stored in the cell vertices/nodes. The equation is then solved by integrating the equations over control volumes  $\Omega$  that are constructed by connecting the cell centroids with the face-midpoints, sometimes referred to as a *centroid dual mesh*. For the Cartesian mesh, this simply consists of square control volumes around the cell vertices, where the nodes of dual mesh (and thus the corners of the control volumes) are the cell centroids of the primary mesh, see Figure B.1.

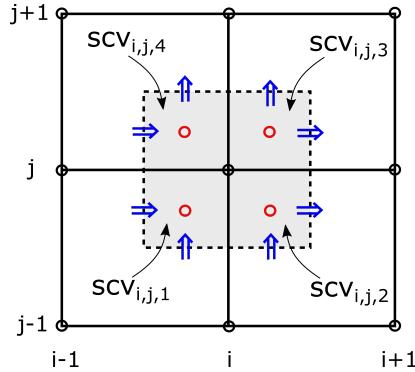


Figure B.1: Illustration of the placement of variables in vertices and of the control volume in a rectangular 2d grid. The red circles show the quadrature points of the one-point Gauss integration rule for the four sub-volumes of control volume  $\Omega_{i,j}$ . The blue arrows correspond to the fluxes across the (dashed) boundary  $\partial\Omega$  of the control volume.

The integrated equation reads:

$$\iint_{\Omega} \frac{\partial c}{\partial t} dV + a \iint_{\Omega} \frac{\partial c}{\partial x} dV = 0 \quad (\text{B.2})$$

Applying a semi-discretization in time and Gauss' divergence theorem (in gradient form) to the flux term, we obtain:

$$\iint_{\Omega} \frac{c^{n+1} - c^n}{\Delta t} dV + a \oint_{\partial\Omega} c(n_f, n_x) dS = 0 \quad (\text{B.3})$$

where  $n$  indicates the time level,  $n_f$  the unit outward normal on a (sub-control) volume face as part of the the control volume boundary  $\partial\Omega$  and  $n_x$  is the normal vector in the

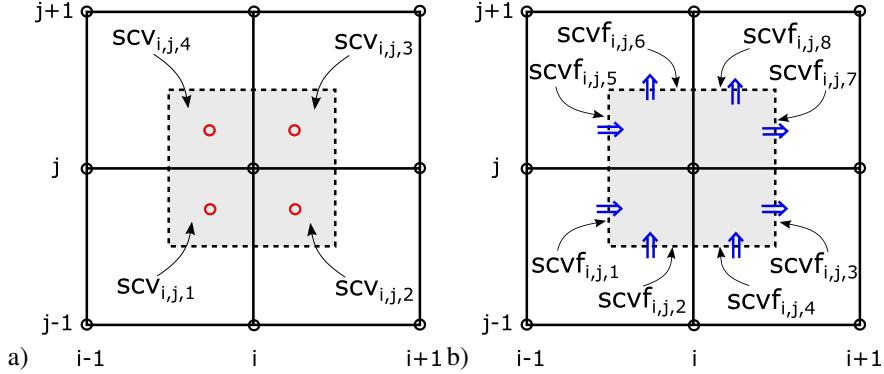


Figure B.2: a) Illustration of the sub-volumes of the control volume in a rectangular 2D grid. b) Illustration of the sub-volume edges of the control volume in a rectangular 2D grid. The red circles show the quadrature points of the one-point Gauss integration rule for the four sub-volumes of control volume  $\Omega_{i,j}$ . The blue arrows correspond to the fluxes across the (dashed) boundary  $\partial\Omega$  of the control volume.

$x$ -coordinate direction. The inner product  $(n_f, n_x)$  can have the following values:

$$(n_f, n_x) = \begin{cases} 1 & \text{for segments of } \partial\Omega \text{ with outward flux in positive } x\text{-dir.} \\ 0 & \text{for segments of } \partial\Omega \text{ with outward flux in positive } y\text{-dir.} \\ -1 & \text{for segments of } \partial\Omega \text{ with outward flux in negative } x\text{-dir.} \\ 0 & \text{for segments of } \partial\Omega \text{ with outward flux in negative } y\text{-dir.} \end{cases} \quad (\text{B.4})$$

Note that for two-dimensional scalar advection:

$$\frac{\partial c}{\partial t} + a_x \frac{\partial c}{\partial x} + a_y \frac{\partial c}{\partial y} = 0 \quad (\text{B.5})$$

the zeros in (B.4), would be 1 and -1, respectively.

Now we divide the control volume  $\Omega$  in four sub-control volumes (scv) and its boundary  $\partial\Omega$  in (eight) sub-control volume faces (scvf), as shown in Figure B.1:

$$\frac{1}{\Delta t} \sum_{scv \in \Omega} \iint_{scv} c^{n+1} dV - \frac{1}{\Delta t} \sum_{scv \in \Omega} \iint_{scv} c^n dV + a \sum_{scvf \in \partial\Omega} \oint_{scvf} c(n_f, n_x) dS = 0 \quad (\text{B.6})$$

The integrals over the scv and scvf are now (for simplicity) approximated using a one-point Gauss quadrature rule:

$$\frac{1}{\Delta t} \sum_{scv \in \Omega} c_{scv,c}^{n+1} A_{scv} - \frac{1}{\Delta t} \sum_{scv \in \Omega} c_{scv,c}^n A_{scv} + a \sum_{scvf \in \partial\Omega} c_{scvf,c}(n_f, n_x) I_{scvf} = 0 \quad (\text{B.7})$$

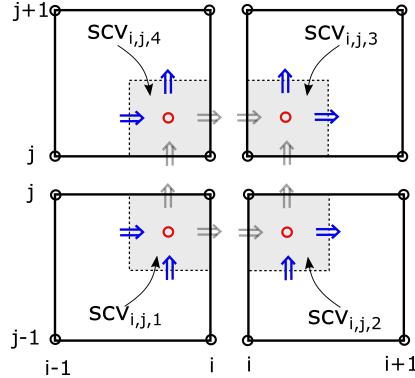


Figure B.3: Illustration of the (split) sub-control volumes (scv) and sub-control volume faces (scvf) of the control volume in a rectangular 2D grid. The red circles show the quadrature points of the one-point Gauss integration rule for the four sub-volumes of control volume  $\Omega_{i,j}$ . The blue arrows correspond to the fluxes across the (dashed) boundary  $\partial\Omega$  of the control volume. The gray arrows are the fluxes across the sub-control volume faces that are internal to the control volume. After accumulation of the sub-control volume fluxes over the whole control volume boundary, these fluxes drop out.

where the subscripts  $scv,c$  and  $scvf,c$  refer to the  $scv$  and  $scvf$  centroids respectively. Additionally,  $l_{scvf}$  is the  $scvf$  length. For the Cartesian grid, this is  $\Delta x/2$  or  $\Delta y/2$ , depending on the face orientation. Due to the multiplication with the inner-product  $(n_f, n_x)$ , only  $scvf$  contributions with  $\Delta y/2$  remain (because we consider only advection in  $x$ -direction).

Now we use bilinear interpolation per cell/element, to obtain the values of the quantity  $c$  in  $scv,c$  and  $scvf,c$ . The computation of the values in the sub-control volume centres, then involves a local coefficient matrix. The expression for the  $scv,c$  reads:

$$c_{scv,c,k} = \frac{1}{16} \begin{bmatrix} 9 & 3 & 1 & 3 \\ 3 & 9 & 3 & 1 \\ 1 & 3 & 9 & 3 \\ 3 & 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} c_1^{n+1} \\ c_2^{n+1} \\ c_3^{n+1} \\ c_4^{n+1} \end{bmatrix} \quad (B.8)$$

where the  $scv,c$  counter  $k \in 1, 2, 3, 4$ , follows a counterclockwise local ordering (per cell/element) of the four  $scv$  (and vertices) of the cell, see Figure B.2a).

The integration over the control-volume faces of a control volume can be split in the

sum over the sub-control volume faces of the sub-control volumes (see Figure B.3). Here, it should, however, be noted that the fluxes over those sub-control volume boundaries that are internal to the control volume, are cancelled out. These are the fluxes over sub-control volume faces that are on the the cell boundaries. Only the sub-control volume boundaries inside a cell remain. In this way, the values in the sub-control volume face centres can be computed using a local coefficient matrix, corresponding to the four cell-internal sub-control volume faces. The expression for the  $scvf,c$  then reads:

$$c_{scvf,c,j} = \frac{1}{8} \begin{bmatrix} 3 & 3 & 1 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} c_1^{n+1} \\ c_2^{n+1} \\ c_3^{n+1} \\ c_4^{n+1} \end{bmatrix} \quad (\text{B.9})$$

where the  $scvf,c$  counter  $j \in 1, 2, 3, 4$ , again follows a counterclockwise local ordering (per cell/element) of the four  $scvf$  of the cell, see Figure B.2b).

Using the coefficient matrices above, the interpolated values in the  $scv$  and  $scyf$  centres can be computed. These are then substituted in (B.7). The first term in (B.7) contains values at time level  $n + 1$ . Substituting the local interpolation matrices in this term, results in a mass matrix. The second term at time level  $n$  results in a right-hand-side contribution.

The third (flux) term in the equation, has not yet been discretized in time. It can be discretized (1) explicitly, resulting in a right-hand-side contribution, (2) implicitly, resulting in a contribution to the coefficient matrix, or (3) in a semi-implicit way (e.g. using a  $\theta$ -method), resulting in contributions to both the matrix and the right-hand side. For the explicit method, considering that the interpolation is central, we may expect that the scheme is not stable as it is similar to a simple *Forward in Time, Central in Space (FTCS)* method. This is investigated in the numerical experiments in the next section and will be analyzed in Section B.6.

## B.4 Numerical experiments

We consider two numerical experiments: a sinusoidal wave and a Gaussian disturbance.

