SWASH

SCIENTIFIC
AND
TECHNICAL
DOCUMENTATION

SWASH version 11.01

by : The SWASH team

mail address : Delft University of Technology
Faculty of Civil Engineering and Geosciences
Environmental Fluid Mechanics Section
P.O. Box 5048
2600 GA Delft
The Netherlands

website : *http://swash.sourceforge.net*

Contents

Chapter 1

Introduction

The main goal of the SWASH model is to solve the nonhydrostatic, nonlinear, shallow water equations on a regular grid.

to be filled in...

1.1 Historical background

This section is under preparation.

1.2 Purpose and motivation

The purpose of this document is to provide relevant information on the mathematical models and numerical techniques for the simulation of shallow water in coastal regions. Furthermore, this document explains the essential steps involved in the implementation of various numerical methods, and thus provides an adequate reference with respect to the structure of the SWASH program.

1.3 Readership

This document is, in the first place, addressed to those, who wish to modify and to extend mathematical and numerical models for shallow water problems. However, this material is also useful for those who are interested in the application of the techniques discussed here. The text assumes the reader has basic knowledge of analysis, partial differential equations and numerical mathematics and provides what is needed both in the main text and in the appendices.

1.4 Scope of this document

SWASH is a general-purpose numerical tool for simulating unsteady, non-hydrostatic, free-surface, rotational flow and transport phenomena in coastal waters as driven by waves, tides, buoyancy and wind forces. It provides a general basis for describing wave transformations from deep water to a beach, port or harbour, complex changes to rapidly varied flows, and density driven flows in coastal seas, estuaries, lakes and rivers.

1.5 Overview

The remainder of this document is subdivided as follows: In Chapter 2 a review of considerations from the Hamiltonian formalism and algebraic topology of the inviscid shallow water equations is provided. This chapter explains why the Arakawa C-grid discretization method was chosen as the basis for the design of SWASH. In Chapter 6 the three-dimensional shallow water equations used in SWASH are presented. These underlying equations and the derivation thereof, i.e. the layer-averaged equations, have been discussed earlier in the Technical documentation of TRIWAQ-in-SIMONA [72] and was written by Marcel Zijlema in 1998. After that this outline has been applied successfully in SWASH. See also the papers [77, 78, 57, 79]. In Chapter 7 the main characteristics of the finite difference method for the discretization of the governing equations in horizontal planes are outlined. Various differencing schemes for spatial propagation are reported. Chapter 8 is concerned with discussing several boundary conditions and their implementation. Chapter 9 is devoted to the linear solvers for the solution of the resulted linear systems of equations. Chapter 10 deals with some consideration on parallelization of SWASH on distributed memory architectures.

This document, however, is not intended as being complete. Although, this document describes the essential steps involved in the simulation of waves, so that the user can see which can be modified or extended to solve a particular problem properly, some subjects involved in SWASH are not included. Below, a list of these subjects is given, of which the information may be available elsewhere (e.g., journal and proceedings papers):

- wave damping induced by vegetation,
- partial reflection and transmission,
- subgrid approach for 3D wave-induced currents,
- floating objects.

1.6 Acknowledgements

The SWASH team are grateful to the original authors from the very first days of SWASH which took place at the Delft University of Technology in Delft, The Netherlands in 2002:

Guus Stelling and Marcel Zijlema.

We further want to acknowledge all contributors who helped us to improve SWASH, reported bugs, and tested SWASH thoroughly: Pieter Smit, Dirk Rijnsdorp, Tomo Suzuki, Panagiotis Vasarmidis, and Joao Dobrochinski.

We are finally grateful to all those other people working on the Public Domain Software without which the development of SWASH would be unthinkable: Linux, Intel, GNU F95, LaTeX, MPICH, Perl and many others.

Chapter 2

Physics-compatible discretizations on simplicial and cubical meshes

2.1 Introduction

This chapter deals with the numerical solution of the two-dimensional nonlinear shallow water equations that form the basis for SWASH. The spatial discretization is based on the staggered Arakawa C-grid finite difference method for orthogonal triangular, rectangular and curvilinear meshes. It is known for a long time that this method exhibits beneficial properties in a wide range of shallow water applications, including nonlinear wave transformation as characterized by energy transfer between the different wave components. This enhances the robustness of the SWASH model. This chapter explains the reasons why this is so. In the following sections below, we will set out a number of relevant topics in depth which are crucial for the exposition of this chapter. The topics covered are related to Hamiltonian formalism and algebraic topology.

There are two issues that play a key role. First there is the issue of the nonlinear computational instability that frequently occurs in the numerical simulation of highly nonlinear shallow water systems, and secondly, the importance of primary and secondary conservation properties that appear naturally in physics and geometry. This dual role underlies a growing body of literature which clearly demonstrates that mimicking the conservation properties of the continuous partial differential equations (PDEs) at the discrete level eliminates the problem of nonlinear instability.

One of the earliest studies on nonlinear computational instability of finite difference schemes was conducted by Phillips in the 1950s [48]. This phenomenon contrasted with the usual (linear) stability that can easily be controlled by reducing the time step. Phillips explained this then new kind of instability in terms of aliasing. Numerical waves shorter than two grid sizes are misinterpreted by the finite grid as long waves and thus create spurious interactions which, according to Phillips, cause the observed instability. Since the nonlinear instability could not be eliminated by decreasing the time step, Phillips applied a smoothing technique to diminish the instability.

Although Phillips "aliasing" clarification could be a plausible one, however, in reality it does not. The solution to the problem of nonlinear computational instability came from Lilly [32] and Arakawa [1]. They demonstrated the cause of this instability to be the lack of conservation of kinetic energy (and vorticity), despite the presence of aliasing errors. The spectral analysis of Lilly [32] further substantiated that a correct redistribution of energy over the scales of motion is closely related to the conservation of kinetic energy and, in turn, eliminates the nonlinear instability. Arakawa later on showed that the staggered C-grid approach has proven to be effective in eliminating the problem of nonlinear computational instability [2, 3]. We will discuss this topic in detail later.

The shallow water equations involve a number of differential operators such as the gradient and the divergence. Basically, such operators are mathematical constructs based on the notion of limit (infinitesimal cube contracting to a point) and contain a number of *hidden* geometrical and physical structures, such as symmetries and conservation properties. The key purpose of algebraic topology in the present work is to reveal these mathematical structures by studying geometric objects. This then forms the starting point for the construction of discrete counterparts of the continuous differential operators, namely, the gradient, curl, and divergence. These operators are referred to as `grad`, `curl` and `div`, respectively.

While there ary many ways to *approximate* the PDEs and their associated operators, such as the finite difference, finite volume, and finite element methods, algebraic topology offers a mimetic approach to their construction in the sense that discrete operators truly mimic the behavior of the differential operators regardless of the mesh type and resolution [23]. Such mimetic discretizations also preserve vector calculus identities, including $\texttt{curl grad} = 0$ and $\texttt{div curl} = 0$, and symmetry relations such as $\texttt{curl} = \texttt{curl}^{\mathsf{T}}$ and $\texttt{div} = -\texttt{grad}^{\mathsf{T}}$. For instance, the latter antisymmetry property is closely related to the Hamiltonian structure of the inviscid shallow water equations which means that the total energy of the system is conserved. As we will see later, by embedding these discrete structures into a discretization process, they capture the essential physics of the PDEs and can have a stabilizing effect on the solution of PDEs.

The development of mimetic discretizations is an active field of research where it is linked to the high demand for physically reliable simulation models to describe and predict complex systems arising in oceanographic and atmospheric flow problems, direct numerical and large-eddy simulations of turbulent flows, but also computer graphics. Some of the contributions in this area have come in the form of mimetic finite differences [4, 61, 33], the support operators method (SOM) [58, 23], mimetic spectral elements [11, 12, 29, 46], discrete exterior calculus [21, 14, 15, 22, 38], and physics-compatible or symmetry-preserving discretizations [49, 66, 67, 65, 6]. Such numerical techniques are especially useful when grid refinement or increasing the order of the discretization is insufficient to resolve the wide range of scales of nonlinear motions (e.g., high-Re turbulent flows, multi-scale atmospheric flows, nonlinear wave-wave and wave-current interactions). Also, nonlinear energy transfer between scales is generally respected by mimetic discretizations which not only promotes the physical fidelity but also aids in the stability of the model simulation.

Let us put this into perspective by situating these mimetic discretizations in relation

to other conventional finite difference and finite volume methods. The latter methods are widely adopted for approximating the shallow water equations on horizontal grids. Arakawa and Lamb[2] define five grid systems (A to E) based on the horizontal staggering of the primitive variables (the velocity vector and the water level). Of these five grids, the unstaggered (or colocated) Arakawa A-grid, the semi-staggered Arakawa B-grid and the staggered Arakawa C-grid are the most prominent ones in CFD and computational hydraulics. With the A-grid, the water level and the components of the velocity vector are stored at the same grid vertices or cell centers. The B-grid places water levels at the corners of cells and the velocity vector at the centers of cells or the other way around. The C-grid evaluates the normal components of the horizontal velocity at the centers of the cell faces and the water levels at the cell centers.

The usual strategy in the development of these discrete frameworks is that first a discretization method is constructed in a mathematical fashion using high-resolution schemes but without an explicit reference to the physical properties that underlie the continuous flow field problem. Next, certain numerical (mostly linear) analysis tools are utilized to prove its accuracy, stability and convergence in the sense of the Lax's equivalence theorem. The hope is then that a numerical solution to the considered PDEs is obtained that is physically realistic, especially when problems with strong nonlinearities [71, 70] are relevant. There are, however, three issues that complicate matters related to controlling the convergence error by mesh refinement.

First, there are ambiguities regarding the validity of the equivalence theorem in the case of nonlinear PDEs. At least, it seems that this theorem can only provide the *necessary* conditions for convergence. Numerical schemes can still fail to capture physically consistent results.

Second, a high order accurate approximation is assumed to be better in the sense that its solution converges faster compared to a low order scheme owing to the lower truncation errors. However, this premise is exceptional, especially when nonlinearity plays a significant role. A key aspect of this that is often overlooked is the necessity to have mesh spacing substantially small to achieve the nonlinear solution convergent at best.

Third, the numerical method established in this way may not obey some of the conservation laws, identities and symmetries and can thus act as a spurious source of mass, momentum or energy. For example, both A-grid and B-grid discretizations ultimately build on *approximating* the conservation of mass and energy [40, 16]. Furthermore, symmetry relations, like $\mathtt{div} = -\mathtt{grad}^{\mathsf{T}}$, cannot be satisfied while the associated discrete operators support spurious computational (or checkerboard pressure) modes. These unphysical modes are typically inert at the grid scale and can contaminate the numerical solution in the long run as various nonlinear processes, including physics parametrizations and bathymetric forcing, can excite them [31]. Though colocated (A-grid) and semi-staggered (B-grid) discretization methods routinely suppress erroneous grid-scale oscillations by some degree of non-physical dissipation, either upwind differencing or space-centred approximation with artificial viscosity, such kind of regularization usually have difficulty to moderate the stationary spurious modes as they do not propagate.

There is a scarcity of literature that discusses the development of colocated (A-)grid

discretizations of the inviscid shallow water equations on general meshes. By contrast, the colocated central discretization method that employs the classical Von Neumann and Richtmyer's artificial viscosity [68] or its variants (e.g., the successful JST scheme of Jameson [24]) for identifying shock waves is very useful for solving the compressible Euler equations at high Mach numbers. This suitability is explained by the fact that the associated physics typically involves a high energetic primary mode and relatively small higher modes. In turn, the related nonlinear cascade of wave energy is less pronounced than that of incompressible flows, which allows the use of less far-reaching discretization methods, including the Lax-Wendroff type method [32] and the A-grid method.

The C-grid discretization is superior to both A-grid and B-grid regarding the accuracy and stability in solving the *highly nonlinear* shallow water equations. Staggered C-grid schemes are practically stable as they exactly conserve discrete analogues of mass and energy and do not typically generate spurious modes. An example is the celebrated finite difference scheme of Arakawa and Lamb [3] for the rotating shallow water equations on Cartesian staggered grids. It does not only conserve mass and energy exactly but also vorticity and enstrophy. Furthermore, this staggered scheme is completely free of unphysical modes. In this sense, the Arakawa and Lamb scheme can be considered as one of the earliest mimetic discretization methods for free surface flows.