### B.4.1 Sinusoidal wave

We consider a domain with length  $L = 100$  m, with an initial sine wave with wave length equal to the domain length and an amplitude  $A$  of 0.01 m. The initial condition reads:

$$c(x, 0) = A \sin(2\pi x/L) \quad (\text{B.10})$$

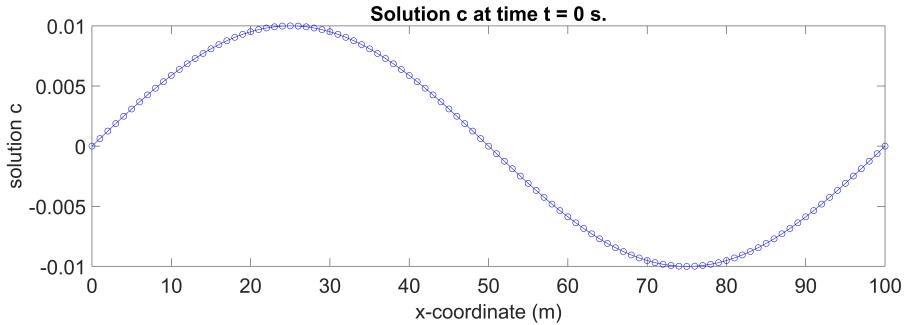


Figure B.4: Initial condition for the scalar advection test with sinusoidal initial condition.

The initial condition is shown in Figure B.4. We specify the advective velocity  $a = 10$  m/s and simulate for  $T = 5$  s. The sine wave should then travel exactly  $10 * 5 = 50$  m. The simple Cartesian grid has a uniform grid size  $\Delta x = 1$  m. The domain consists of 100 cells  $x$ -direction. We choose the time step  $\Delta t$  again based on the Courant-Friedrichs-Levy condition.

### B.4.2 Gaussian disturbance

To allow a Fourier-mode analysis, we consider the evolution of a small (Gaussian) disturbance. This disturbance is set as initial condition. The initial condition reads:

$$c(x, 0) = 1 + A \exp(-l_d(x_c - x)^2) \quad (\text{B.11})$$

where  $A$  is the amplitude of the disturbance,  $l_d$  is a measure for the length of the disturbance, and  $x_c$  is the centre of the Gaussian disturbance.

We consider a domain with length  $L = 100$  m, where the centre  $x_c$  of the Gaussian disturbance is at  $x_c = L/4 = 25$  m. We assume an amplitude  $A$  of 0.01 m and a

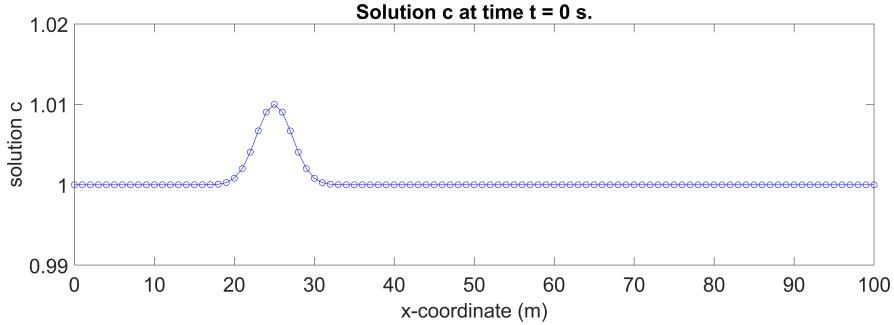


Figure B.5: Initial condition for the scalar advection test.

(dimensionless) length  $l_d$  of the disturbance of 0.1. The initial condition is shown in Figure B.5. We specify the advective velocity  $a = 10$  m/s and simulate for  $T = 5$  s. The disturbance should then move exactly  $10 * 5 = 50$  m. The simple Cartesian grid has a uniform grid size  $\Delta x = 1$  m. The domain thus consists of 100 cells  $x$ -direction. We choose the time step  $\Delta t$  based on the Courant-Friedrichs-Levy condition for stability:

$$v = \frac{a\Delta t}{\Delta x} \quad (\text{B.12})$$

where  $v$  is the Courant number.

## B.5 Results

### B.5.1 Sinusoidal wave

Results are shown in Figure B.6 for the sinusoidal wave with a Courant number  $v = 0.5$  and three values of  $\theta$ :  $\theta = 0$ ,  $\theta = 0.5$  and  $\theta = 0.7$ . The explicit simulation with  $\theta = 0$  becomes unstable (as expected). Therefore the solution is shown for time  $t = 2$  seconds after the start of the simulation, and not for the end of the simulation (after 20 seconds).

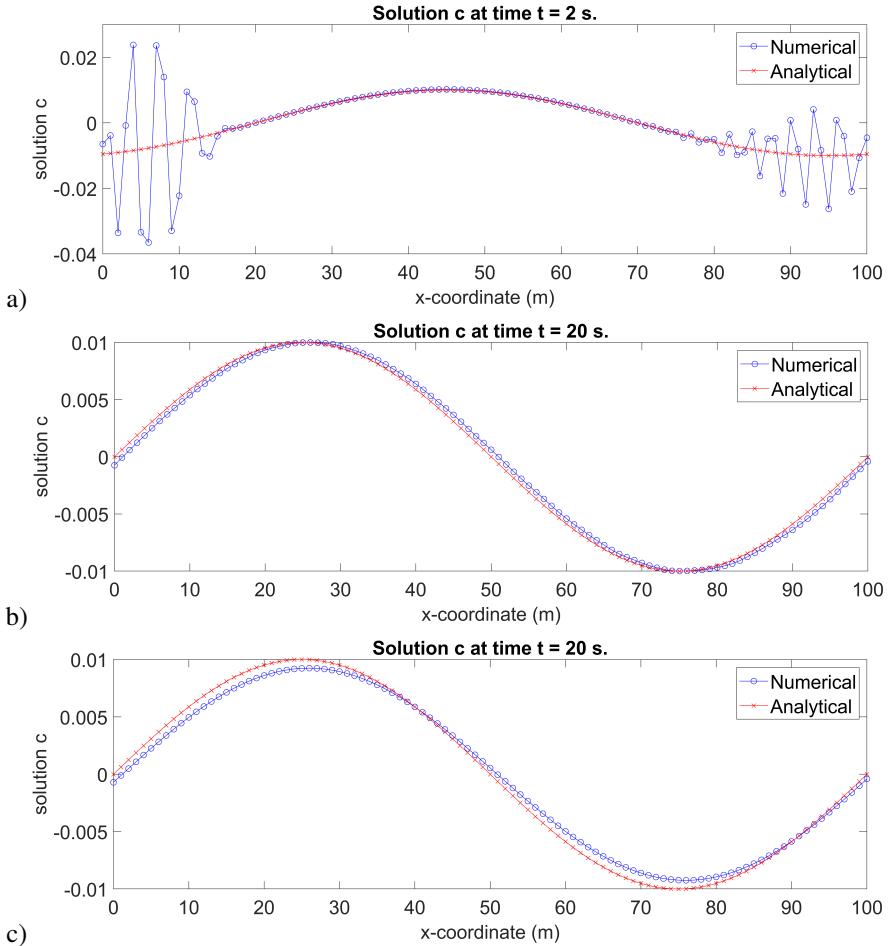


Figure B.6: Final solution (after  $T = 5$  s.) for scalar advection of a sine wave, for a Courant number of  $v = 1$  and three values of  $\theta$ : a)  $\theta = 0$  (solution shown after 2 seconds, simulation becomes unstable); b)  $\theta = 0.5$ ; and c)  $\theta = 0.7$ .

### B.5.2 Gaussian disturbance

Results are shown in Figure B.7 for the sinusoidal wave with a Courant number  $v = 0.5$  and three values of  $\theta$ : 0, 0.5 and 0.7. The explicit simulation with  $\theta = 0$  becomes

unstable (as expected). Therefore the solution is shown for time  $t = 2$  seconds after the start of the simulation, instead of for the end of the simulation (after 20 seconds),

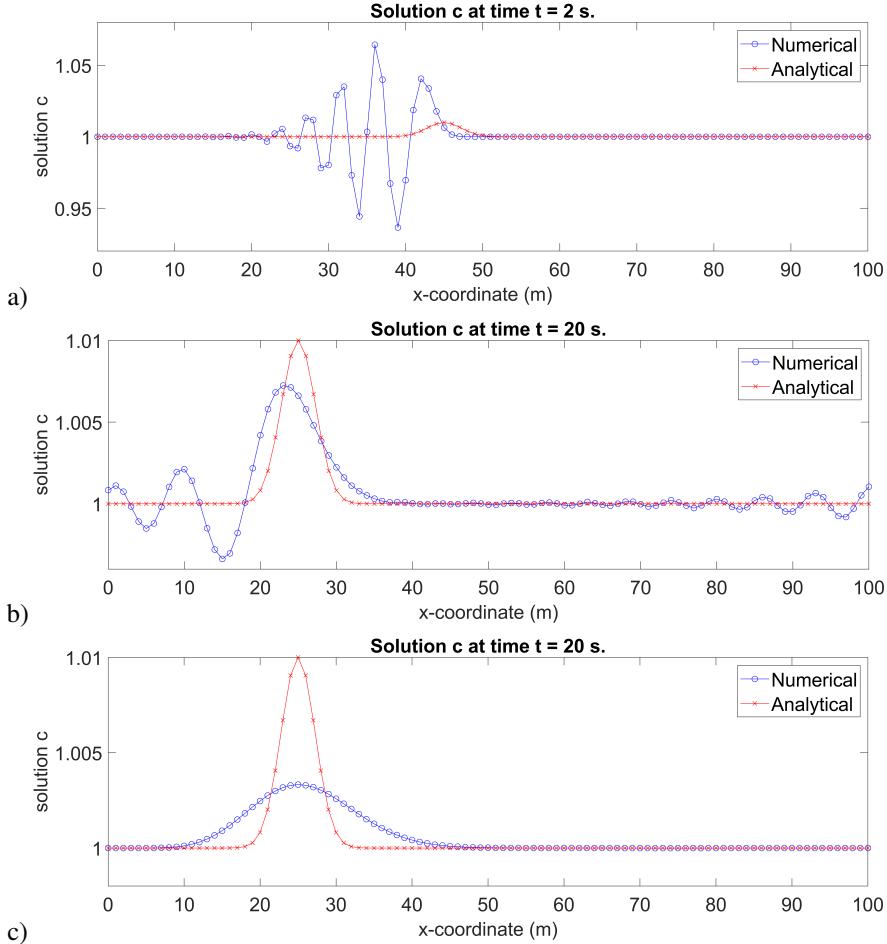


Figure B.7: Final solution (after  $T = 5$  s.) for scalar advection of a Gaussian disturbance, for a Courant number of  $\nu = 0.5$  and three values of  $\theta$ : a)  $\theta = 0$  (solution shown after 2 seconds, simulation becomes unstable); b)  $\theta = 0.5$ ; and c)  $\theta = 0.7$ .