Despite these advantages, staggered C-grid methods tend to have a low order of accuracy, especially on nonuniform meshes. Yet, they often produce smaller global truncation errors than other traditional (usually non-mimetic) methods of the same or higher order even on nonuniform grids [66]. This is because of the fact that the associated discrete operators exactly represent conservation properties (mass and energy), vector calculus identities, including the vanishing of the curl of the gradient of any scalar field, and fundamental symmetries, most notably the divergence is the negative transpose of the gradient. These specific properties contribute at improving the physical accuracy of nonlinear flow and transport problems at diverse space and time scales. In essence, they generally improve simulation fidelity and thus potentially increase physical reliability regardless of the chosen resolution in the simulations.

Additionally, previous studies like Manteuffel and White [34] have demonstrated that low order schemes can achieve second order accuracy on nonuniform grids. Still, high order acurrate schemes can be desirable when one wants to avoid the use of excessively fine meshes, especially Cartesian grids. It should be noted, however, that unstructured mesh methods typically do not allow for ease of implementation of high order discretizations as they do not take full advantage of higher order accuracy that can be easily achieved on structured rectangular grids. On the other hand, unstructured meshes have their unique quality to easily enhance the flexibility by allowing local mesh refinements. For this reason, we will also present an extension of the classical staggered C-grid approach to unstructured triangular grids. This extended method is referred to as the covolume method and is described in detail in Chapter 5.

Over the years, successful staggered C-grid schemes have been developed for the simulation of incompressible flows on curvilinear staggered grids [69, 64, 76] and on unstructured triangular Delaunay-Voronoi meshes [18, 44, 45, 10, 47, 26], modelling of large-scale ocean and

small-scale coastal flows on both orthogonal curvilinear grids, see, e.g., [30, 55, 7, 53, 56, 79], and unstructured triangular meshes, e.g. [8, 9, 17, 59, 28, 25, 27, 20, 74]. Additionally, many papers have been published over the last few decades on the use of Arakawa C-grid discretizations for large-scale atmospheric flows on the sphere using arbitrarily structured (hexagonal) meshes, see, e.g., [5, 60, 62, 50, 52, 11, 61]).

This chapter provides support for a physically based strategy to develop numerical methods that are capable of dealing with symmetries and conservation properties at a discrete level. These methods do not discretize the continuous PDEs in the traditional sense with scalars and vectors as fundamental entities of differential calculus. Rather, they are driven by the topological interpretation of the physical fields as discrete differential forms. Such forms are the integrals of the physical quantities over the various geometric elements (points, curves, planes and volumes) and constitute a discrete representation for solution fields over discretized (mesh) objects (vertices, edges, faces, and cells).

The notion of discrete differential forms is at the heart of algebraic topology. The framework of algebraic topology provides the basis for the development of mimetic discretizations used in this work. As we will see later on, this goal serves as the basis and justification for using staggered grids. The importance of the discrete forms becomes apparent in identifying which parts of the PDEs are conservation laws that do not depend on any notion of a metric, and which parts are relationships that are approximative by nature such as the material constitutive relations and the local relationships between the various physical quantities due to inhomogeneous media (e.g., non-uniform depth and fluid density). The discretizations are then constructed to exactly satisfy the former, that is, without any discretization error involved, and accurately approximate the latter. As a result, they aim to mimic the fundamental properties of the continuous differential operators `grad`, `curl` and `div`. Furthermore, certain crucial symmetry relations, like for instance $\texttt{div} = -\texttt{grad}^{\mathsf{T}}$, are respected at the discrete level, and these, in turn, contribute to the nonlinear computational stability.

This chapter begins with the formulation of the inviscid nonlinear shallow water equations; they are covered in detail in Section 2.2. Next, Section 2.3 reveals the mathematical structure of the governing equations, namely, the Hamiltonian which represents the total energy of the system, and then deals with some theoretical aspects of the Hamiltonian formalism. The use of the Hamiltonian form is beneficial since it provides conditions for the stability of the spatial discretization of the shallow water equations.

Mimetic discretization methods aim to preserve essential geometrical and physical structures in a discrete setting. The core rationale here is the agreement of the numerical solution with physical measurements rather than convergence to an exact solution of PDEs. As a preliminary to this approach, we informally introduce the two essential notions of differential geometry, namely, differential forms and generalized Stokes' theorem. These physically based concepts are addressed in Section 2.4. This is then followed by an extensive review of some fundamental concepts from algebraic topology, which is the discrete counterpart of differential geometry. They serve as the building blocks of the discretization infrastructure. Section 2.5 elaborates upon this matter.

Finally, Section 2.6 discusses a general mimetic framework for the inviscid nonlinear

shallow water equations which will be used to derive the staggered Arakawa C-grid for rectilinear grids in Chapter 3 and for curvilinear grids in Chapter 4, and the covolume method for unstructured triangular meshes in Chapter 5.

This chapter (and also Chapters 3, 4 and 5) focusses on the spatial discretization in the horizontal for both 2DH and 3D shallow water equations. Discretization in the vertical dimension for 3D flow domains will be dealt with in Chapter 6.

2.2 Inviscid shallow water equations

(Un)SWASH solves the two- and three-dimensional nonlinear shallow water equations. These equations describe the behavior of a shallow incompressible fluid layer and are suitable to model hydrodynamics in coastal seas, estuaries, lakes and rivers. They are derived from the depth-integrated Euler or Navier-Stokes equations under the hydrostatic pressure assumption. The equations of motion are commonly written in the language of vector calculus.

For applications to water waves we deal with the barotropic flow of an incompressible fluid in a two-dimensional bounded domain, denoted by $\Omega \subset \mathbb{R}^2$, with a thin layer of water between a rigid bottom at $z = -d(\mathbf{x})$ and a single-valued free surface $\zeta(\mathbf{x}, t)$ where $\mathbf{x} = (x, y) \in \Omega$ indicates the horizontal position. The inviscid shallow water equations in the flux-form are given by

$$\frac{\partial h}{\partial t} + \nabla \cdot \mathbf{q} = 0 \tag{2.1}$$

$$\frac{\partial h\mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{q} \otimes \mathbf{u}) = -gh\nabla\zeta \tag{2.2}$$

where $h = \zeta + d$ is the water depth and $\mathbf{u} = (u, v)$ is the depth-averaged flow velocity vector with the components $u(\mathbf{x}, t)$ and $v(\mathbf{x}, t)$ along the x and y coordinates, respectively, as given by

$$\mathbf{u}\,(\mathbf{x}, t) = \frac{1}{h}\int_{z=-d}^{z=\zeta} \mathbf{v}\,(\mathbf{x}, z, t)\,dz$$

with $\mathbf{v}(\mathbf{x}, z, t)$ the three-dimensional flow velocity. Note that for water waves the three-dimensional flow is considered to be irrotational, that is, $\nabla \times \mathbf{v} = 0$. However, $\nabla \times h\mathbf{u} \neq 0$. Furthermore, $\mathbf{q} = h\mathbf{u}$ is the mass flux, $\nabla = (\partial_x, \partial_y)$ is the two-dimensional gradient operator on Ω, and finally, g is the gravitational acceleration. The governing equations are combined with appropriate boundary conditions. This is discussed in Chapter 8.

Eqs. (2.1) and (2.2) naturally describe the water wave motion on top of the ambient flow. The essential terms here are the pressure gradient term in the right-hand side of Eq. (2.2) and the divergence of the mass flux, the second term of Eq. (2.1). Mathematically, they are adjoint to each other; see Section 2.3 for further clarification.

The quantity $h\mathbf{u}$ in the first term of Eq. (2.2) represents the depth-integrated velocity along a path of fluid motion while the pressure gradient is a driving force due to the surface

slope along the flow line. The second divergence term of Eq. (2.2) can be expanded as

$$\nabla \cdot (\mathbf{q} \otimes \mathbf{u}) = (\mathbf{q} \cdot \nabla)\ \mathbf{u} + (\nabla \cdot \mathbf{q})\ \mathbf{u}$$

The first term in the right-hand side describes advection in the background flow while the second term is linked to the wave dynamics. Additionally, the combination of the terms $\nabla \cdot (\mathbf{q} \otimes \mathbf{u})$ and $gh\nabla\zeta$ in the momentum equation (2.2) characterizes the embedding of the multi-scale interactions between the various wave components.

As demonstrated above, the depth-averaged velocity $\mathbf{u}$ is transported by the mass flux $\mathbf{q}$. Although the reversed statement, that is, $h\mathbf{u} = \mathbf{q}$ is the conserved quantity that is advected by the velocity $\mathbf{u}$, might makes sense, as suggested by Eq. (2.2), it is actually wrong from a physical point of view. This is because $h\mathbf{u}$ is not the physical quantity of a fluid particle, but instead the quantity $\mathbf{u}$ is, or rather $\mathbf{v}$, which is also conserved by advection.

2.3 Hamiltonian formulation

In this section we demonstrate how, using the Hamiltonian formalism, we can systematically derive conditions required for the conservation of energy that can be used to construct mimetic discretizations of the inviscid nonlinear shallow water equations on general meshes. Though energy is usually not preserved in the majority of coastal water systems, energy conservation conceived as a *constraint* is relevant in view of the spatial discretization for two reasons. First, it can guarantee the stability of the discretization. Second, on physical grounds, it ensures that energy is conservatively transferred from low wave frequencies to high frequencies, which then causes waves to break, and dissipation of wave energy. This nonlinear energy cascade requires certain contributions to the governing equations (2.1)−(2.2) to be independently energy conserved, namely, the pressure gradient and the advective transport of momentum. When mimicking this requirement at a discrete level, it thus reflects the physical fidelity of the discretization.

Like many physical systems, the inviscid, barotropic shallow water equations (2.1)−(2.2) possess a Hamiltonian structure (see, e.g. [13]). In the absence of shocks and a horizontal frictionless bed, this system conserves the total energy, or Hamiltonian, which is the sum of the kinetic energy and gravitational potential energy per unit volume

$$\int_\Omega d\mathbf{x} \int_{z=-d}^{z=\zeta} dz\ \left[\frac{1}{2}\mathbf{u}\cdot\mathbf{u} + gz\right]$$

Since the equations of motion are described using the field variables h and $\mathbf{u}$, their Hamiltonion structure is of a non-canonical (or generalized) form. This is explained further below.

The exposition starts by first considering an infinite-dimensional real vector space $\mathcal{V}$ of fields equipped with an inner product (called a Hilbert space) defined on some domain $\mathbf{x} \in \Omega$ in $\mathbb{R}^2$. We establish the inner product $\langle \cdot\,, \cdot \rangle : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ in the following way. We have

$$\langle f, g \rangle = \int_\Omega f\, g\, d\mathbf{x} \tag{2.3}$$

for scalar fields f and g on Ω, and

$$\langle \mathbf{v}, \mathbf{w} \rangle = \int_\Omega \mathbf{v} \cdot \mathbf{w} \, d\mathbf{x} \tag{2.4}$$

for vector fields $\mathbf{v}$ and $\mathbf{w}$ on Ω with $\cdot$ denoting the standard element-wise dot product. Note that the inner product is positive definite and symmetric.

Next, a key assumption is made that the scalar and vector fields have a compact support, that is, they vanish on the boundary of Ω. Let us integrate the following vector calculus identity over Ω,

$$\nabla \cdot (f\mathbf{v}) = f \nabla \cdot \mathbf{v} + (\nabla f) \cdot \mathbf{v}$$

and subsequently apply the divergence theorem. We obtain

$$\int_\Omega f \nabla \cdot \mathbf{v} \, d\mathbf{x} + \int_\Omega (\nabla f) \cdot \mathbf{v} \, d\mathbf{x} = \int_\Omega \nabla \cdot (f\mathbf{v}) \, d\mathbf{x} = \int_{\partial\Omega} f\mathbf{v} \cdot d\mathbf{S}$$

with the last term indicating the surface integral of $f\mathbf{v}$ over the boundary of Ω and $d\mathbf{S}$ the surface normal. Since the boundary term is zero, we infer

$$\langle f, \nabla \cdot \mathbf{v} \rangle = - \langle \nabla f, \mathbf{v} \rangle \tag{2.5}$$

which implies that the adjoint of the divergence operator is minus the the gradient operator.

Eq. (2.5) displays the property of skew (or anti) symmetry. A more general form of this property that is useful to the discretization process is the following. Let be given a real-valued operator (or tensor) $A : \mathcal{V} \to \mathcal{V}$. This operator is called skew-symmetric when

$$\langle \mathbf{u}, A\mathbf{v} \rangle = - \langle A\mathbf{u}, \mathbf{v} \rangle , \quad \forall \mathbf{u} , \mathbf{v} \in \mathcal{V} \tag{2.6}$$

As the inner product is symmetric, this implies $\langle \mathbf{u}, A\mathbf{u} \rangle = 0$ for any $\mathbf{u} \in \mathcal{V}$. The converse is also true, that is, if for a given operator A, we have $\langle \mathbf{u}, A\mathbf{u} \rangle = 0$, then this operator is skew-symmetric. The importance of the antisymmetry relations (2.5) and (2.6) will be discussed later in this section.

Below, we employ some relevant concepts of the Hamiltonian formalism that appear to be useful for the analysis of conservation properties. For an introducton, see e.g. [51]. In particular, the building blocks for a Hamiltonian formulation that might be most relevant here are a functional, a functional derivative, and a Poisson tensor.

A functional $\mathcal{F}$ is a mapping $\mathcal{F} : \mathcal{V} \to \mathbb{R}$, so that its arguments are field variables which, in turn, are functions of space and time, and it assigns a real number to them. An example of such a functional is integration of a function. Suppose $\mathbf{u} \in \mathcal{V}$, then we have, for instance,

$$\mathcal{F}(\mathbf{u}) = \int_\Omega F(\mathbf{x}, \mathbf{u}, \nabla \mathbf{u}) \, d\mathbf{x}$$

which yields a value of $\mathcal{F}$ depending on all the values taken by $\mathbf{u}$ on Ω, provided that the function F is real-valued. (Note that F is an ordinary function. Also note that $\nabla \mathbf{u}$ is

the derivative of $\mathbf{u}$ with respect to $\mathbf{x}$, which is the Jacobian matrix.) We use calligraphic capitals to denote functionals.

The functional (or variational) derivative of $\mathcal{F}$ with respect to $\mathbf{u}$, denoted $\delta\mathcal{F}/\delta\mathbf{u}$, is defined by

$$\lim_{\epsilon\to 0}\frac{\mathcal{F}\left(\mathbf{u}+\epsilon\mathbf{v}\right)-\mathcal{F}\left(\mathbf{u}\right)}{\epsilon}=\frac{d}{d\epsilon}\mathcal{F}\left(\mathbf{u}+\epsilon\mathbf{v}\right)\Big|_{\epsilon=0}=\langle\frac{\delta\mathcal{F}}{\delta\mathbf{u}},\mathbf{v}\rangle \tag{2.7}$$

Let us take the above example of the functional $\mathcal{F}(\mathbf{u})$. To compute its functional derivative it is assumed that F is continuously differentiable and $\mathbf{v}$ vanishes on the boundary of Ω. Upon substitution yields

$$\begin{aligned}\frac{d}{d\epsilon}\int_\Omega F\left(\mathbf{x},\mathbf{u}+\epsilon\mathbf{v},\nabla\mathbf{u}+\epsilon\nabla\mathbf{v}\right)\,d\mathbf{x}\,\Big|_{\epsilon=0} &= \int_\Omega\left(\frac{\partial F}{\partial\mathbf{u}}\cdot\mathbf{v}+\frac{\partial F}{\partial\nabla\mathbf{u}}\cdot\nabla\mathbf{v}\right)\,d\mathbf{x}\\ &= \langle\frac{\partial F}{\partial\mathbf{u}},\mathbf{v}\rangle+\langle\frac{\partial F}{\partial\nabla\mathbf{u}},\nabla\mathbf{v}\rangle\\ &\overset{2.5}{=} \langle\frac{\partial F}{\partial\mathbf{u}},\mathbf{v}\rangle-\langle\nabla\cdot\left(\frac{\partial F}{\partial\nabla\mathbf{u}}\right),\mathbf{v}\rangle\end{aligned}$$

so that the functional derivative is

$$\frac{\delta\mathcal{F}}{\delta\mathbf{u}}=\frac{\partial F}{\partial\mathbf{u}}-\nabla\cdot\left(\frac{\partial F}{\partial\nabla\mathbf{u}}\right)$$

Note that the above derivation can be generalized to higher order derivatives [41].

Let $\mathbf{p}\in\mathcal{V}$ be a state vector of (non-canonical) field variables describing an infinite-dimensional system. Then this system is said to be Hamiltonian if there exists a functional $\mathcal{H}(\mathbf{p})$ and a Poisson tensor J with certain properties such that the system is represented by

$$\frac{\partial\mathbf{p}}{\partial t}=J\frac{\delta\mathcal{H}}{\delta\mathbf{p}} \tag{2.8}$$

This formulation is called the symplectic form. Note that this is just one of the many equivalent ways of defining Hamiltonian both for canonical and non-canonical systems.

Let us elaborate further on the Hamiltonian description of Eqs. (2.1)−(2.2). We do this by expressing it in Cartesian tensor notation. First, we denote the momentum density by $\mathbf{m}=(m_x,m_y)^\mathsf{T}=(hu,hv)^\mathsf{T}$ and the mass flux by $\mathbf{q}=(q_x,q_y)^\mathsf{T}=(hu,hv)^\mathsf{T}$. For the current shallow-water system, a suitable Hamiltonian reads

$$\mathcal{H}\left(h,m_x,m_y\right)=\frac{1}{2}\int_\Omega\left[\frac{m_x^2+m_y^2}{h}+gh^2\right]\,dx\,dy$$

whose functional derivatives are

$$\frac{\delta\mathcal{H}}{\delta h}=\frac{1}{2}\left(-\frac{m_x^2+m_y^2}{h^2}+2gh\right)=-\frac{1}{2}\left(u^2+v^2\right)+gh\,,\quad\frac{\delta\mathcal{H}}{\delta m_x}=u\,,\quad\frac{\delta\mathcal{H}}{\delta m_y}=v$$

while the associated dynamics is controlled by the following Poisson tensor [13]

$$J = -\begin{bmatrix} 0 & \partial_x h & \partial_y h \\ h\,\partial_x & m_x\,\partial_x + \partial_x m_x & m_y\,\partial_x + \partial_y m_x \\ h\,\partial_y & m_x\,\partial_y + \partial_x m_y & m_y\,\partial_y + \partial_y m_y \end{bmatrix} \tag{2.9}$$

Like the Hamiltonian formulation, there are many known forms of the Poisson tensor. The current tensor is of the Lie-Poisson form which means that it (a) is linear in the state vector $(p_1, p_2, p_3)^\mathsf{T} \equiv (h, m_x, m_y)^\mathsf{T}$, (b) is skew-adjoint (or skew-symmetric), $J_{ij} = -J_{ji}$, and (c) satisfies the Jacobi condition [51, 13]

$$J_{il}\frac{\partial J_{jk}}{\partial p_l} + J_{jl}\frac{\partial J_{ki}}{\partial p_l} + J_{kl}\frac{\partial J_{ij}}{\partial p_l} = 0$$

for $i, j, k, l = 1, \ldots, 3$ (the Einstein convention is used). With the help of the antisymmetry relation (2.5), it can be verified that the above three conditions are indeed met by the tensor given by Eq. (2.9).

Now, if we use the components of the vector $(hu, hv)^\mathsf{T}$ instead of $(m_x, m_y)^\mathsf{T}$, then expanding the symplectic form in terms of the field variables h, $h\mathbf{u}$ and $\mathbf{q}$ results in

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \\ hv \end{bmatrix} = -\begin{bmatrix} 0 & \partial_x h & \partial_y h \\ h\,\partial_x & hu\,\partial_x + \partial_x uh & hv\,\partial_x + \partial_y uh \\ h\,\partial_y & hu\,\partial_y + \partial_x vh & hv\,\partial_y + \partial_y vh \end{bmatrix}\begin{bmatrix} gh - \frac{1}{2}(u^2+v^2) \\ u \\ v \end{bmatrix}$$

$$= \begin{bmatrix} -\partial_x q_x - \partial_y q_y \\ -gh\partial_x h - \partial_x(uq_x) - \partial_y(uq_y) \\ -gh\partial_y h - \partial_x(vq_x) - \partial_y(vq_y) \end{bmatrix}$$

which are indeed the shallow water equations (2.1)−(2.2), provided that the bed is uniform.

For our purposes, we want to show that the Hamiltonian is conserved at all times. To this end we consider a functional $\mathcal{F}(\mathbf{p})$ and examine variation of $\mathbf{p}$ to t, namely, $\delta\mathbf{p} = \mathbf{p}(\mathbf{x}, t+\delta t) - \mathbf{p}(\mathbf{x}, t)$, so that in the limit $\delta t \to 0$, we have $\delta\mathbf{p} = \delta t\,\partial\mathbf{p}/\partial t$. Recall Eq. (2.7), then one has

$$\lim_{\delta t\to 0}\frac{\mathcal{F}(\mathbf{p}+\delta\mathbf{p}) - \mathcal{F}(\mathbf{p})}{\delta t} = \boxed{\frac{d\mathcal{F}}{dt} = \langle\frac{\delta\mathcal{F}}{\delta\mathbf{p}}, \frac{\partial\mathbf{p}}{\partial t}\rangle}$$

which describes the time evolution of $\mathcal{F}$. Owing to Eq. (2.8) we observe that

$$\frac{d\mathcal{F}}{dt} = \langle\frac{\delta\mathcal{F}}{\delta\mathbf{p}}, J\frac{\delta\mathcal{H}}{\delta\mathbf{p}}\rangle$$

Since J is skew-symmetric we conclude that

$$\frac{d\mathcal{H}}{dt} = \langle\frac{\delta\mathcal{H}}{\delta\mathbf{p}}, J\frac{\delta\mathcal{H}}{\delta\mathbf{p}}\rangle = 0$$

implying the conservation of the Hamiltonian. This is basically a rendition of a special case of the classical Noether's theorem that relates the symmetry of a Hamiltonian system under translation in time to the conservation of energy.

Let us examine the time evolution of the total energy of the conservative shallow-water system in detail. We first discuss the contributions to the kinetic energy balance, followed by those of the gravitational potential energy. The total kinetic energy is

$$\mathcal{H}_{\text{kin}} = \frac{1}{2}\int_{\Omega} h\mathbf{u}\cdot\mathbf{u}\, d\mathbf{x} = \frac{1}{2}\langle \mathbf{u}, h\mathbf{u}\rangle$$

while its rate of change is given by

$$\frac{d\mathcal{H}_{\text{kin}}}{dt} = \langle \frac{\delta \mathcal{H}_{\text{kin}}}{\delta \mathbf{p}}, \frac{\partial \mathbf{p}}{\partial t}\rangle = \langle \frac{\delta \mathcal{H}_{\text{kin}}}{\delta h}, \frac{\partial h}{\partial t}\rangle + \langle \frac{\delta \mathcal{H}_{\text{kin}}}{\delta \mathbf{m}}, \frac{\partial \mathbf{m}}{\partial t}\rangle$$

Evaluating the functional derivatives as $\delta\mathcal{H}_{\text{kin}}/\delta h = -\frac{1}{2}\mathbf{u}\cdot\mathbf{u}$ and $\delta\mathcal{H}_{\text{kin}}/\delta\mathbf{m} = \mathbf{u}$ and substituting Eq. (2.2) into the above equation yield

$$\frac{d}{dt}\tfrac{1}{2}\langle \mathbf{u}, h\mathbf{u}\rangle = -\tfrac{1}{2}\langle \mathbf{u}, \mathbf{u}\frac{\partial h}{\partial t}\rangle - \langle \mathbf{u}, \nabla\cdot(\mathbf{q}\otimes\mathbf{u})\rangle - \langle \mathbf{q}, \nabla gh\rangle$$

where it is assumed that the bed is uniform. The last term converses kinetic energy into potential energy.