## B.6 Analysis

In this section, we analyze the collocated vertex-centered, finite-volume scheme for its stability and accuracy, using a Von Neumann stability analysis based Fourier-modes. For simplicity, we consider the scheme in one dimension (1D).

Since we will be introducing Fourier modes of the form  $e^{-ikx}$ , we will use  $j$  as the spatial index, see Figure B.8.

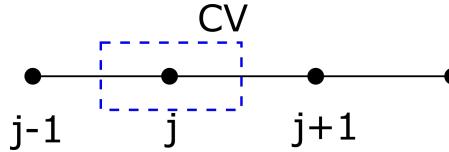


Figure B.8: One-dimensional grid with vertex-centred storage around vertex  $j$ . The dashed rectangle shows the (1D) control volume around vertex  $j$ .

In 1D, a control volume  $\Omega$  around vertex  $j$  consists of a left and right sub-control volume (scv), at positions  $j-1/4$  and  $j+1/4$ , respectively. The left, centre and right sub-control volume faces (scvf) are then located at positions  $j-1/2$ ,  $j$  and  $j+1/2$ . Additionally, we use the fact that the "area" of a sub-control volume  $A_{scv} = \Delta x/2$ . Using these definitions, we can write the integrated 1D equation as:

$$\frac{\Delta x}{2} \left( c_{j-1/4}^{n+1} - c_{j-1/4}^n + c_{j+1/4}^{n+1} - c_{j+1/4}^n \right) + a\Delta t \left( c_{j+1/2}^{n+\theta} - c_j^{n+\theta} + c_j^{n+\theta} - c_{j-1/2}^{n+\theta} \right) = 0 \quad (\text{B.13})$$

where we have written the flux in  $\theta$  notation, to be able to assess the properties of the explicit ( $\theta = 0$ ), implicit ( $\theta = 1$ ) and semi-implicit ( $0 < \theta < 1$ ) scheme.

In (B.13), it can be seen that the centre flux (at scvf or vertex  $j$  itself) drops out. We are left with:

$$\frac{\Delta x}{2} \left( c_{j-1/4}^{n+1} - c_{j-1/4}^n + c_{j+1/4}^{n+1} - c_{j+1/4}^n \right) + a\Delta t \left( c_{j+1/2}^{n+\theta} - c_{j-1/2}^{n+\theta} \right) = 0 \quad (\text{B.14})$$

Interpolating bilinearly (in 1D) the values at positions between the vertices, we obtain:

$$\begin{aligned} \frac{\Delta x}{2} \left( \frac{1}{4}c_{j-1}^{n+1} + \frac{3}{4}c_j^{n+1} - \frac{1}{4}c_{j-1}^n - \frac{3}{4}c_j^n + \frac{3}{4}c_j^{n+1} + \frac{1}{4}c_{j+1}^{n+1} - \frac{3}{4}c_j^n - \frac{1}{4}c_{j+1}^n \right) \\ + \frac{a\Delta t}{2} \left( c_{j+1}^{n+\theta} + c_j^{n+\theta} - c_j^{n+\theta} - c_{j-1}^{n+\theta} \right) = 0 \quad (\text{B.15}) \end{aligned}$$

or collecting terms with equal time level:

$$\begin{aligned} \Delta x \left( \frac{1}{8} c_{j-1}^{n+1} + \frac{3}{4} c_j^{n+1} + \frac{1}{8} c_{j+1}^{n+1} - \frac{1}{8} c_{j-1}^n - \frac{3}{4} c_j^n - \frac{1}{8} c_{j+1}^n \right) \\ + \frac{a\Delta t}{2} (c_{j+1}^{n+\theta} - c_{j-1}^{n+\theta}) = 0 \end{aligned} \quad (\text{B.16})$$

or introducing the Courant number  $v = a\Delta t / \Delta x$ :

$$\frac{1}{8} c_{j-1}^{n+1} + \frac{3}{4} c_j^{n+1} + \frac{1}{8} c_{j+1}^{n+1} - \frac{1}{8} c_{j-1}^n - \frac{3}{4} c_j^n - \frac{1}{8} c_{j+1}^n + \frac{v}{2} (c_{j+1}^{n+\theta} - c_{j-1}^{n+\theta}) = 0 \quad (\text{B.17})$$

To analyze the stability of this discretization, we substitute a solution of the form  $C = Ae^{-ikx} = Ae^{-ikj\Delta x}$ , where  $A = A(k)$  is the amplification factor,  $k$  is a wave number and  $i = \sqrt{-1}$  is the (unit) imaginary number. Upon substitution in (B.17), we obtain:

$$\begin{aligned} \frac{1}{8} A^{n+1} e^{-ik(j-1)\Delta x} + \frac{3}{4} A^{n+1} e^{-ikj\Delta x} + \frac{1}{8} A^{n+1} e^{-ik(j+1)\Delta x} \\ - \frac{1}{8} A^n e^{-ik(j-1)\Delta x} - \frac{3}{4} A^n e^{-ikj\Delta x} - \frac{1}{8} A^n e^{-ik(j+1)\Delta x} \\ + \frac{v}{2} (A^{n+\theta} e^{-ik(j+1)\Delta x} - A^{n+\theta} e^{-ik(j-1)\Delta x}) = 0 \end{aligned} \quad (\text{B.18})$$

Factoring out the solution at time  $n$ :  $C^{(n)} = A^n e^{-ikj\Delta x}$ , we get:

$$\begin{aligned} A^n e^{-ikj\Delta x} \left( \frac{1}{8} A e^{ik\Delta x} + \frac{3}{4} A + \frac{1}{8} A e^{-ik\Delta x} - \frac{1}{8} e^{ik\Delta x} - \frac{3}{4} - \frac{1}{8} e^{-ik\Delta x} \right. \\ \left. + \frac{v}{2} (A^\theta e^{-ik\Delta x} - A^\theta e^{ik\Delta x}) \right) = 0 \end{aligned} \quad (\text{B.19})$$

Assuming  $A \neq 0$ , the expression in (B.19) can only be zero, when the term in brackets is zero, i.e. when:

$$\begin{aligned} \frac{1}{8} A e^{ik\Delta x} + \frac{3}{4} A + \frac{1}{8} A e^{-ik\Delta x} - \frac{1}{8} e^{ik\Delta x} - \frac{3}{4} - \frac{1}{8} e^{-ik\Delta x} \\ + \frac{v}{2} (A^\theta e^{-ik\Delta x} - A^\theta e^{ik\Delta x}) = 0 \end{aligned} \quad (\text{B.20})$$

Now we can use the following trigonometric identities to replace the complex expo-

nents:

$$e^{ix} = \cos(x) + i\sin(x) \quad (\text{B.21})$$

$$e^{-ix} = \cos(x) - i\sin(x) \quad (\text{B.22})$$

$$\frac{e^{ix} + e^{-ix}}{2} = \cos(x) \quad (\text{B.23})$$

$$\frac{e^{ix} - e^{-ix}}{2i} = \sin(x) \quad (\text{B.24})$$

Using these identities, we obtain:

$$\frac{1}{4}A\cos(k\Delta x) + \frac{3}{4}A - \frac{1}{4}\cos(k\Delta x) - \frac{3}{4} - ivA^\theta \sin(k\Delta x) = 0 \quad (\text{B.25})$$

Noticing that  $A^\theta = \theta A + (1 - \theta)A^0 = \theta A + 1 - \theta$ , we get:

$$\frac{1}{4}A\cos(k\Delta x) + \frac{3}{4}A - \frac{1}{4}\cos(k\Delta x) - \frac{3}{4} - iv(\theta A + 1 - \theta) \sin(k\Delta x) = 0 \quad (\text{B.26})$$

Rearranging results in the following expression for the amplification factor  $A$ :

$$A = \frac{\frac{1}{4}\cos(k\Delta x) + \frac{3}{4} + iv(1 - \theta)\sin(k\Delta x)}{\frac{1}{4}\cos(k\Delta x) + \frac{3}{4} - iv\theta\sin(k\Delta x)} \quad (\text{B.27})$$

which we can simplify using the complex conjugate of the denominator as follows:

$$A = \frac{\frac{1}{4}\cos(k\Delta x) + \frac{3}{4} + iv(1 - \theta)\sin(k\Delta x)}{\frac{1}{4}\cos(k\Delta x) + \frac{3}{4} - iv\theta\sin(k\Delta x)} \cdot \frac{\frac{1}{4}\cos(k\Delta x) + \frac{3}{4} + iv\theta\sin(k\Delta x)}{\frac{1}{4}\cos(k\Delta x) + \frac{3}{4} + iv\theta\sin(k\Delta x)} \quad (\text{B.28})$$

or

$$A = \frac{\frac{1}{16}\cos^2(k\Delta x) + \frac{3}{8}\cos(k\Delta x) + \frac{9}{16} + iv\left(\frac{1}{4}\cos(k\Delta x) + \frac{3}{4}\right)\sin(k\Delta x) - v^2\theta(1 - \theta)\sin^2(k\Delta x)}{\frac{1}{16}\cos^2(k\Delta x) + \frac{3}{8}\cos(k\Delta x) + \frac{9}{16} + v^2\theta^2\sin^2(k\Delta x)} \quad (\text{B.29})$$

or

$$A = \frac{\frac{1}{4}\cos^2(k\Delta x) + \frac{3}{2}\cos(k\Delta x) + \frac{9}{4} + iv(\cos(k\Delta x) + 3)\sin(k\Delta x) - 4v^2\theta(1 - \theta)\sin^2(k\Delta x)}{\frac{1}{4}\cos^2(k\Delta x) + \frac{3}{2}\cos(k\Delta x) + \frac{9}{4} + 4v^2\theta^2\sin^2(k\Delta x)} \quad (\text{B.30})$$

We can now find the amplification factor by computing  $\|A\|^2 = AA^*$ , where  $A^*$  is the complex conjugate of  $A$ . The amplification factor  $\|A\|$  is depicted in Figure B.9 for nine combinations of  $\theta$  and  $v$  (for  $\theta = 0.5$ ,  $\|A\| = 1$ , irrespective of the Courant number  $v$ , so only one line is shown).

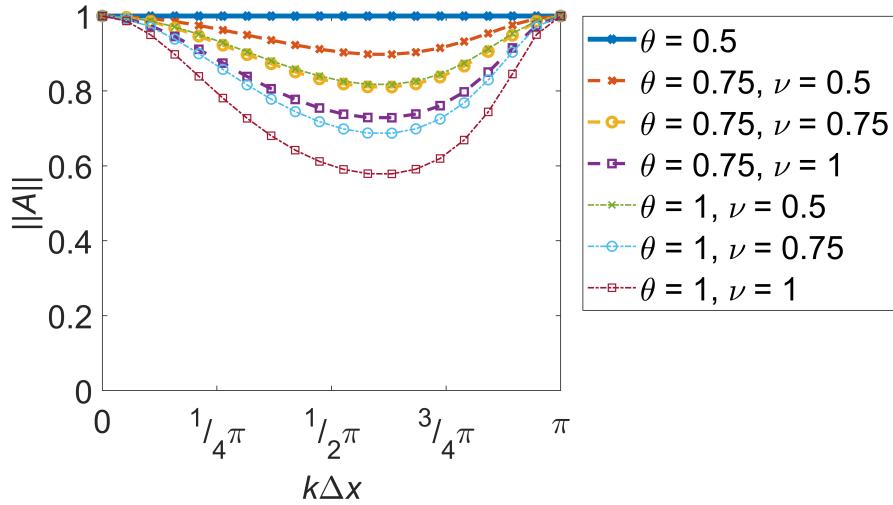


Figure B.9: Amplification factor  $\|A\|$  versus wave number  $k\Delta x$  for the 1D vertex-centred finite volume scheme. Depicted are nine different combinations of  $\theta$  and  $\nu$  (for  $\theta = 0.5$ ,  $\|A\| = 1$ , irrespective of the Courant number  $\nu$ , so only one line is shown).

## B.7 Discussion

## B.8 Conclusions



## APPENDIX C

---

### BOUNDARY CONDITIONS

#### C.1 Introduction

To allow a simple illustration of the handling of boundary conditions for the collocated, vertex-centered, finite-volume scheme, we consider a one-dimensional test model for scalar advection. However, we do consider the definitions and implementation aspects as they would be required for a two-dimensional implementation.

#### C.2 Equations

As in Appendix B, we consider the advection of a scalar quantity  $c$  in one dimension (the  $x$ -direction) with a constant flow/transport velocity  $a$ . The governing equation reads:

$$\frac{\partial c}{\partial t} + a \frac{\partial c}{\partial x} = 0 \quad (\text{C.1})$$

with  $a > 0$ , indicating flow in positive  $x$ -direction.

## C.3 Boundary handling

For the internal control volumes, this problem is discretized using the method described in Appendix B. This means that the first step in the solution algorithm is to loop over all mesh cells/elements and, for each element, handle the internal sub-control volumes and volume faces, by computing the matrix and vector contributions for the nodes corresponding to the sub-control volume and its faces. For details, see Appendix B.

The next step is then the handling of the domain boundaries and the inclusion of possible prescribed boundary conditions. This is done in two steps:

1. Discretizing the model equation(s) at the boundary volumes and corner volumes
2. Applying the user-specified boundary conditions

The first of these two steps mentions the notion of *boundary volumes* and *corner volumes*. The former are line segments of control volumes that contain internal grid nodes that are adjacent to the domain boundary, see Figure D.1. The latter are single points, that correspond to corners of a control volume, adjacent to the grid/domain corners, see Figure D.1. These two types of boundary entities are handled separately in the the first step.

### C.3.1 Discretizing the model equation(s) at the boundary

In this step, the incomplete, one-sided discretization for the virtual nodes, that results from the loop over all internal sub-control volumes, is replaced by an appropriate (complete) discretization of the model equations at so-called boundary volumes and corner volumes, see Figure D.1.

#### C.3.1.1 Discretization at boundary volumes

It can be seen that a boundary volume is positioned between an internal node  $n_{in}$  and a virtual node  $n_v$ . The discretization of the equation at this boundary volume is now realized using a similar recipe as for the internal (sub-control volume) faces.

For a boundary volume  $\Omega_{BV}$ , the integrated equation reads:

$$\int_{\Omega_{BV}} \frac{\partial c}{\partial t} dV + a \int_{\Omega_{BV}} \frac{\partial c}{\partial x} dV = 0 \quad (\text{C.2})$$

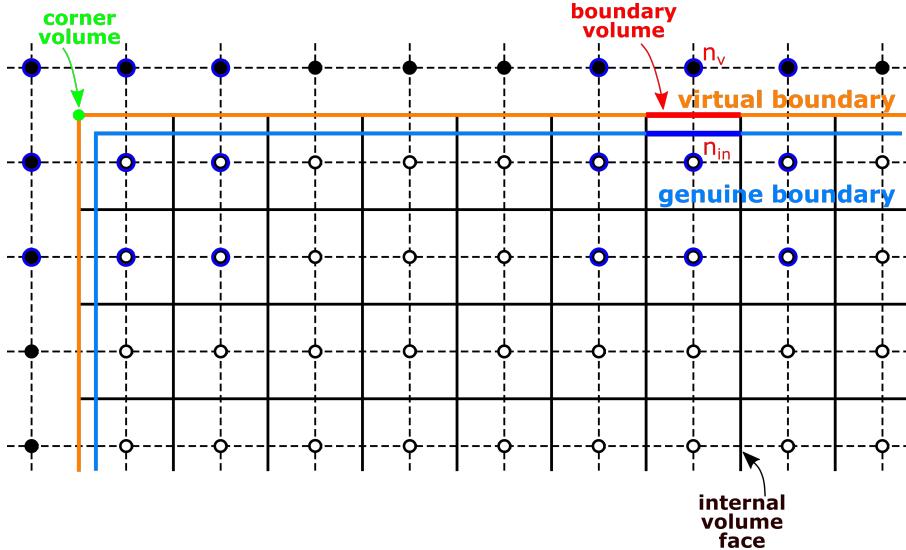


Figure C.1: Definitions related to the handling of domain boundaries, for a rectangular domain.

Applying a semi-discretization in time, we obtain:

$$\int_{\Omega_{BV}} \frac{c^{n+1} - c^n}{\Delta t} dV + a \int_{\Omega_{BV}} \frac{\partial c}{\partial x} dV = 0 \quad (\text{C.3})$$

where  $n$  indicates the time level. Now we divide the control volume  $\Omega_{BV}$  in two sub-control volume faces (scvf):

$$\sum_{scvf \in \Omega_{BV}} \int_{scvf} \frac{c^{n+1} - c^n}{\Delta t} dV + a \sum_{scvf \in \Omega_{BV}} \int_{scvf} \frac{\partial c}{\partial x} dV = 0 \quad (\text{C.4})$$

Similar to the handling of the internal control volumes, the formulation for the boundary volume also contains two contributions from two different elements/cells. These can be again handled on a *per-cell* basis. So we reduce ourselves here to the integral over a single *scvf* (inside a single cell). These integrals are again (for simplicity) approximated using a one-point Gauss quadrature rule. However, here we encounter an important difference from the discretization for the internal grid cells. Since the boundary volume  $\Omega_{BV}$  is a line element, the volume integral of the flux term, containing the gradient  $\partial c / \partial x$  (or in 2D  $\partial c / \partial y$ ), can only be rewritten to a line integral

using Gauss' divergence theorem, for gradients along (tangential to) such a boundary volume, not for gradients normal to the boundary. Therefore, the flux term needs to be integrated differently, depending on the relative orientation of the boundary volume, with respect to the  $x$ -direction. This is handled using either the inner-product of the unit outward normal/tangential vector of the boundary volume and the unit vector in the  $x$ -direction, i.e. either  $(\vec{n}_{scvf}, \vec{n}_x)$  or  $(\vec{t}_{scvf}, \vec{n}_x)$ , as follows:

$$\begin{aligned} & \frac{c_{scvf,C}^{n+1} - c_{scvf,C}^n}{\Delta t} l_{scvf} \\ & + a \frac{\partial c}{\partial x}_{scvf,C} (\vec{n}_{scvf}, \vec{n}_x) l_{scvf} \\ & + a (c_{scvf,2} - c_{scvf,1}) (\vec{t}_{scvf}, \vec{n}_x) = 0 \end{aligned} \quad (\text{C.5})$$

where the subscript  $scvf,C$  refers to the  $scvf$  centre. The  $c_{scvf,1}$  and  $c_{scvf,2}$  are the values of  $c$  at the two end points of the  $scvf$  (according to some directional definition). Additionally,  $l_{scvf}$  is the  $scvf$  length. For the Cartesian grid, this is  $\Delta x/2$  or  $\Delta y/2$ , depending on the face orientation.

The first term in the equation – the time derivative – requires the values of the quantity  $c$  at the  $scvf,C$ , which are obtained using bilinear interpolation per cell/element, just as for the internal control volumes. This involves the already derived local coefficient matrix, see (B.9). This matrix contains the coefficients that determine the weighing factors of the bilinear interpolation for the four nodes of a cell (the four columns of the matrix), for the four internal  $scvf$  of a cell (the four rows of the matrix). However, the boundary volume only consists of two  $scvf$ . The local matrix therefore reduces to a 2x4 matrix (two rows, four columns, corresponding to either  $scvf$  1 and 3, or  $scvf$  2 and 4, of the original internal  $scvf$  numbering), depending on the orientation of the boundary volume. The local matrix for the bilinear interpolation  $\bar{\tilde{A}}_{BV}$  at a boundary volume  $\tilde{A}_{BV}$  on the west/east boundary of the domain then reads:

$$\bar{\tilde{A}}_{BV} = \frac{1}{8} \begin{pmatrix} 3 & 3 & 1 & 1 \\ 1 & 1 & 3 & 3 \end{pmatrix} \quad (\text{C.6})$$

The second term in (C.5) involves an integration of the derivative perpendicular to the boundary volume (i.e. for boundary volumes perpendicular to the gradient direction). The derivative  $(\partial c / \partial x)_{scvf,C}$  is computed from the values in the surrounding vertices. For instance, at  $scvf_{R,1}$  as in Figure D.2, the gradient is computed as:

$$\frac{\partial c}{\partial x}_{scvf_{R,1}} = \frac{3}{4} \frac{c_{R_2} - c_{R_1}}{\Delta x} + \frac{1}{4} \frac{c_{R_3} - c_{R_4}}{\Delta x} \quad (\text{C.7})$$

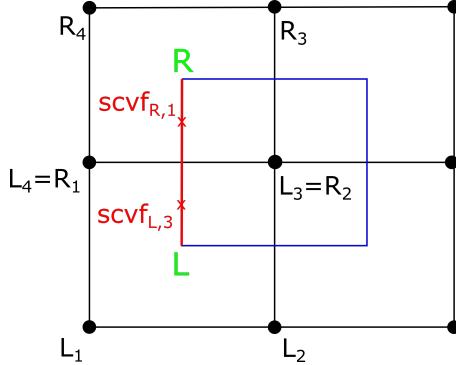


Figure C.2: Local stencil for a boundary volume.