Next, the total gravitational potential energy reads

$$\mathcal{H}_{\text{pot}} = \frac{1}{2}\int_{\Omega} gh^2\, d\mathbf{x} = \frac{1}{2}g\langle h, h\rangle$$

The associated variational derivatives are then $\delta\mathcal{H}_{\text{pot}}/\delta h = gh$ and $\delta\mathcal{H}_{\text{pot}}/\delta\mathbf{m} = \mathbf{0}$. The rate of change of potential energy is determined by the following expression

$$\boxed{\frac{d}{dt}\tfrac{1}{2}g\langle h, h\rangle} = \langle \frac{\delta \mathcal{H}_{\text{pot}}}{\delta h}, \frac{\partial h}{\partial t}\rangle + \langle \frac{\delta \mathcal{H}_{\text{pot}}}{\delta \mathbf{m}}, \frac{\partial \mathbf{m}}{\partial t}\rangle = \boxed{-\langle gh, \nabla\cdot\mathbf{q}\rangle}$$

Finally, the total energy is given by

$$\mathcal{H} = \frac{1}{2}\langle \mathbf{u}, h\mathbf{u}\rangle + \frac{1}{2}g\langle h, h\rangle$$

The two contributions above can be combined into the equation of total energy as

$$0 = \frac{d\mathcal{H}}{dt} = -\tfrac{1}{2}\langle \mathbf{u}, \mathbf{u}\frac{\partial h}{\partial t}\rangle - \langle \mathbf{u}, \nabla\cdot(\mathbf{q}\otimes\mathbf{u})\rangle - \langle \nabla gh, \mathbf{q}\rangle - \langle gh, \nabla\cdot\mathbf{q}\rangle$$

By virtue of Eq. (2.5), the last two terms essentially cancel each other out, leaving only the first two terms while their sum must be zero. This result can be written as

$$\langle \mathbf{u}, \tfrac{1}{2}\frac{\partial h}{\partial t}\mathbf{u} + \nabla\cdot(\mathbf{q}\otimes\mathbf{u})\rangle = 0$$

Let us define the operator A with

$$A\mathbf{u} \coloneqq \nabla \cdot (\mathbf{q} \otimes \mathbf{u})$$

and denote I as the identity tensor. (Note that we may write $A = \mathbf{q} \cdot \nabla + (\nabla \cdot \mathbf{q})\, I$.) Now, the following must holds

$$\langle\, \mathbf{u}, \left[\, A + \tfrac{1}{2}\frac{\partial h}{\partial t} I \,\right] \mathbf{u} \,\rangle = 0$$

which implies that the operator

$$A + \tfrac{1}{2}\frac{\partial h}{\partial t} I$$

must be skew-symmetric. To accomplish this, the tensor A may be expressed as

$$A = \tfrac{1}{2}C - \tfrac{1}{2}C^{\mathsf{T}} - \tfrac{1}{2}\frac{\partial h}{\partial t} I \tag{2.10}$$

or, alternatively,

$$A = \tfrac{1}{2}C - \tfrac{1}{2}C^{\mathsf{T}} + \tfrac{1}{2}\,(\nabla \cdot \mathbf{q})\, I \tag{2.11}$$

so that C is a skew-symmetric tensor. (An arbitrary tensor C can be written as the sum of two parts, one symmetric, the other skew-symmetric: $C = \frac{1}{2}(C + C^{\mathsf{T}}) + \frac{1}{2}(C - C^{\mathsf{T}})$. If C is skew-symmetric, then the symmetric part is identically zero.)

We conclude this section by pointing out that an envisaged semi-discretization method should also possess a Hamiltonian structure not only to ensure its computational stability but also to respect the conservative cascade of energy from large to small scales through nonlinear interactions. This is particularly significant for describing nonlinear wave transformation in coastal regions. In this respect, some terms in the shallow water equations should also be individually energy conserving, namely, the pressure gradient term and the advection terms.

Conservation of energy by the pressure gradient term requires skew symmetry of the associated operator. More specifically, a discrete analogue of Eq. (2.5) is needed regarding the pressure gradient ∇gh and the divergence of mass flux $\nabla \cdot \mathbf{q}$. In Section 2.5, we will show how this desired mimetic property can be constructed by using the techniques from algebraic topology.

Additionally, skew symmetry should also be taken into account when discretizing the divergence term $\nabla \cdot (\mathbf{q} \otimes \mathbf{u})$ in the momentum equation (2.2), as indicated by Eq. (2.10). However, to include the shock formation as manifested in hydraulic jumps and tidal bores, some form of energy dissipation must be added. We will return to this matter in Chapter 3.

2.4 Differential forms and the Stokes' theorem

The purpose of this section is to present a brief introduction to some of the main concepts of differential geometry and to demonstrate their utility for the development of a numerical method for the solution of the shallow water equations on general meshes. These include

the differential forms, the exterior derivative and the generalized Stokes' theorem [4, 15, 43]. Their discrete counterparts will be elucidated in detail in Section 2.5 which form the starting point for the mimetic discretizations of Chapter 3, 4 and 5.

The equations presented in the previous sections are expressed in terms of vector calculus. The fundamental attributes are the scalar and vector fields. A field variable is a local function that describe the variable at each point in space (and at each instant in time, but we will not consider that here; see, e.g. [63]). This is also called a density and is essentially the result of a limit process. For example, mass density, denoted as $\rho(\mathbf{x}, t)$, is the result of the ratio of an infinitesimally small mass δm to an infinitesimally small volume δV while taking the limit $\delta V \to 0$. Obviously, such a scalar field does not make sense physically, since a zero volume would contain no mass. It is a pure mathematical concept resulted from the process of limit.

By contrast, differential forms are defined informally as physical variables that are associated with a geometrically *finite* object, such as a curve, plane or volume. For example, we can express mass as

$$m = \int_V \rho \, dV$$

which has a clear physical meaning irrespective of the size and shape of volume V. So, here mass is defined as a volume integral and the quantity $\rho \, dV$ is called a differential form.

Another example is the mass flux density which is defined as

$$\lim_{\delta S \to 0} \frac{\dot{m}}{\delta S}$$

with $\dot{m} = dm/dt$ the mass flow rate and δS the infinitesimal area through which the mass flows. This mathematically well-defined quantity is by itself physically meaningless: it only provides a local measure of the mass current per unit cross area emanating from a *point* in space.

Another view is that the mass flux density is a vector field, denoted $\mathbf{q} = \rho \mathbf{u}$ where $\mathbf{u}$ is the flow velocity vector, and the integral of this flux over a cross section S yields the total amount of mass passing through the cross section in a unit of time. The surface integral is expressed as

$$\int_S \mathbf{q} \cdot d\mathbf{S}$$

where $d\mathbf{S}$ is the surface element pointing outward normal to the surface. Here, $\mathbf{q} \cdot d\mathbf{S}$ represents the mass flux and is physically defined for any size and shape of section S. This physical quantity is another example of a differential form. Note, however, that by Gauss' divergence theorem in vector calculus, one has

$$\int_V \nabla \cdot \mathbf{q} \, dV = \oint_{\partial V} \mathbf{q} \cdot d\mathbf{S}$$

which provides a geometric interpretation of the divergence operator $\nabla \cdot$ in the sense that integrating this operator over a *finite* volume yields the total flux through the volume boundaries.

Quantity $\rho\mathbf{u}$ can also be interpreted as the mass circulation density and is identified with the curl of $\rho\mathbf{u}$ at a single point: $\nabla \times \rho\mathbf{u}$. Indeed, by Stokes' curl theorem, if A is a finite surface in $\mathbb{R}^2$ and $d\mathbf{l}$ is a curve element locally tangent to the boundary of A, then the total circulation of the vector $\rho\mathbf{u}$ around the perimeter of A is computed as

$$\oint_{\partial A} \rho\mathbf{u} \cdot d\mathbf{l} = \int_A (\nabla \times \rho\mathbf{u}) \cdot d\mathbf{A}$$

Thus, mass circulation is symbolized by the differential form $\rho\mathbf{u} \cdot d\mathbf{l}$ integrated on a *finite* line segment.

There are also, however, quantities that can be sampled at single locations, such as surface elevation, bed level and dynamic pressure. These are differential forms associated to a spatial point. Such forms commonly manifest themselves as the argument of the gradient operator ∇. This is clarified by the fundamental theorem of calculus for line integrals. Let $\pi : \mathbb{R}^2 \to \mathbb{R}$ be a differentiable function given on a continuous curve $\ell \subset \mathbb{R}^2$ that starts at point p and ends at point q. Then the integral of the gradient of π over the curve ℓ is equal to the total change in π between the two endpoints of ℓ, that is,

$$\int_\ell \nabla\pi \cdot d\mathbf{l} = \pi(q) - \pi(p)$$

Differential forms are characterized by the dimension of the underlying geometric objects. A differential $k-$form integrates over a $k-$dimensional smooth (infinitely differentiable) manifold embedded in a $n-$dimensional space ($k = 0, 1, \ldots, n$), and takes this to $\mathbb{R}$. For instance, in $\mathbb{R}^3$ there are four types of differential forms, that is, $0-$, $1-$, $2-$ and $3-$forms, associated with points, curves, planes and volumes, respectively. It is important to note that unlike scalar and vector fields, differential forms are independent of coordinate systems and metric (e.g., length, area, angle).

In what follows, forms are denoted by lower case Greek letters with the superscript indicating the dimension. Hence, referring to the first example above, $\mu^{(3)} = \rho\, dV$ is a $3-$form which is a scalar. In the case of the flow velocity vector, there are two distinct differential forms, namely, the $2-$form $\phi^{(2)} = \mathbf{q} \cdot d\mathbf{S}$, that is, the normal to a plane, and the $1-$form $\gamma^{(1)} = \rho\mathbf{u} \cdot d\mathbf{l}$, which is the vector tangent to a curve. And in the last example, function π is a $0-$form, $\pi^{(0)} = \pi$, which trivially gives a scalar.

The calculus of differential forms is based on the exterior derivative and the generalized Stokes' theorem which extend the notion of differentation and integration, respectively, to arbitrary dimensions. Let d denotes the exterior derivative and let $\alpha^{(k)}$ be a $k-$form defined on some manifold $\mathcal{M}$ of dimension k. The exterior derivative of $\alpha^{(k)}$ is a $(k+1)-$form that is written as $\mathrm{d}\alpha^{(k)}$, for $k = 0, 1, \ldots, n-1$. Indeed, the action of exterior derivative on differential forms provides us a coordinate invariant way to calculate the gradient, curl and divergence operators of vector calculus. For example, $\mathrm{d}\alpha^{(0)}$ is the same as the gradient of a scalar and the result is a $1-$form, which represents a tangential component of a vector. Likewise, $\mathrm{d}\alpha^{(1)}$ and $\mathrm{d}\alpha^{(2)}$ are equivalent to the curl of a vector (tangent to a curve) and the divergence of a vector (normal to a plane), respectively. The result of the former is a

2−form, which is actually the normal component of a vector, while the result of the latter is scalar, a 3−form.

As we have seen in the above examples, the gradient, curl and divergence operators can be linked to the corresponding geometric objects (curve, plane and volume, respectively) with lower-dimensional boundaries (points, curves and planes, respectively) by means of the corresponding integral theorems (the fundamental theorem of calculus for line integrals, the Stokes' curl theorem and the Gauss' divergence theorem, respectively). In the same vein, the exterior derivative can be connected to a $(k+1)-$dimensional manifold $\mathcal{M}$ with $k-$dimensional boundary $\partial\mathcal{M}$ through the generalized Stokes' theorem, which is stated in the following very elegant and simple formula

$$\int_{\mathcal{M}} \mathrm{d}\alpha^{(k)} = \int_{\partial\mathcal{M}} \alpha^{(k)}$$

for a given $k-$form $\alpha^{(k)}$. This theorem equates the integral of the exterior derivative of a form on a manifold to the integral of this form on the boundary of the manifold. We observe that the three key theorems of vector calculus as outlined above are all special cases of the generalized Stokes' theorem.