A similar expression can be formed for the gradient at  $scvf_{L,3}$ , as a function of the values of  $c$  in nodes  $L_1$ ,  $L_2$ ,  $L_3$  and  $L_4$ .

These expressions can also be written in local matrix form, for the evaluation of the gradient in  $x$ -direction at a boundary volume. The local gradient matrix  $\bar{A}_{BV}^{\nabla}$  on the west/east boundary of the domain then reads:

$$\bar{A}_{BV}^{\nabla} = \frac{1}{4} \begin{pmatrix} -3 & 3 & 1 & -1 \\ -1 & 1 & 3 & -3 \end{pmatrix} \quad (\text{C.8})$$

With this matrix definition, the second (normal-derivative) flux term becomes:

$$a \frac{\partial c}{\partial x}_{scvf,C} (\vec{n}_{scvf}, \vec{n}_x) l_{scvf} = a \bar{A}_{BV}^{\nabla} \vec{c}_{E_{scvf}} (\vec{n}_{scvf}, \vec{n}_x) l_{scvf} \quad (\text{C.9})$$

where  $\vec{c}_{E_{scvf}}$  is the vector of nodal values of  $c$  of the element  $E$  that contains the  $scvf$ .

The last term in (C.5) involves an integration of the derivative along the boundary volume (i.e. for boundary volumes parallel to the gradient direction). It contains the values at the  $scvf$  endpoints, i.e. at a cell centre and at a midpoint/centre of an edge of the primary grid. The interpolations in these points are straightforward (see

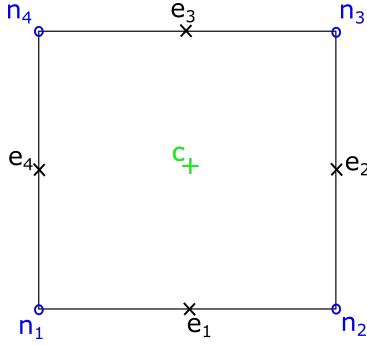


Figure C.3: Definitions of the node, edge and cell centre positions and numbering (for a quadrilateral cell).

Figure D.3):

$$c_{\text{centre}} = \frac{1}{4}(c_1 + c_2 + c_3 + c_4) \quad (\text{C.10})$$

$$\begin{pmatrix} c_{e_1} \\ c_{e_2} \\ c_{e_3} \\ c_{e_4} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} c_1 + c_2 \\ c_2 + c_3 \\ c_3 + c_4 \\ c_4 + c_1 \end{pmatrix} \quad (\text{C.11})$$

where the  $c_{e_1}, c_{e_2}, \dots$  correspond to the values of  $c$  in the centres of the edges of the primary grid.

These expressions are all substituted in (C.5) to complete the spatial discretization part for the boundary volume.

The first term in (C.5) contains values at time level  $n + 1$ . Substituting the local interpolation matrices in this term, results in a mass matrix. The second term at time level  $n$  results in a right-hand-side contribution.

The (second and third) flux terms in the equation, have not yet been discretized in time. They can be discretized (1) explicitly, resulting in a right-hand-side contribution, (2) implicitly, resulting in a contribution to the coefficient matrix, or (3) in a semi-implicit way (e.g. using a  $\theta$ -method), resulting in contributions to both the matrix and the right-hand side.

For instance, applying a  $\theta$ -method for time integration, the discretization for a single *scvf* (here *scvf,RI* from Figure D.2) contributing to the integral over the boundary

volume, becomes:

$$\begin{aligned}
 & \frac{1}{8\Delta t} (3(c_{R1})^{n+1} + 3(c_{R2})^{n+1} + (c_{R3})^{n+1} + (c_{R4})^{n+1} \\
 & \quad - 3(c_{R1})^n - 3(c_{R2})^n - (c_{R3})^n - (c_{R4})^n) l_{scvf,R1} \\
 & + \frac{a\theta}{4} (3(c_{R2})^{n+1} - 3(c_{R1})^{n+1} + (c_{R3})^{n+1} - (c_{R4})^{n+1}) l_{scvf,R1} \\
 & + \frac{a(1-\theta)}{4} (3(c_{R2})^n - 3(c_{R1})^n + (c_{R3})^n - (c_{R4})^n) l_{scvf,R1}
 \end{aligned} \tag{C.12}$$

where the numbering of the four nodal values of  $c$  of cell  $R$  is defined in Figure D.2. The inner product  $(\vec{n}_{scvf,R1}, \vec{n}_x)$  has dropped out as it is 1 for this  $scvf$  and the third flux term (tangential to the boundary) has dropped out because  $(\vec{l}_{scvf,R1}, \vec{n}_x)$  is equal to 0, since we only consider transport/gradients in  $x$ -direction, in this simple 1D transport problem).

This equation provides the matrix and vector coefficients for the degree of freedom corresponding to the virtual node that is directly adjacent to the boundary volume (i.e. closest to the  $scvf$  centre), in this case for  $R_1$ , which is the same node as  $L_4$ . Assuming for simplicity that node  $R_1$  corresponds with degree of freedom (and matrix row) 1, node  $R_2$  corresponds with degree of freedom (and matrix row) 2, node  $R_3$  corresponds with degree of freedom (and matrix row) 3 and node  $R_4$  corresponds with degree of freedom (and matrix row) 4, the local matrix and vector entries coming from  $scvf_{R1}$  would be:

$$\left( \begin{array}{cccc} \frac{3}{8} - \frac{3a\theta\Delta t}{4\Delta x} & \frac{3}{8} + \frac{3a\theta\Delta t}{4\Delta x} & \frac{1}{8} + \frac{a\theta\Delta t}{4\Delta x} & \frac{1}{8} - \frac{a\theta\Delta t}{4\Delta x} \\ \dots & \dots & \dots & \dots \end{array} \right) \left( \begin{array}{c} c_{R1}^{n+1} \\ c_{R2}^{n+1} \\ c_{R3}^{n+1} \\ c_{R4}^{n+1} \\ \dots \end{array} \right) = \\
 \left( \begin{array}{c} \frac{3}{8} (c_{R1}^n + c_{R2}^n) + \frac{1}{8} (c_{R3}^n + c_{R4}^n) - \frac{3a}{4} (1-\theta) \Delta t \frac{c_{R2}^n - c_{R1}^n}{\Delta x} - \frac{a}{4} (1-\theta) \Delta t \frac{c_{R3}^n - c_{R4}^n}{\Delta x} \\ \dots \end{array} \right)$$

Written out for the first row of the system of equations (corresponding to the virtual node):

$$\begin{aligned}
 & \left( \frac{3}{8} - \frac{3a\theta\Delta t}{4\Delta x} \right) c_{R1}^{n+1} + \left( \frac{3}{8} + \frac{3a\theta\Delta t}{4\Delta x} \right) c_{R2}^{n+1} + \\
 & \left( \frac{1}{8} - \frac{a\theta\Delta t}{4\Delta x} \right) c_{R4}^{n+1} + \left( \frac{1}{8} + \frac{a\theta\Delta t}{4\Delta x} \right) c_{R3}^{n+1} = \\
 & \frac{3}{8} (c_{R1}^n + c_{R2}^n) + \frac{1}{8} (c_{R3}^n + c_{R4}^n) \\
 & - \frac{3a}{4} (1-\theta)\Delta t \frac{c_{R2}^n - c_{R1}^n}{\Delta x} \\
 & - \frac{a}{4} (1-\theta)\Delta t \frac{c_{R3}^n - c_{R4}^n}{\Delta x}
 \end{aligned} \tag{C.13}$$

### C.3.1.2 Discretization at corner volumes

In the previous section, we have seen how the model equations are discretized at boundary volumes, i.e. at line segments between virtual and internal nodes. The equations that follow from these discretizations correspond to the degrees of freedom of the virtual nodes adjacent to those boundary volumes. The virtual nodes that are in the corner of the domain do not have a directly connected internal node. However, they do have a corresponding degree of freedom and, therefore, also require a representative equation. For this purpose, the notion of a *corner volume* is introduced. It is a "point volume" that is used for discretizing the model equations, corresponding to the degree of freedom of the virtual corner node. The discretization is built up very similar to that of the boundary volume, with the exception that there is now no part of the flux integral along the boundary (since a point has no length). The flux term, containing the gradient, is thus simply evaluated at the corner volume  $CV$ :

$$\frac{c_{CV}^{n+1} - c_{CV}^n}{\Delta t} + a \left. \frac{\partial c}{\partial x} \right|_{CV} = 0 \tag{C.14}$$

The position of a corner volume  $CV$  is at the centre of a cell (of the primary mesh). The interpolation is thus a straightforward average. Again applying a  $\theta$ -method, the resulting discretization looks as follows (where we use the subscript  $CV$  for corner

volume):

$$\begin{pmatrix} \frac{1}{4} - \frac{a\Delta t \theta}{4\Delta x} & \frac{1}{4} + \frac{a\Delta t \theta}{4\Delta x} & \frac{1}{4} + \frac{a\Delta t \theta}{4\Delta x} & \frac{1}{4} - \frac{a\Delta t \theta}{4\Delta x} \\ \dots & \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} c_{R_1}^{n+1} \\ c_{R_2}^{n+1} \\ c_{R_3}^{n+1} \\ c_{R_4}^{n+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{4} (c_{R_1}^n + c_{R_2}^n + c_{R_3}^n + c_{R_4}^n) - \frac{a(1-\theta)\Delta t}{4\Delta x} (c_{R_2}^n - c_{R_1}^n + c_{R_3}^n - c_{R_4}^n) \\ \dots \end{pmatrix} \quad (\text{C.15})$$

or written out for the first row of the system of equations (corresponding to the virtual corner node):

$$\begin{aligned} & \left( \frac{1}{4} - \frac{a\theta\Delta t}{4\Delta x} \right) c_{R_1}^{n+1} + \left( \frac{1}{4} + \frac{a\theta\Delta t}{4\Delta x} \right) c_{R_2}^{n+1} + \\ & \left( \frac{1}{4} + \frac{a\theta\Delta t}{4\Delta x} \right) c_{R_3}^{n+1} + \left( \frac{1}{4} - \frac{a\theta\Delta t}{4\Delta x} \right) c_{R_4}^{n+1} = \\ & \frac{1}{4} (c_{R_1}^n + c_{R_2}^n + c_{R_3}^n + c_{R_4}^n) - \frac{a(1-\theta)\Delta t}{4\Delta x} (c_{R_2}^n - c_{R_1}^n + c_{R_3}^n - c_{R_4}^n) \end{aligned} \quad (\text{C.16})$$

### C.3.2 Applying the user-specified boundary conditions

The next step is to introduce possible user-prescribed boundary conditions. For now, we limit ourselves to Dirichlet- or Neumann-type boundary conditions. The discretizations set up for nodes that have such a boundary specification overwrite the discretizations constructed in section D.3.1.

#### C.3.2.1 Dirichlet-type conditions

For a Dirichlet-type condition, the value of  $c$  at the boundary is prescribed. This can be written as:

$$\alpha c_v^{n+1} + (1 - \alpha) c_{in}^{n+1} = c_{given} \quad (\text{C.17})$$

where  $c_{given}$  is the user-prescribed value for the concentration  $c$ . The coefficient  $\alpha$  accounts for the position of the actual boundary, with respect to the virtual and inner nodes. For  $\alpha$  is 1, the genuine boundary coincides with the virtual node position and for  $\alpha = 0$ , the genuine boundary is located at the inner node, see Figure D.1. A value

of  $\alpha$  between 0 and 1 can handle a positioning of the genuine boundary somewhere between the inner and virtual nodes.

This equation overwrites the equation constructed from the discretization of the model equations in section D.3.1. For systems of equations, it might be convenient to combine the discretized model equations with the Dirichlet condition. This will be discussed in a separate chapter.

### C.3.2.2 Neumann-type conditions

For a Neumann-type condition, the gradient of  $c$  at the boundary is prescribed. This can be written as:

$$\vec{n}_{BV} \frac{c_{in}^{n+1} - c_v^{n+1}}{\Delta s} = \nabla c_{given} \quad (\text{C.18})$$

where  $\vec{n}_{BV}$  accounts for the orientation of the boundary volume,  $\Delta s$  is the local grid size (for 1D transport, this can only be  $\Delta x$ ) and where  $\nabla c_{given}$  is the user-prescribed value for the gradient of the concentration  $\partial c / \partial x$ .

This equation overwrites the equation constructed from the discretization of the model equations in section D.3.1. For systems of equations, it might be convenient to combine the discretized model equations with the Neumann condition. This will be discussed in a separate chapter.

## C.4 Numerical experiments

... To do ...

## C.5 Results

... To do ...

## C.6 Analysis

... To do ...

## C.7 Discussion

... To do ...

## C.8 Conclusions

... To do ...



## APPENDIX D

### BOUNDARY CONDITIONS FOR 2D SHALLOW WATER

#### D.1 Introduction

In this section we illustrate in more detail the handling of boundary conditions for the collocated, vertex-centered, finite-volume scheme, for the two-dimensional shallow-water model.

#### D.2 Equations

We assume the pressure-gradient term is written as  $\vec{P} = (0, gh\partial\zeta/\partial x, gh\partial\zeta/\partial y)^T$ , which is a source term. Focusing here only at the flux terms, the governing equations

read (excluding viscous fluxes):

$$\frac{\partial h}{\partial t} + \nabla \cdot (\vec{q}) = 0 \quad (\text{D.1})$$

$$\frac{\partial q}{\partial t} + \nabla \cdot (\vec{q}q/h) = 0 \quad (\text{D.2})$$

$$\frac{\partial r}{\partial t} + \nabla \cdot (\vec{q}r/h) = 0 \quad (\text{D.3})$$

with  $h$  the total water depth,  $\vec{q} = (q, r)^T$ , the flux vector,  $q$  the flux/discharge (per unit width) in  $x$ -direction and  $r$  the flux/discharge (per unit width) in  $y$ -direction.

This equation can also be written in vector form:

$$\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \vec{F} = 0 \quad (\text{D.4})$$

with  $\vec{U} = (h, q, r)^T$  the state vector and with flux tensor  $\vec{F}$  given by:

$$\vec{F} = \begin{pmatrix} q & r \\ q^2/h & qr/h \\ qr/h & r^2/h \end{pmatrix} \quad (\text{D.5})$$

or split in  $x$ - and  $y$ -flux vectors:

$$\vec{F}^x = \begin{pmatrix} q \\ q^2/h \\ qr/h \end{pmatrix} \quad (\text{D.6})$$

$$\vec{F}^y = \begin{pmatrix} r \\ qr/h \\ r^2/h \end{pmatrix} \quad (\text{D.7})$$

These equations are integrated over finite control volumes. For a control volume  $\Omega$ , the equation system becomes:

$$\int_{\Omega} \frac{\partial h}{\partial t} dV + \int_{\Omega} \nabla \cdot (\vec{q}) dV = 0 \quad (\text{D.8})$$

$$\int_{\Omega} \frac{\partial q}{\partial t} dV + \int_{\Omega} \nabla \cdot (\vec{q}q/h) dV = 0 \quad (\text{D.9})$$

$$\int_{\Omega} \frac{\partial r}{\partial t} dV + \int_{\Omega} \nabla \cdot (\vec{q}r/h) dV = 0 \quad (\text{D.10})$$

or

$$\int_{\Omega} \frac{\partial \vec{U}}{\partial t} dV + \int_{\Omega} \nabla \cdot \vec{F} dV = 0 \quad (\text{D.11})$$

For all internal control volumes, the volume integral of the flux is rewritten to a surface integral using Gauss' divergence theorem.

## D.3 Boundary handling

For the internal control volumes, this problem is discretized using a 2D extension of the method described in Appendix B. This means that the first step in the solution algorithm is to loop over all mesh cells/elements and, for each element, handle the internal sub-control volumes and volume faces, by computing the matrix and vector contributions for the nodes corresponding to the sub-control volume and its faces. For details, see Appendix B.

The next step is then the handling of the domain boundaries and the inclusion of possible prescribed boundary conditions. This is done in two steps:

1. Discretizing the model equation(s) at the boundary volumes and corner volumes
2. Applying the user-specified boundary conditions

The first of these two steps mentions the notion of *boundary volumes* and *corner volumes*. The former are line segments of control volumes that contain internal grid nodes that are adjacent to the domain boundary, see Figure D.1. The latter are single points, that correspond to corners of a control volume, adjacent to the grid/domain corners, see Figure D.1. These two types of boundary entities are handled separately in the the first step.

### D.3.1 Discretizing the model equation(s) at the boundary

In this step, the incomplete, one-sided discretization for the virtual nodes, that results from the loop over all internal sub-control volumes, is replaced by an appropriate (complete) discretization of the model equations at so-called boundary volumes and corner volumes, see Figure D.1.

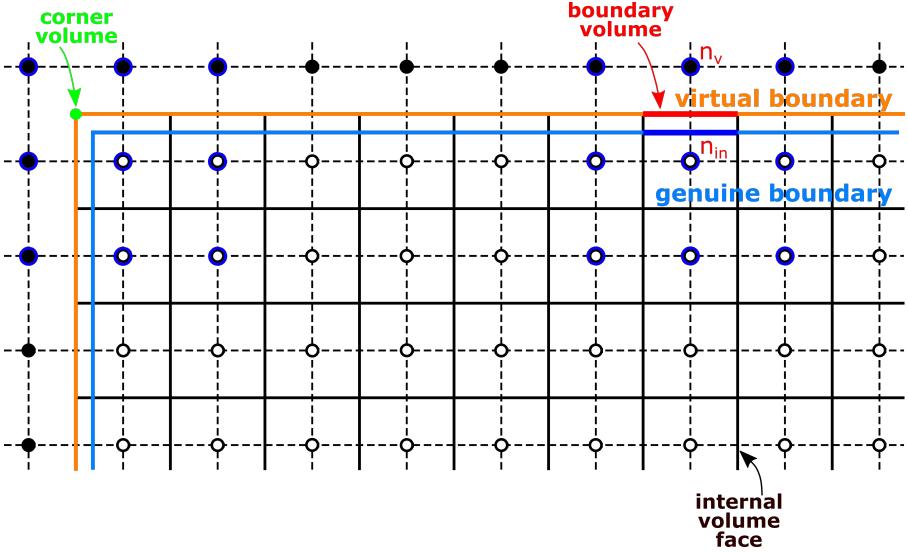


Figure D.1: Definitions related to the handling of domain boundaries, for a rectangular domain.

### D.3.1.1 Discretization at boundary volumes

It can be seen that a boundary volume is positioned between an internal node  $n_{in}$  and a virtual node  $n_v$ . The discretization of the equation at this boundary volume is now realized using a similar recipe as for the internal (sub-control volume) faces.