Differential forms are the essential building blocks in the study of differential geometry [43]. This mathematical language allows one to express differential forms on smooth and curved manifolds in a consistent manner, not dependent on a coordinate system. But most relevant to our discussion is that the use of differential forms is motivated by the physical fact that the measurements of physical quantities, e.g., mass, mass circulation, mass flux, pressure, are typically linked to integration over geometrically finite manifolds. As such, differential forms naturally lend themselves to a discrete representation. In particular, different global variables can be represented as coordinate-free discrete differential forms integrated on different mesh elements (vertices, edges, faces and cells). This is the approach that we will use in the discrete setting.

Consider a three-dimensional computational mesh which consists of vertices, edges, faces and volumes. Let a vertex, an edge, a face and a cell be denoted by $\sigma_{(k)}$ with $k = 0, 1, 2, 3$, respectively. A discrete $k-$form is defined as the integral of a $k-$form over a k-dimensional mesh element $\sigma_{(k)}$, symbolized by

$$\int_{\sigma_{(k)}} \alpha^{(k)}$$

and yields a single real number associated with $\sigma_{(k)}$. Note that the discrete form is the *whole* integral quantity, not the integrand as with differential forms. In this way, we distinguish between 0-forms represented by their values at a set of vertices, 1-forms by their line integrals over a set of edges, 2-forms by their surface integrals over a set of faces, and 3-forms by their volume integrals over a set of cells. We use from now on Greek letters for discrete forms, that is, $\pi^{(0)}$ for the pressure at a vertex, $\gamma^{(1)}$ for the mass circulation along a mesh edge, $\phi^{(2)}$ for the mass flux through a mesh face, $\mu^{(3)}$ for the mass in a cell, etc.

It is also apparent that the discrete forms serve as the degrees of freedom for our numerical framework, rather than grid point values as used in finite difference methods

or cell averages in finite volume methods. Since these forms are topological, that is, independent of metric, the intended numerical method can easily be extended to any type of computational meshes embedded in two-dimensional Euclidean spaces, including rectilinear meshes (Chapter 3), curvilinear meshes (Chapter 4) and triangular meshes (Chapter 5).

The generalized Stokes' theorem reveals that differential forms and integral theorems are intimately connected. More specifically, the integration of a differential form, or the discrete form for that matter, can be used to establish any of the differential operators such as gradient, curl or divergence. This is illustrated by Figure 2.1 that shows the three fundamental theorems with which the integral of the corresponding differential operator

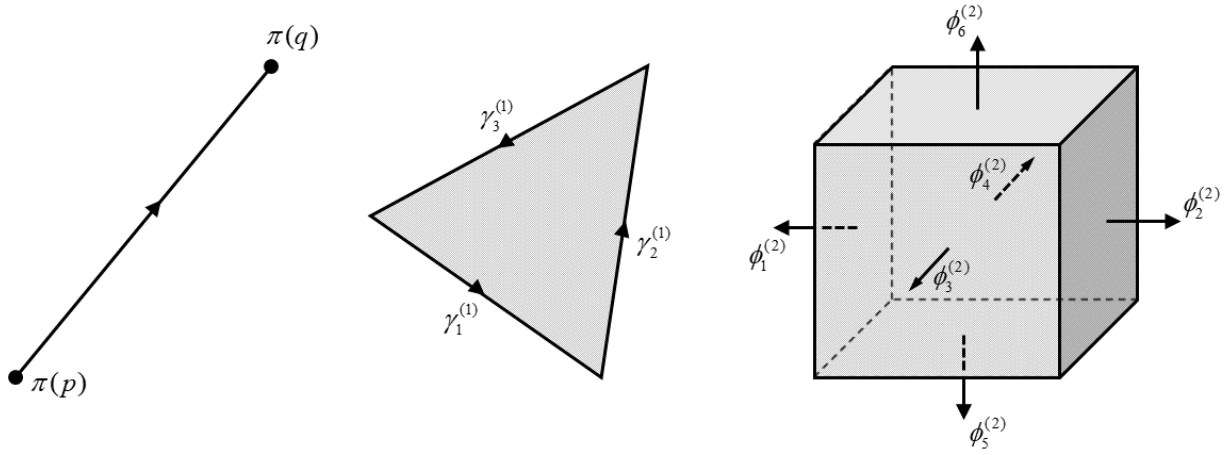


Figure 2.1: The evaluation of the gradient operator along a line segment by means of the discrete $0-$forms at the two endpoints as per the fundamental theorem of calculus for line integrals (left), the curl operator in a triangle by summing the discrete $1-$forms over the triangle edges using the Stokes' curl theorem (center), and the computation of the divergence in a cube is the same as taking the sum of the discrete $2-$forms on the six faces according to the Gauss' divergence theorem (right). The displayed arrows indicate the orientation of the geometric object; see Section 2.5.2 for further discussion.

over a finite geometric object is computed by a direct evaluation of the associated discrete form over the boundary of that object. This observation is the central theme for the mimetic discretizations used in this work. In the next section, we will apply the Stokes' theorem to construct a discrete counterpart of the exterior derivative, with which the continuous differential operators, `grad`, `curl` and `div` can then be mimicked at the discrete level.

2.5 Basic concepts of algebraic topology

2.5.1 Introduction

This section concerns with some of the essential definitions and tools of algebraic topology for two-dimensional manifolds. They establish a formalization of the notion of discretization of physical space in which *physical laws* are embedded. This formalism lay the foundation of the numerical framework of SWASH in the next sections.

Algebraic topology is a branch of mathematics that essentially deals with the study of a manifold (a geometric object) which is encoded by means of the (graph) connectivity. In turn, algebra and discrete boundary relations determined by this connectivity are employed to find topological invariants and symmetries of the manifold implied by differential geometry. As we will see later on, it defines a clean separation between the process of *exact discretization* of physical conservation properties and the process of approximation of constitutive relations that should be implemented anyway. Original ideas about this approach were proposed two decades ago by [35, 54, 36, 63].

A good introduction to algebraic topology is provided by [42]. Somewhat more abstract is the book of [19].

2.5.2 Cell complex and orientation

A manifold is a topological space living in $n-$dimensional Euclidean space $\mathbb{R}^n$ that is equipped with a topological structure to allow defining mappings of (sub)manifolds, but not measured by a metric. Such a structure refers to the essential relationships that describe the connectivity between geometric objects and the integral relations that underlie certain invariant and symmetry properties.

A finite dimensional manifold that we will consider here is a computational mesh. It provides a means of partitioning a computational domain $\Omega \subset \mathbb{R}^n$ into a collection of distinct geometric objects (or submanifolds) called $k-$cells with $k = 0, 1, \ldots, n$ indicating their spatial dimension. The associated mesh is thus discretely represented by a finite collection of vertices ($0-$cells), edges ($1-$cells), faces ($2-$cells) and cells ($3-$cells).

A $k-$cell is denoted by $\sigma_{(k)}$ and its size or (intrinsic) volume is denoted $|\sigma_{(k)}|$. We define $|\sigma_{(0)}| = 1$. The collection of $k-$cells is a subset of $\mathbb{R}^n$, denoted $\mathcal{M}_k$, and is called a $k-$dimensional manifold. This manifold is assumed to have a boundary. The boundary of a $k-$cell, denoted $\partial\sigma_{(k)}$, is made up of $(k-1)-$cells that are directly connected to. These lower dimensional cells are elements of $\mathcal{M}_{k-1}$ $(k = 1, \ldots, n)$ and are referred to as the faces of the $k-$cell. Note that the boundary of a $0-$cell is empty. The set $\{\mathcal{M}_0, \ldots, \mathcal{M}_n\}$ is called the mesh.

A cell complex K on Ω is a finite set of $k-$cells, with $k = 0, 1, \ldots, n$, such that (i) the $n-$cells cover Ω, (ii) each face of a $k-$cell of K is in K, and (iii) the intersection of any two $k-$cells of K is either a face of each of them or is empty. We simply write the cell complex as $K = \{\mathcal{M}_0, \ldots, \mathcal{M}_n\}$, which is a mesh. Note that the converse is not necessarily true (see below). Figure 2.2 illustrates an example of a cell complex in a two-dimensional domain.

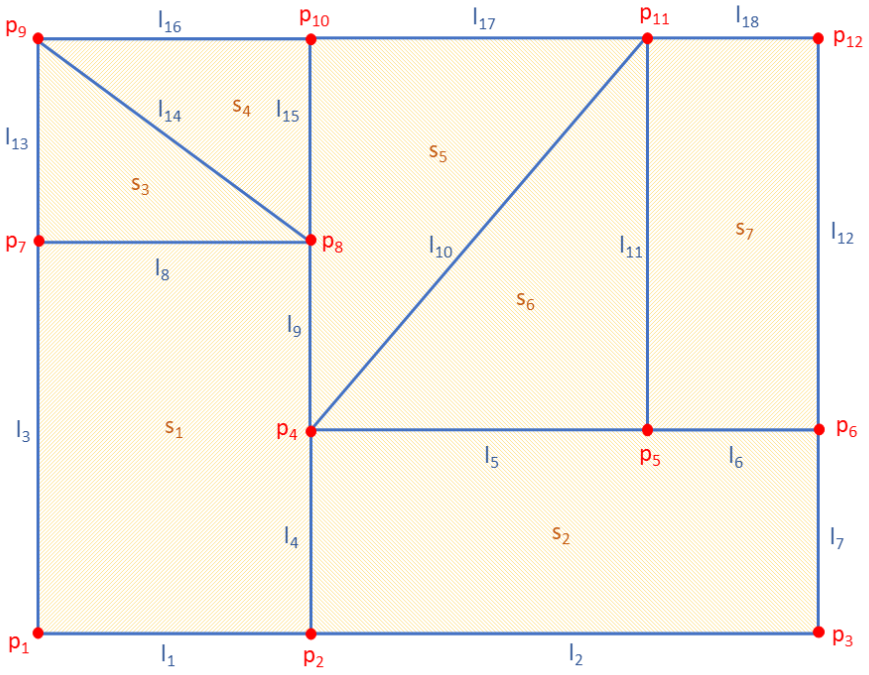


Figure 2.2: Example of a two-dimensional cell complex with labeled 0−cells (vertices), 1−cells (edges) and 2−cells (faces). The 2−cells, i.e. computational cells, are a mixed of triangles and rectangles.

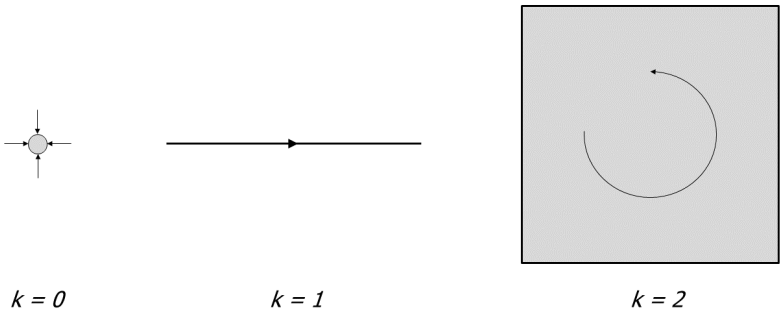
Manifolds are also endowed with an *orientation* which is a key element for identifying the conservation properties in the construction of mimetic discretizations. Two types of orientation can be distinguished for a manifold $\mathcal{M}_k$: inner and outer orientation. The first type defines the orientation *in* the geometric object, while the second designates the orientation *outside* the geometric object embedded in space $\mathbb{R}^n$.

Every $k-$cell is oriented and has exactly two directions. In this work, we choose a positive orientation according to the right-hand rule. Consequently, the other is negative. In particular, an inner-oriented 0−cell is positively oriented as a sink (*into* the vertex), an inner-oriented 1−cell is oriented by a direction, pointing to the right, *along* the edge, an inner-oriented 2−cell by a sense of rotation, in the counterclockwise direction, *on* its face and an inner-oriented 3−cell by a right-handed screw *inside* its cell. Additionally, the inner orientation on $\partial\sigma_{(k)}$ is induced by $\sigma_{(k)}$. It is important to note that the inner orientation on a $k-$cell is identical for each such cell embedded in an $n-$dimensional space.