For a boundary volume  $\Omega_{BV}$ , the integrated equation reads:

$$\int_{\Omega_{BV}} \frac{\partial \vec{U}}{\partial t} dV + \int_{\Omega_{BV}} \nabla \cdot \vec{F} dV = 0 \quad (\text{D.12})$$

Applying a semi-discretization in time, we obtain:

$$\int_{\Omega_{BV}} \frac{\vec{U}^{n+1} - \vec{U}^n}{\Delta t} dV + \int_{\Omega_{BV}} \nabla \cdot \vec{F} dV = 0 \quad (\text{D.13})$$

where  $n$  indicates the time level. Now we divide the control volume  $\Omega_{BV}$  in two sub-control volume faces (scvf):

$$\sum_{scvf \in \Omega_{BV}} \int_{scvf} \frac{\vec{U}^{n+1} - \vec{U}^n}{\Delta t} dS + \sum_{scvf \in \Omega_{BV}} \int_{scvf} \nabla \cdot \vec{F} dS = 0 \quad (\text{D.14})$$

Similar to the handling of the internal control volumes, the formulation for the boundary volume also contains two contributions from two different elements/cells. These can be again handled on a *per-cell* basis. So we reduce ourselves here to the integral over a single *scvf* (inside a single cell). These integrals are again (for simplicity) approximated using a one-point Gauss quadrature rule. However, here we encounter an important difference from the discretization for the internal grid cells. Since the boundary volume  $\Omega_{BV}$  is a line element, the volume integral of the divergence of the flux term, can only be rewritten to a line integral using Gauss' divergence theorem, for gradients along (tangential to) such a boundary volume, not for gradients normal to the boundary. Therefore, the flux term needs to be integrated in two separate steps, one part for the flux normal to the boundary and one part for the flux tangential to the boundary. This is handled by taking the inner-product of the flux with the unit outward normal/tangential vector of the boundary volume, i.e. either with  $\vec{n}_{scvf}$  or with  $\vec{l}_{scvf}$ , as follows:

$$\begin{aligned} & \frac{\vec{U}_{scvf,C}^{n+1} - \vec{U}_{scvf,C}^n}{\Delta t} l_{scvf} \\ & + (\nabla \cdot \vec{F})_{scvf,C} l_{scvf} = 0 \end{aligned} \quad (\text{D.15})$$

where the subscript *scvf,C* refers to the *scvf* centre. Additionally,  $l_{scvf}$  is the *scvf* length. For the Cartesian grid, this is  $\Delta x/2$  or  $\Delta y/2$ , depending on the face orientation.

The first term in the equation – the time derivative – requires the values of the state vector  $\vec{U}$  at the *scvf,C*, which are obtained using bilinear interpolation per cell/element, just as for the internal control volumes. This involves the already derived local coefficient matrix, see (B.9). This matrix contains the coefficients that determine the weighing factors of the bilinear interpolation for the four nodes of a cell (the four columns of the matrix), for the four internal *scvf* of a cell (the four rows of the matrix). However, the boundary volume only consists of two *scvf*. Therefore, only two rows of the local coefficient matrix are used for a boundary volume (two rows, four columns, corresponding to either *scvf* 1 and 3, or *scvf* 2 and 4, of the original internal *scvf* numbering), depending on the orientation of the boundary volume.

The second term in (D.15) involves the evaluation of the gradients of the flux at the quadrature point (the *scvf* centre). The *x*- and *y*-gradients are computed from the values in the surrounding vertices as: For instance, at *scvf*<sub>R,1</sub> as in Figure D.2, the *x*-gradient of the state vector  $\vec{U}$  is computed as:

$$\frac{\partial \vec{U}}{\partial x}_{scvf_{R,1}} = \frac{3}{4} \frac{\vec{U}_{R_2} - \vec{U}_{R_1}}{\Delta x} + \frac{1}{4} \frac{\vec{U}_{R_3} - \vec{U}_{R_4}}{\Delta x} \quad (\text{D.16})$$

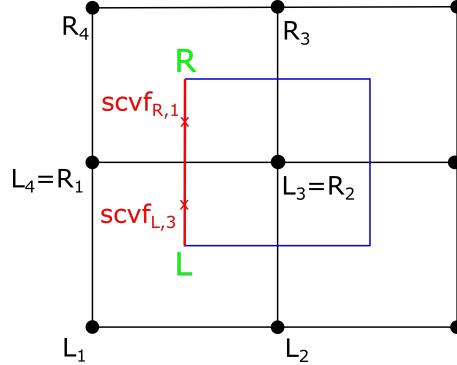


Figure D.2: Local stencil for a boundary volume.

Similarly, the  $y$ -gradient at this point reads:

$$\frac{\partial \vec{U}}{\partial y}_{scvf_{R,1}} = \frac{1}{2} \frac{\vec{U}_{R_4} - \vec{U}_{R_1}}{\Delta y} + \frac{1}{2} \frac{\vec{U}_{R_3} - \vec{U}_{R_2}}{\Delta y} \quad (\text{D.17})$$

Similar expressions hold for the other  $scvfs$  in a cell.

These expressions can also be written in local matrix form, for the evaluation of the gradient in  $x$ -direction (and  $y$ -direction) at the  $scvfs$  of a boundary volume. The local gradient (coefficient) matrix  $\bar{A}_{BV}^{\nabla x}$  then reads:

$$\bar{A}^{\nabla x} = \frac{1}{4} \begin{pmatrix} -3 & 3 & 1 & -1 \\ -2 & 2 & 2 & -2 \\ -1 & 1 & 3 & -3 \\ -2 & 2 & 2 & -2 \end{pmatrix} \quad (\text{D.18})$$

for the four  $scvfs$  in a cell.

Similarly for the gradient in  $y$ -direction, the local gradient (coefficient) matrix becomes:

$$\bar{A}^{\nabla y} = \frac{1}{4} \begin{pmatrix} -2 & -2 & 2 & 2 \\ -1 & -3 & 3 & 1 \\ -2 & -2 & 2 & 2 \\ -3 & -1 & 1 & 3 \end{pmatrix} \quad (\text{D.19})$$

To be able to compute the gradients of the possibly nonlinear fluxes, we write the

gradients of the flux in quasi-linear form. For the  $x$ -gradient this becomes:

$$\frac{\partial \vec{F}}{\partial x} = \frac{\partial}{\partial x} \begin{pmatrix} q \\ q^2/h \\ qr/h \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -q^2/h^2 & 2q/h & 0 \\ qr/h^2 & r/h & q/h \end{pmatrix} \begin{pmatrix} h_x \\ q_x \\ r_x \end{pmatrix} \quad (\text{D.20})$$

and similarly for the gradient in  $y$ -direction:

$$\frac{\partial \vec{F}}{\partial y} = \frac{\partial}{\partial y} \begin{pmatrix} r \\ qr/h \\ r^2/h \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ -qr/h^2 & r/h & q/h \\ -r^2/h^2 & 0 & 2r/h \end{pmatrix} \begin{pmatrix} h_y \\ q_y \\ r_y \end{pmatrix} \quad (\text{D.21})$$

Using these expressions, the evaluation of the gradients of the flux is transformed to an evaluation of the terms appearing in these quasi-linear forms (non-gradients) and evaluation of the gradients of the linear terms  $h$ ,  $q$  and  $r$ .

For instance, at  $scvf_{R,1}$  as in Figure D.2, the quasi-linear form for the  $x$ -gradient of the  $x$ -momentum flux  $q^2/h$  becomes:

$$\left. \frac{\partial q^2/h}{\partial x} \right|_{scvf_{R,1}} = (-q^2/h^2, 2q/h, 0)^T \Big|_{scvf_{R,1}} \cdot \begin{pmatrix} h_x \\ q_x \\ r_x \end{pmatrix}_{scvf_{R,1}} \quad (\text{D.22})$$

where the interpolation of the values to the  $scvf.c$  is done as in B.9 and where the gradients of the individual terms ( $h$ ,  $q$  and  $r$ ) is realized using expressions (D.18) and (D.19).

Similar expressions can be formed for the  $y$ -gradients, as well as for the gradients at  $scvf_{L,3}$  and for the other equations in the system, as a function of the values of  $\vec{U} = (h, q, r)^T$  in nodes  $L_1$ ,  $L_2$ ,  $L_3$  and  $L_4$ .

These expressions are all substituted in (D.15) to complete the spatial discretization part for the boundary volume.

The first term in (C.5) contains values at time level  $n + 1$ . Substituting the local interpolation matrices in this term, results in a mass matrix. The second term at time level  $n$  results in a right-hand-side contribution.

The flux term in the equation, has not yet been discretized in time. Since the flux term for the 2D shallow-water equations is nonlinear, an implicit or semi-implicit (e.g.  $\theta$ ) method would require the solution of a nonlinear system, for instance using Newton's method.

For simplicity, therefore, we now consider only an explicit time integration of the flux, resulting in a contributions to the right-hand side of the linear system only.

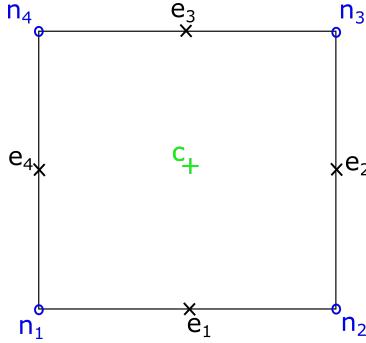


Figure D.3: Definitions of the node, edge and cell centre positions and numbering (for a quadrilateral cell).

The discretization of the system of equations for a single  $scvf$  (here  $scvf,R1$  from Figure D.2) contributing to the integral over the boundary volume, becomes:

$$\begin{aligned} & \frac{1}{8\Delta t} \left( 3(\vec{U}_{R1})^{n+1} + 3(\vec{U}_{R2})^{n+1} + (\vec{U}_{R3})^{n+1} + (\vec{U}_{R4})^{n+1} \right. \\ & \quad \left. - 3(\vec{U}_{R1})^n - 3(\vec{U}_{R2})^n - (\vec{U}_{R3})^n - (\vec{U}_{R4})^n \right) l_{scvf,R1} \\ & \quad + \frac{\partial}{\partial x} \left( \begin{array}{c} q \\ q^2/h \\ qr/h \end{array} \right) \Big|_{R1}^n + \frac{\partial}{\partial y} \left( \begin{array}{c} r \\ qr/h \\ r^2/h \end{array} \right) \Big|_{R1}^n \end{aligned} \quad (D.23)$$

or

$$\begin{aligned} & \frac{1}{8\Delta t} \left( 3(\vec{U}_{R1})^{n+1} + 3(\vec{U}_{R2})^{n+1} + (\vec{U}_{R3})^{n+1} + (\vec{U}_{R4})^{n+1} \right. \\ & \quad \left. - 3(\vec{U}_{R1})^n - 3(\vec{U}_{R2})^n - (\vec{U}_{R3})^n - (\vec{U}_{R4})^n \right) l_{scvf,R1} \\ & \quad + \frac{\partial}{\partial x} \left( \begin{array}{ccc} 0 & 1 & 0 \\ -q^2/h^2 & 2q/h & 0 \\ -qr/h^2 & r/h & q/h \end{array} \right) \left( \begin{array}{c} h_x \\ q_x \\ r_x \end{array} \right) \Big|_{R1}^n \\ & \quad + \frac{\partial}{\partial y} \left( \begin{array}{ccc} 0 & 0 & 1 \\ -qr/h^2 & r/h & q/h \\ -r^2/h^2 & 0 & 2r/h \end{array} \right) \left( \begin{array}{c} h_y \\ q_y \\ r_y \end{array} \right) \Big|_{R1}^n \end{aligned} \quad (D.24)$$

where the numbering of the four nodal values of  $c$  of cell  $R$  is defined in Figure D.2. Again, the interpolation of the values to the  $scvf,c$  is done as in B.9 and where the

gradients of the individual terms ( $h$ ,  $q$  and  $r$ ) is realized using expressions (D.18) and (D.19).