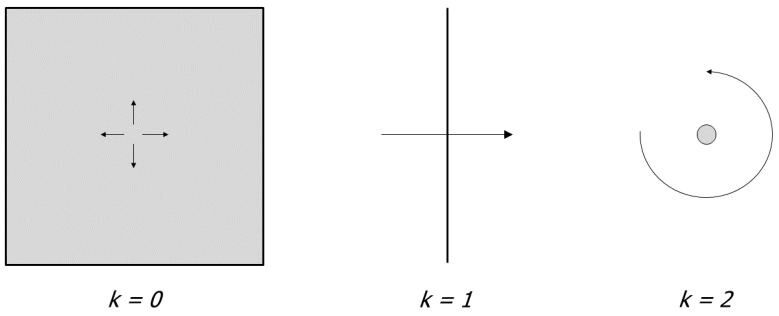
Outer orientation, on the other hand, depends on the dimension of the embedding space. The outer orientation specifies, for instance, a transverse direction *through* a vertex embedded in $\mathbb{R}^1$, *across* an edge in $\mathbb{R}^2$ and *through* a face for a 3D embedding space. Here, a positive orientation is the one implied by the orientation of the embedded space that is equipped with a right-handed coordinate system in $\mathbb{R}^n$. Another example is a

counterclockwise rotation *around* a vertex or an edge embedded in $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively. Finally, the outer orientation of an $n-$cell in $\mathbb{R}^n$ is induced by the outer orientation of its faces with outward normals. Thus, the same geometric object has different types of outer orientation depending on the dimension of embedding space $\mathbb{R}^n$.

The concept of inner and outer orientation gives rise to a pair of meshes embedded in $\mathbb{R}^n$, each endowed with a different type of orientation. Moreover, they are topologically dual to each other in the sense that an inner-oriented $k-$cell corresponds to an outer-oriented $(n-k)-$cell, and vice versa. The former is referred to as the primal mesh, denoted K, the latter is called the dual mesh, denoted $\tilde{K}$. We will use the tilde throughout this chapter to indicate a dual object. Here, K is inner oriented while $\tilde{K}$ is outer oriented, but this is merely a choice and either choice is equally fine. What is important is that all of the $k-$cells in one particular mesh must have the same type of orientation (i.e. inner *or* outer). Figure 2.3 depicts a graphical representation of the orientation of the various primal and dual cells in a 2D space.



(a) $k-$cells with inner orientation.



(b) $(2-k)-$cells with outer orientation.

Figure 2.3: Oriented primal cells (a) and dual cells (b) embedded in $\mathbb{R}^2$.

The computational mesh is an oriented cell complex K that covers the domain Ω. This mesh is designated as the primal mesh. We denote by $\tilde{K}$ its associated dual mesh. However, not all faces of the $(n-k)-$cells in $\tilde{K}$ (for $k = 1, \ldots, n$) are contained in $\tilde{K}$. Nevertheless,

as we will see later, the dual mesh is not required to be a cell complex in our discretization method. Also, it does not need to be created or stored explicitly − only its metric will be computed. This will be elaborated in detail in Section 2.5.7.

2.5.3 The computational mesh and its dual

The topology of the computational mesh is routinely described by means of simplices (e.g., triangles, tetrahedrons) or cuboids (e.g., quadrilaterals, hexahedrons). One should note, however, that both descriptions, though topologically equivalent, are geometrically different; see Section 2.5.7. The present work is entirely devoted to polygonal meshes in $(x, y) \in \Omega \subset \mathbb{R}^2$ even though the space dimension n is kept general in the present exposition. Here, mesh edges are straight lines and mesh faces are planar. Note that, although there is no difference between the edge and the face of a 2D mesh, their distinction will nevertheless clarify the derivations to be presented.

A polygonal mesh consists of a finite number of polygons. A polygon is said to be cyclic if it can be inscribed in a circle, that is, if there exists a circle so that every vertex of the polygon lies on the circle. This circle is called the circumcircle. For example, all triangles and all rectangles are cyclic. The center of the circumcircle is known as the circumcenter and can be found as the intersection of the perpendicular bisectors of the edges of the polygon.

A polygon is *well centered* if its circumcenter is contained in its interior. A well-centered computational mesh has all of its polygonal cells that are well centered. For example, an acute triangulation is well centered. The mesh constructed by joining the primal cell circumcenters is called the circumcentric dual mesh. Any well-centered primal mesh and its dual are mutually orthogonal. A classic example is the Delaunay triangulation (primal mesh) and the associated Voronoi tessellation (dual mesh).

Rectangular and curvilinear meshes are always well centered. This is not necessarily true for triangular cells. In particular, the circumcenter of a right-angled triangle lies at the midpoint of the hypotenuse and the circumcenter of an obtuse triangle lies outside the triangle. We will come back to this issue later on.

An alternative would be the use of the barycentric dual mesh. This mesh is formed by connecting the cell centroids and the edge midpoints. The barycentric dual mesh greatly facilitates flexibility in mesh generation and also in adaptive mesh refinement. However, the lack of orthogonality between the primal edges and their barycentric duals generally increases the complexity of the discretization [21, 37, 38, 39] and may additionally affect the numerical stability.

As will be demonstrated in Section 2.5.7, the circumcentric dual mesh is the preferred one as it allows for computationally tractable and stable discretizations. For the present SWASH applications, only orthogonal rectangular grids (Chapter 3), orthogonal curvilinear grids (Chapter 4) and Delaunay triangular meshes (Chapter 5) are considered.

2.5.4 Chains and boundary operator

Let $C_k(K)$ be a group generated by a basis consisting of all the $k-$cells of cell complex K. An element of $C_k(K)$ is called a $k-$chain and is a linear combination of oriented $k-$cells,

$$c_{(k)} = \sum_i c^i \, \sigma_{(k),i}$$

where $\sigma_{(k),i}$ is the ith $k-$cell in $\mathcal{M}_k$ and $c^i \in \{-1, 0, 1\}$ is the ith component of $c_{(k)}$. The $k-$cells form the canonical basis for the vector space of $k-$chains. The dimension of $C_k(K)$ equals the number of elements of $\mathcal{M}_k$ and is written as $|C_k|$. A $k-$chain $c_{(k)}$ is represented as a row vector of length $|C_k|$. Furthermore, integer component c^i of $c_{(k)}$ refers to the orientation of the cell in the chain with respect to its default orientation in cell complex K (positive if they agree or negative if they disagree) or to the cell not being a part of the chain, that is, $c^i = 0$. Note that a $k-$cell is also named as an elementary $k-$chain.

Just like the boundary of a $k-$cell is an element of $\mathcal{M}_{k-1}$, so is the boundary of a $k-$chain, denoted $\partial c_{(k)}$, an element of $C_{k-1}(K)$. In this regard, we define the boundary operator as a linear operator $\partial_k : C_k(K) \to C_{k-1}(K)$ which returns a $(k-1)-$chain after applying to the $k-$chain,

$$\partial_k c_{(k)} = \partial_k \sum_i c^i \, \sigma_{(k),i} \coloneqq \sum_i c^i \, \partial_k \sigma_{(k),i}$$

with $\partial_k \sigma_{(k),i}$ the boundary of $\sigma_{(k),i}$ which is a $(k-1)-$cell formed by the faces of the oriented $k-$cell, as follows

$$\partial_k \sigma_{(k),i} = \sum_j o_{i,j} \, \sigma_{(k-1),j} \tag{2.12}$$

where $o_{i,j}$ equals $+1$ if $\sigma_{(k-1),j} \in \mathcal{M}_{k-1}$ and the orientations of $\sigma_{(k-1),j}$ and $\sigma_{(k),i}$ agree, -1 if these orientations disagree, or 0 if $\sigma_{(k-1),j}$ is not a face of $\sigma_{(k),i}$. Hence,

$$\partial_k c_{(k)} = \sum_i \sum_j c^i \, o_{i,j} \, \sigma_{(k-1),j} \tag{2.13}$$

The boundary operator has the important property that the boundary of a boundary is empty, so that using the boundary operator twice to any $k-$chain gives a null value, that is, $\partial_{k-1} \partial_k c_{(k)} = 0 \,, \; \forall c_{(k)} \in C_k(K)$. The boundary operator is called nilpotent.

Given a basis for the vector spaces $C_k(K)$ and $C_{k-1}(K)$, the boundary operator is represented as an incidence matrix $\mathbb{D}_k$ of size $|C_k| \times |C_{k-1}|$. Each row corresponds to each element of $C_k(K)$ and each column to each element of $C_{k-1}(K)$. Owing to Eq. (2.12), the entries of the matrix are given by

$$[\mathbb{D}_k]_{i,j} = o_{i,j} \,, \quad k = 1, \ldots, n \tag{2.14}$$

An entry is -1 or $+1$ (the sign depending on the orientation) if an element of $C_{k-1}(K)$ is incident with an element of $C_k(K)$, or 0 if they are not related. Thus the action of boundary operator ∂_k on a chain $c_{(k)}$ amounts to the matrix-vector multiplication $c_{(k)} \, \mathbb{D}_k$, which is a row vector of length $|C_{k-1}|$. We note that $c_{(k)} \, \mathbb{D}_k \, \mathbb{D}_{k-1} = \mathbf{0}^\mathsf{T} \,, \; \forall c_{(k)} \in C_{(k)}(K) \,, \; k = 2, \ldots, n$.

2.5.5 Cochains and coboundary operator

The vector space of chains $C_k(K)$ of cell complex K coexists with a dual vector space of linear functions $\gamma^{(k)} : C_k(K) \to \mathbb{R}$. This dual space is denoted by $C^k(K)$ and its elements are called $k-$cochains. Let $c_{(k)}$ be the $k-$chain of K and $\gamma^{(k)}$ the $k-$cochain of K. We write

$$\langle c_{(k)}, \gamma^{(k)} \rangle := \gamma^{(k)} \left(c_{(k)} \right) \in \mathbb{R}$$

for the value of $\gamma^{(k)}$ on $c_{(k)}$. This linear mapping is called the *duality pairing* of $k-$cochain with $k-$chain. As it will be clear shortly, the notion of duality pairing between cochains and chains plays a centrol role in the discretization process.

Given a basis of $C_k(K)$, $\{\sigma_{(k),i} \,|\, i = 1, \ldots, |C_k|\}$, there is a dual basis of $C^k(K)$, $\{\sigma^{(k),i} \,|\, i = 1, \ldots, |C_k|\}$, such that

$$\langle \sigma_{(k),j}, \sigma^{(k),i} \rangle = \delta^i_j$$

so that the ith elementary $k-$cochain is associated with the ith $k-$cell only. By linearity, a $k-$cochain $\gamma^{(k)} \in C^k(K)$ can be expressed as

$$\gamma^{(k)} = \sum_i \gamma_i \sigma^{(k),i}$$

and is represented as a column vector with its components $\gamma_i \in \mathbb{R}$. The length of this vector is $|C_k|$. Duality pairing of a $k-$cochain $\gamma^{(k)}$ with a $k-$chain $c_{(k)}$ can now be calculated as

$$\langle c_{(k)}, \gamma^{(k)} \rangle = \sum_i \sum_j c^j \gamma_i \langle \sigma_{(k),j}, \sigma^{(k),i} \rangle = \sum_i \sum_j c^j \gamma_i \delta^i_j = \sum_i c^i \gamma_i = c_{(k)}\, \gamma^{(k)}$$

This duality pairing is a metric-free operation. (This is not the inner product of Section 2.3 because $c_{(k)}$ and $\gamma^{(k)}$ are not defined in one single vector space.)