This equation provides the matrix and vector coefficients for the degree of freedom corresponding to the virtual node that is directly adjacent to the boundary volume (i.e. closest to the *scvf* centre), in this case for  $R_1$ , which is the same node as  $L_4$ .

### D.3.1.2 Discretization at corner volumes

In the previous section, we have seen how the model equations are discretized at boundary volumes, i.e. at line segments between virtual and internal nodes. The equations that follow from these discretizations correspond to the degrees of freedom of the virtual nodes adjacent to those boundary volumes. The virtual nodes that are in the corner of the domain do not have a directly connected internal node. However, they do have a corresponding degree of freedom and, therefore, also require a representative equation. For this purpose, the notion of a *corner volume* is introduced. It is a "point volume" that is used for discretizing the model equations, corresponding to the degree of freedom of the virtual corner node. The discretization is built up very similar to that of the boundary volume. The flux term, containing the gradient, is again simply evaluated at the corner volume  $CV$ :

$$\frac{\vec{U}_{CV}^{n+1} - \vec{U}_{CV}^n}{\Delta t} + \nabla \cdot \vec{F} \Big|_{CV} = 0 \quad (\text{D.25})$$

The position of a corner volume  $CV$  is at the centre of a cell (of the primary mesh). The interpolation is thus a straightforward 4-point average. And the gradient 'matrices' (now vectors, because they are identical for all *scvfs*) are simply:

$$\bar{A}_c^{\nabla x} = \frac{1}{2} \begin{pmatrix} -1 & 1 & 1 & -1 \end{pmatrix} \quad (\text{D.26})$$

for all four *scvfs* in a cell.

Similarly for the gradient in  $y$ -direction, the local gradient (coefficient) matrix (actually vector) becomes:

$$\bar{A}_c^{\nabla y} = \frac{1}{2} \begin{pmatrix} -1 & -1 & 1 & 1 \end{pmatrix} \quad (\text{D.27})$$

again for all four *scvfs* in a cell.

## D.3.2 Applying the user-specified boundary conditions

The next step is to introduce possible user-prescribed boundary conditions. For now, we limit ourselves to Dirichlet- or Neumann-type boundary conditions. The discretizations set up for nodes that have such a boundary specification overwrite the discretizations constructed in section D.3.1.

### D.3.2.1 Dirichlet-type conditions

For a Dirichlet-type condition, the value of a quantity  $c$  (e.g. the water depth  $h$ ) at the boundary is prescribed. This can be written as:

$$\alpha c_v^{n+1} + (1 - \alpha) c_{in}^{n+1} = c_{given} \quad (\text{D.28})$$

where  $c_{given}$  is the user-prescribed value for quantity  $c$ . The coefficient  $\alpha$  accounts for the position of the actual boundary, with respect to the virtual and inner nodes. For  $\alpha$  is 1, the genuine boundary coincides with the virtual node position and for  $\alpha = 0$ , the genuine boundary is located at the inner node, see Figure D.1. A value of  $\alpha$  between 0 and 1 can handle a positioning of the genuine boundary somewhere between the inner and virtual nodes.

This equation overwrites the equation constructed from the discretization of the model equations in section D.3.1. For systems of equations, it might be convenient to combine the discretized model equations with the Dirichlet condition. This will be discussed in a separate chapter.

### D.3.2.2 Neumann-type conditions

For a Neumann-type condition, the gradient of some quantity  $c$  at the boundary is prescribed. This can be written as:

$$\vec{n}_{BV} \frac{c_{in}^{n+1} - c_v^{n+1}}{\Delta s} = \nabla c_{given} \quad (\text{D.29})$$

where  $\vec{n}_{BV}$  accounts for the orientation of the boundary volume,  $\Delta s$  is the local grid size (for 1D transport, this can only be  $\Delta x$  and where  $\nabla c_{given}$  is the user-prescribed value for the gradient of the concentration  $\partial c / \partial x$ .

This equation overwrites the equation constructed from the discretization of the model equations in section D.3.1. For systems of equations, it might be convenient to combine the discretized model equations with the Neumann condition. This will be discussed in a separate chapter.

## D.4 Solving loosely-coupled systems

When solving systems of equations in a loosely-coupled way, by solving one sub-system after another, the aforementioned solution algorithm needs to be generalized slightly. For instance, when solving the 2D shallow-water equations, by first solving the continuity equation (for an updated value of the water depth  $h$ ) and then solving the (system of) momentum equations (for updated values of unit discharges  $q$  and  $r$ ), the sub-systems (continuity and momentum) have their own state vector of primary variables  $\vec{PV}$ . The primary variables are those that the sub-system solves for. Additionally, each sub-system has a vector of secondary variables  $\vec{SV}$ , that couple it to the other equation(s).

In this case, for 2D shallow-water (loosely coupled):

Continuity:

$$\begin{aligned}\vec{PV}^{cont} &= (h) \\ \vec{SV}^{cont} &= (q, r)^T\end{aligned}$$

Momentum:

$$\begin{aligned}\vec{PV}^{mom} &= (q, r)^T \\ \vec{SV}^{mom} &= (h)\end{aligned}$$

Or in a general form (for now for 2 sub-systems, assuming the same splitting as before):

Sub-system 1:

$$\begin{aligned}\vec{PV}^{sub1} &= (u_1) \\ \vec{SV}^{sub2} &= (u_2, u_3)^T\end{aligned}$$

Sub-system 2:

$$\begin{aligned}\vec{PV}^{sub1} &= (u_2, u_3)^T \\ \vec{SV}^{sub2} &= (u_1)\end{aligned}$$

With such definitions, the quasi-linear form of the fluxes, as described in Section D.3.1, also needs to be split in a part containing derivatives with respect to the primary

variables and a part with derivatives with respect to the secondary variables.

Taking as example the (advective) flux in  $x$ -direction:

$$\vec{F}^x = \bar{J} \cdot \vec{U}_x = \bar{J}^{sub1} \cdot \vec{U}_x^{sub1} + \bar{J}^{sub2} \cdot \vec{U}_x^{sub2} \quad (\text{D.30})$$

where  $\bar{J}$  is the flux Jacobian, containing the derivatives of the fluxes with respect to the primary variables. If we now again assume that  $\vec{U}^{sub1} = (u_1)$  and  $\vec{U}^{sub2} = (u_2, u_3)^T$ , we get:

$$\vec{F}^x = \begin{pmatrix} \frac{\partial F_1}{\partial u_1} \\ \frac{\partial F_2}{\partial u_1} \\ \frac{\partial F_3}{\partial u_1} \end{pmatrix} \cdot (u_1)_x + \begin{pmatrix} \frac{\partial F_1}{\partial u_2} & \frac{\partial F_1}{\partial u_3} \\ \frac{\partial F_2}{\partial u_2} & \frac{\partial F_2}{\partial u_3} \\ \frac{\partial F_3}{\partial u_2} & \frac{\partial F_3}{\partial u_3} \end{pmatrix} \cdot \begin{pmatrix} u_2 \\ u_3 \end{pmatrix}_x \quad (\text{D.31})$$

For the 2D shallow-water equations, this becomes:

$$\vec{F}^x = \begin{pmatrix} 0 \\ -q^2/h^2 \\ -qr/h^2 \end{pmatrix} \cdot (h)_x + \begin{pmatrix} 1 & 0 \\ 2q/h & 0 \\ r/h & q/h \end{pmatrix} \cdot \begin{pmatrix} q \\ r \end{pmatrix}_x \quad (\text{D.32})$$

or split in the equations (or systems) that are solved separately:

$$\vec{F}^{cont,x} = (0) \cdot (h)_x + (1 \ 0) \cdot \begin{pmatrix} q \\ r \end{pmatrix}_x \quad (\text{D.33})$$

and

$$\vec{F}^{mom,x} = \begin{pmatrix} 2q/h & 0 \\ r/h & q/h \end{pmatrix} \cdot \begin{pmatrix} q \\ r \end{pmatrix}_x + \begin{pmatrix} -q^2/h^2 \\ -qr/h^2 \end{pmatrix} \cdot (h)_x \quad (\text{D.34})$$

where it can be seen that each flux (in both equations) is composed from a primary variables (gradient) part and a secondary variables (gradient) part.

Similarly, the  $y$ -flux can then be written as:

$$\vec{F}^{cont,y} = (0) \cdot (h)_y + (0 \ 1) \cdot \begin{pmatrix} q \\ r \end{pmatrix}_y \quad (\text{D.35})$$

and

$$\vec{F}^{mom,y} = \begin{pmatrix} r/h & q/h \\ 0 & 2r/h \end{pmatrix} \cdot \begin{pmatrix} q \\ r \end{pmatrix}_y + \begin{pmatrix} -qr/h^2 \\ -r^2/h^2 \end{pmatrix} \cdot (h)_y \quad (\text{D.36})$$

## D.5 Numerical experiments

... To do ...

## D.6 Results

... To do ...

## D.7 Analysis

... To do ...

## D.8 Discussion

... To do ...

## D.9 Conclusions

... To do ...