A $k-$cochain acts as a function that associates to every $k-$chain of K a discrete real number that is independent of any coordinate system. In particular, the value

$$\langle \sigma_{(k),i}, \gamma^{(k)} \rangle = \gamma_i$$

is a coordinate-free scalar evaluated on the ith $k-$cell. This function evaluation is interpreted as the geometric integration of $\gamma^{(k)}$ over the $k-$cell of cell complex K. The integral of $\gamma^{(k)}$ over an inner-oriented $k-$cell is symbolized by

$$\int_{\sigma_{(k)}} \gamma^{(k)} := \langle \sigma_{(k)}, \gamma^{(k)} \rangle$$

and is referred to as an inner-oriented $k-$form, or inner $k-$form for short. By linearity, the integral of $\gamma^{(k)}$ over a $k-$chain can be calculated as

$$\int_{c_{(k)}} \gamma^{(k)} = \sum_i c^i \int_{\sigma_{(k),i}} \gamma^{(k)} = \sum_i c^i \langle \sigma_{(k),i}, \gamma^{(k)} \rangle = \sum_i c^i \gamma_i = \langle c_{(k)}, \gamma^{(k)} \rangle$$

Let $c_{(k+1)}$ be a $(k+1)-$chain of cell complex K, then its boundary $\partial_{k+1}c_{(k+1)}$ is a $k-$chain. Here, the orientation on $\partial_{k+1}c_{(k+1)}$ is induced by $c_{(k+1)}$. The adjoint of this boundary operator with respect to the duality pairing $\langle\cdot,\cdot\rangle$ is called the coboundary operator δ^k and is defined by

$$\langle c_{(k+1)}, \delta^k\gamma^{(k)}\rangle = \langle \partial_{k+1}c_{(k+1)}, \gamma^{(k)}\rangle\,, \quad \forall c_{(k+1)} \in C_{k+1}(K)\,,\, \forall \gamma^{(k)} \in C^k(K)$$

The coboundary operator $\delta^k : C^k(K) \to C^{k+1}(K)$ is a linear operator that relates a $k-$cochain to a $(k+1)-$cochain. The above adjoint relation can also be written in the following integral form

$$\int_{c_{(k+1)}} \delta^k\gamma^{(k)} = \int_{\partial_{k+1}c_{(k+1)}} \gamma^{(k)}$$

which can be viewed as the discrete counterpart of the generalized Stokes' theorem. The operator δ^k is also called the kth discrete exterior derivative. Recall that the three theorems of vector calculus, namely, the fundamental theorem of calculus for line integrals, the Stokes' curl theorem and the Gauss's divergence theorem (cf. Figure 2.1) are a special case of the generalized Stokes' theorem applied to oriented $0-$forms, $1-$forms and $2-$forms, respectively, in $\mathbb{R}^3$. This observation is key in constructing the mimetic discretization of continuous differential operators `grad`, `curl` and `div`.

The operator δ^k can be expressed as an incidence matrix $\mathbb{D}^k$ of size $|C_{k+1}| \times |C_k|$ so that the action of $\mathbb{D}^k$ on a cochain $\gamma^{(k)}$ is the matrix-vector multiplication $\mathbb{D}^k\gamma^{(k)}$. Matrix $\mathbb{D}^k$ is the adjoint of matrix $\mathbb{D}_{k+1}$, which can be demonstrated by the above theorem, namely,

$$c_{(k+1)}\mathbb{D}^k\gamma^{(k)} = c_{(k+1)}\mathbb{D}_{k+1}\gamma^{(k)}$$

which implies $\mathbb{D}^k = \mathbb{D}_{k+1}$ for $k = 0, \ldots, n-1$.

By virtue of the duality pairing between the boundary operator and coboundary operator, the discrete exterior derivative is a metric independent operator, which additionally has the property that

$$\mathbb{D}^{k+1}\,\mathbb{D}^k = \mathbf{0}^{\mathsf{T}}\,,\, \forall k = 0, \ldots, n-2$$

reflecting the nilpotency of the coboundary operator, that is, $\delta^{(k+1)}\,\delta^{(k)} = 0$.

Let us choose an inner orientation for cell complex K in $\mathbb{R}^2$. Then the discrete exterior derivative represents an exact discretization of `grad` if $k = 0$ and `curl` if $k = 1$ such that the Stokes' theorem holds for the associated *inner-oriented* $k-$cells (cf. left and center panels of Figure 2.1, respectively). In addition, we have that $\mathbb{D}^1\mathbb{D}^0 = \mathbf{0}^{\mathsf{T}}$, which implies `curl grad` $= 0$ (the curl of a gradient is zero). Exact representation of this vector calculus identity is crucial for a physics-compatible and stable numerical scheme.

2.5.6 The dual mesh: discrete $k-$forms and exterior derivative

The notions of chains, the boundary operator, discrete forms and the discrete exterior derivative can also be applied on the dual mesh. Let K be a primal mesh (or cell complex) and $\tilde{K}$ the associated dual mesh on $\Omega \subset \mathbb{R}^2$. Remember that the dual mesh is not a cell

complex, but we omit this feature here for simplicity as it does not change the exposition in the following.

There exists a bijective map between the dual mesh elements and the primal ones, namely, each $(n-k)-$cell of the dual mesh is dual to a primal $k-$cell, for $k=0,\dots,n$. We denote the dual of $k-$cell by $\tilde{\sigma}_{(n-k)}$ and the map by $\star$, that is, $\star\sigma_{(k)}=\tilde{\sigma}_{(n-k)}$. The set of dual cells is denoted by $\tilde{\mathcal{M}}_{n-k}$. The dual mesh is then given by $\tilde{K}=\{\tilde{\mathcal{M}}_n,\dots,\tilde{\mathcal{M}}_0\}$. In line with the choice of primal mesh K in Section 2.5.2, $\tilde{K}$ is outer oriented. Recall that the outer orientation of a dual cell depends on the dimension of the embedded space $\mathbb{R}^n$.

We denote the vector space of dual $k-$chains by $C_{n-k}(\tilde{K})$ and its canonical basis by $\{\tilde{\sigma}_{(n-k),i}\,|\,i=1,\dots,|C_k|\}$. Similarly, we have the space of dual $k-$cochains, denoted $C^{n-k}(\tilde{K})$, generated by its dual basis $\{\tilde{\sigma}^{(n-k),i}\,|\,i=1,\dots,|C_k|\}$. The elements of $C^{n-k}(\tilde{K})$ are the dual discrete forms by which an outer $(n-k)-$form is dual to an inner $k-$form.

The boundary of an outer-oriented cell $\tilde{\sigma}_{(n-k+1)}$ of the dual mesh, with $k=1,\dots,n$, constitutes a number of connected faces $\tilde{\sigma}_{(n-k)}$, for which not all of them are in the mesh. By duality, we have the dual boundary operator $\tilde{\partial}_{n-k+1}:C_{n-k+1}(\tilde{K})\to C_{n-k}(\tilde{K})$ obtained as follows

$$\tilde{\partial}_{n-k+1}\tilde{\sigma}_{(n-k+1),i}=\sum_j \tilde{o}_{i,j}\,\tilde{\sigma}_{(n-k),j}\,,\quad i=1,\dots,|C_{k-1}|\,,\;\; j=1,\dots,|C_k|$$

with $\tilde{o}_{i,j}$ indicating the outer orientation of $\tilde{\sigma}_{(n-k)}$ induced by $\tilde{\sigma}_{(n-k+1)}$ ($+1$ if they agree, -1 if they disagree, and 0 otherwise). This orientation coefficient is related to that on the primal mesh, viz. Eq. (2.12), according to

$$\tilde{o}_{i,j}=(-1)^k\,o_{j,i}\,,\quad \forall k=1,\dots,n \tag{2.15}$$

Here the sign coefficient follows from Figure 2.3. In particular, the orientation of the inner-oriented $0-$cell is the opposite of the orientation of the outer-oriented $2-$cell while those of the $1-$cells have the same orientation.

The coefficients $\tilde{o}_{i,j}$ constitute an incidence matrix $\tilde{\mathbb{D}}_{n-k+1}$ of size $|C_{k-1}|\times|C_k|$ that represents the dual boundary operator $\tilde{\partial}_{n-k+1}$. Eq. (2.15) is used to establish the relationship between the primal and dual boundary operators as follows

$$\tilde{\mathbb{D}}_{n-k+1}=(-1)^k\,(\mathbb{D}_k)^{\mathsf{T}}\,,\quad k=1,\dots,n$$

Clearly, on the dual mesh we also have $\tilde{\mathbb{D}}_k\,\tilde{\mathbb{D}}_{k-1}=\mathbf{0}\,,\,\forall k=2,\dots,n$.

A dual coboundary operator can be defined analogously. The dual coboundary operator $\tilde{\delta}^{n-k-1}:C^{n-k-1}(\tilde{K})\to C^{n-k}(\tilde{K})$ is the adjoint of the dual boundary operator $\tilde{\partial}_{n-k}$ based on the generalized Stokes' theorem

$$\langle\tilde{c}_{(n-k)},\tilde{\delta}^{n-k-1}\tilde{\gamma}^{(n-k-1)}\rangle=\langle\tilde{\partial}_{n-k}\tilde{c}_{(n-k)},\tilde{\gamma}^{(n-k-1)}\rangle$$

for all $\tilde{c}_{(n-k)}\in C_{n-k}(\tilde{K})$ and $\tilde{\gamma}^{(n-k-1)}\in C^{n-k-1}(\tilde{K})$. Note that, by virtue of the duality pairing, the dual cochain $\tilde{\gamma}^{(n-k-1)}$ is an outer form that is integrated over the outer-oriented

boundary of $\tilde{c}_{(n-k)}$. The dual coboundary operator is represented by a $|C_k| \times |C_{k+1}|$ matrix $\tilde{\mathbb{D}}^{n-k-1}$ and is given by

$$\tilde{\mathbb{D}}^{n-k-1} = (-1)^{k+1} \left(\mathbb{D}^k\right)^{\mathsf{T}} , \quad k = 0, \ldots, n-1$$

since $\tilde{\mathbb{D}}^{n-k-1} = \tilde{\mathbb{D}}_{n-k}$ (and $\mathbb{D}^k = \mathbb{D}_{k+1}$) as per the Stokes' theorem. Additionally, we have $\tilde{\mathbb{D}}^{k+1}\tilde{\mathbb{D}}^k = \mathbf{0}$, $\forall k = 0, \ldots, n-2$.

By virtue of the Stokes' theorem, the discrete exterior derivative defined on the dual mesh is the same as the dual coboundary operator. The operator $\tilde{\mathbb{D}}^{n-k-1}$ turns an integral over a dual $k-$cell into a boundary integral, that is, over the boundary of the dual $k-$cell, and maps a discrete outer $(n-k-1)-$form to a discrete outer $(n-k)-$form. For example, $\tilde{\mathbb{D}}^{n-1}$ acting on the outer $(n-1)-$form, that is, the flux through a mesh face, yields an outer $n-$form (volume form). This is exactly the application of the divergence theorem and operator $\tilde{\mathbb{D}}^{n-1}$ is thus identified with the operator `div` (cf. right panel of Figure 2.1). Furthermore, we observe that $\tilde{\mathbb{D}}^{n-1} = -(\mathbb{D}^0)^{\mathsf{T}}$ which is the discrete version of the antisymmetry relation $\texttt{div} = -\texttt{grad}^{\mathsf{T}}$. As demonstrated in Section 2.3, this property plays a vital role in developing a physically consistent and stable numerical scheme.

Another form of symmetry is found for $k = 1$, namely $\tilde{\mathbb{D}}^{n-2} = (\mathbb{D}^1)^{\mathsf{T}}$. For $n = 2$ and $n = 3$ this yields $\tilde{\mathbb{D}}^0 = (\mathbb{D}^1)^{\mathsf{T}}$ and $\tilde{\mathbb{D}}^1 = (\mathbb{D}^1)^{\mathsf{T}}$, respectively. This is the discrete representation of $\texttt{curl} = \texttt{curl}^{\mathsf{T}}$, meaning that the curl operator is symmetric (aka self-adjoint). Finally, it can be observed that expression $\tilde{\mathbb{D}}^1\tilde{\mathbb{D}}^0 = \mathbf{0}$ for $n = 2$ or $\tilde{\mathbb{D}}^2\tilde{\mathbb{D}}^1 = \mathbf{0}$ for $n = 3$ discretely represents the identity $\texttt{div curl} = 0$ (the divergence of a curl is zero).

2.5.7 Discrete Hodge star operators

The duality between the primal and dual mesh elements suggests that we can define a mapping between a primal $k-$form and a dual $(n-k)-$form. In algebraic topology, this mapping is achieved with the help of the discrete Hodge star operator. This discrete operator will be used later on in the discretization process. While the discrete exterior derivative is uniquely determined by the Stokes' theorem, there is a great variety of discrete Hodge star operators. In a nutshell, a choice of dual mesh induces a choice of discrete Hodge star.

Given the vector space of discrete $k-$forms of primal mesh K on $\Omega \subset \mathbb{R}^2$, $C^k(K)$ and its dual, $C^{n-k}(\tilde{K})$, the kth discrete Hodge star operator is defined as a linear map $\mathbb{H}^k : C^k(K) \to C^{n-k}(\tilde{K})$ and is represented as a square matrix of size $|C_k| \times |C_k|$. The structure of the primal Hodge star matrix $\mathbb{H}^k$ (for $k = 0, \ldots, n$) depends on the dual mesh. In particular, the metric of the mesh geometry, such as the size and shape of the mesh elements, is the key ingredient of the Hodge star operator.

As will be explained in Section 2.5.8, a required condition for numerical stability is that the discrete Hodge star matrix is positive definite and symmetric. For this reason, we will use the circumcentric dual for the construction of a primal discrete Hodge star matrix. Particularly, the DEC (Discrete Exterior Calculus) approach of [21, 14, 15] is adopted in the current work.

Matrix $\mathbb{H}^k$ maps from a primal $k-$form to a dual $(n-k)-$form. This mapping must be consistent in the sense that both the primal and dual quantities have the same density. For example, for a given velocity vector field in $\mathbb{R}^3$, a primal $1-$form characterizes the total circulation along the primal edge while a dual $2-$form represents the total flux through to the dual face. To be consistent, the primal $1-$form must be scaled by the size of the edge while the dual $2-$form by the size of the face. Thus we wish to have the following

$$\frac{1}{|\star\,\sigma_{(k)}|}\int_{\star\sigma_{(k)}}\star\gamma^{(k)} = \frac{1}{|\sigma_{(k)}|}\int_{\sigma_{(k)}}\gamma^{(k)}$$

where $\star\sigma_{(k)}$ is the dual of the $k-$cell $\sigma_{(k)}$ and $\star\gamma^{(k)}$ is the dual of the $k-$form $\gamma^{(k)}$. Now any primal cell $\sigma_{(k)}$ and its circumcentric dual $\star\sigma_{(k)}$ are mutually orthogonal. This implies that for a particular primal cell and its dual, we have

$$\frac{1}{|\tilde{\sigma}_{(n-k),i}|}\langle\tilde{\sigma}_{(n-k),i},\star\gamma^{(k)}\rangle = \frac{1}{|\sigma_{(k),i}|}\langle\sigma_{(k),i},\gamma^{(k)}\rangle$$

and so,

$$\frac{\star\gamma_i}{|\tilde{\sigma}_{(n-k),i}|} = \frac{\gamma_i}{|\sigma_{(k),i}|}$$

The kth circumcentric primal Hodge star is thus a diagonal matrix with the entries

$$[\mathbb{H}^k]_{i,i} = \frac{|\tilde{\sigma}_{(n-k),i}|}{|\sigma_{(k),i}|}\,, \quad k=0,\ldots,n$$

Hence, the action of $\mathbb{H}^k$ on $\gamma^{(k)}$ (a column vector) is obtained by the multiplication $\mathbb{H}^k\gamma^{(k)}$ which is a dual $(n-k)-$form.

We also define the circumcentric dual Hodge star $\tilde{\mathbb{H}}^{n-k}$ which is the map from $C^{n-k}(\tilde{K})$ to $C^k(K)$, with $k=0,\ldots,n$, and is simply the inverse of the primal Hodge star, that is, $\tilde{\mathbb{H}}^{n-k} = \left(\mathbb{H}^k\right)^{-1}$. This inverse can be found immediately when a circumcentric dual mesh is employed, provided that the primal mesh is well centered. The choice for a circumcentric dual mesh is considered as an advantage compared to a barycentric dual mesh because barycentric primal Hodge matrices are not always invertible while dual Hodge matrices are typically dense due to the loss of the orthogonality property. This advantage will be elaborated in Section 2.5.8.

Let us continue with dimension $n=2$. As an example, we consider a simplicial mesh consisting of well-centered triangular cells. Figure 2.4 illustrates the different primal and dual $k-$cells of a $2-$simplex. Their volumes are also indicated in the figure. Let N_v, N_e and N_c be the number of primal vertices, edges and cells, respectively. The corresponding (circumcentric) primal Hodge star matrices are then given by

$$[\mathbb{H}^0]_{i,i} = A_{\mathrm{v}}\,, \quad i=1,\ldots,N_v \tag{2.16}$$

$$[\mathbb{H}^1]_{i,i} = \frac{d_{\mathrm{e}}}{l_{\mathrm{e}}}\,, \quad i=1,\ldots,N_e \tag{2.17}$$

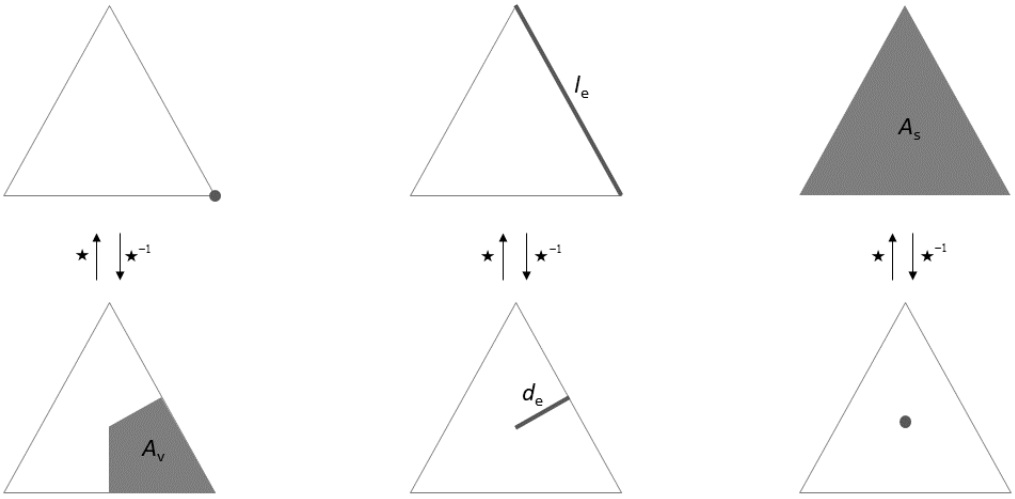


Figure 2.4: The primal triangular cell with its circumcentric dual. On the top row are shown three triangles with a $0-$cell, $1-$cell and $2-$cell, respectively. Furthermore, l_e is the length of the primal edge and A_s is the area of the primal cell. By convention, the size of the vertex is 1. The bottom row shows the respective dual $(2-k)-$cells as highligted *inside* the triangles and are constructed from the circumcentric subdivision. Quantities A_v and d_e indicate the area of the *whole* dual cell and the length of the *whole* dual face, respectively (they both are resided in adjacent primal cells as well). The symbol $\star$ signifies the Hodge star that maps from the primal mesh to the dual mesh and conversely. Note that the dual mesh is not explicitly used, only the volume of the dual mesh elements is stored.

$$[\mathbb{H}^2]_{i,i} = \frac{1}{A_s}, \quad i = 1, \ldots, N_c \tag{2.18}$$

Clearly, the entries of the Hodge matrices are metric dependent and are generally not dimensionless. It is interesting to note that the geometric interpretation of the action of Hodge matrix $\mathbb{H}^1$ is a rotation of a vector in $\mathbb{R}^2$ counterclockwise by 90^o. In addition, matrix $\mathbb{H}^0$ converts a scalar field to an area-integrated field while the action of $\mathbb{H}^2$ is to get a cell-averaged value of the cell-integrated field variable.

If the computational mesh is well centered, the above matrices are positive definite. However, for a right-angled triangle, matrix $\mathbb{H}^1$ is singular since the length of the edge dual to its hypotenuse is zero. Moreover, the circumcenter of an obtuse triangle is located outside the triangle, implying that $\mathbb{H}^1$ is negative definite. To overcome these unwanted cases, with SWASH the barycenter (or the centroid) is chosen locally instead of the circumcenter; see Section 2.5.10. Note that in case of a rectangular mesh, the matrices $\mathbb{H}^k$ $(k = 0, 1, 2)$ are always positive definite.

2.5.8 Discrete inner products

A special feature of an invertible Hodge star matrix is that it induces a discrete inner product. Let now $\alpha^{(k)}$ and $\beta^{(k)}$ be the real-valued $k-$forms defined on primal mesh K. An inner product of these two forms is defined by

$$\langle \alpha^{(k)}, \beta^{(k)} \rangle_{\mathbb{H}^k} \coloneqq \left(\alpha^{(k)}\right)^{\mathsf{T}} \mathbb{H}^k \beta^{(k)}$$

This bilinear operator $\langle \cdot, \cdot \rangle_{\mathbb{H}^k} : C^k(K) \times C^k(K) \to \mathbb{R}$ is symmetric and positive definite, provided that $\mathbb{H}^k$ is a symmetric positive definite matrix. Thus for such a matrix, its inverse exists and is symmetric and positive definite as well. Consequently, another discrete inner product can be provided in the following way,

$$\langle \tilde{\alpha}^{(k)}, \tilde{\beta}^{(k)} \rangle_{\tilde{\mathbb{H}}^k} = \left(\tilde{\alpha}^{(k)}\right)^{\mathsf{T}} \tilde{\mathbb{H}}^k \tilde{\beta}^{(k)} = \left(\tilde{\beta}^{(k)}\right)^{\mathsf{T}} \left(\tilde{\mathbb{H}}^k\right)^{\mathsf{T}} \tilde{\alpha}^{(k)} = \left(\tilde{\beta}^{(k)}\right)^{\mathsf{T}} \tilde{\mathbb{H}}^k \tilde{\alpha}^{(k)} = \langle \tilde{\beta}^{(k)}, \tilde{\alpha}^{(k)} \rangle_{\tilde{\mathbb{H}}^k}$$

where $\tilde{\alpha}^{(k)}, \tilde{\beta}^{(k)} \in C^k(\tilde{K})$.

Next, let the following discrete forms be given

$$\tilde{\beta}^{(n-k)} = \mathbb{H}^k \beta^{(k)}, \quad \alpha^{(k)} = \tilde{\mathbb{H}}^{n-k}\, \tilde{\alpha}^{(n-k)}$$

then we have

$$\langle \alpha^{(k)}, \beta^{(k)} \rangle_{\mathbb{H}^k} = \left(\alpha^{(k)}\right)^{\mathsf{T}} \tilde{\beta}^{(n-k)}$$

and

$$\langle \tilde{\alpha}^{(n-k)}, \tilde{\beta}^{(n-k)} \rangle_{\tilde{\mathbb{H}}^{n-k}} = \left(\tilde{\beta}^{(n-k)}\right)^{\mathsf{T}} \alpha^{(k)}$$

so that

$$\langle \tilde{\alpha}^{(n-k)}, \tilde{\beta}^{(n-k)} \rangle_{\tilde{\mathbb{H}}^{n-k}} = \langle \alpha^{(k)}, \beta^{(k)} \rangle_{\mathbb{H}^k}$$

As a matter of notation, inner products of mixed forms can be written as

$$\langle \alpha^{(k)}, \tilde{\beta}^{(n-k)} \rangle_{\mathbb{H}^k} = \langle \alpha^{(k)}, \beta^{(k)} \rangle_{\mathbb{H}^k}$$

and similarly,

$$\langle \tilde{\alpha}^{(n-k)}, \beta^{(k)} \rangle_{\tilde{\mathbb{H}}^{n-k}} = \langle \tilde{\alpha}^{(n-k)}, \tilde{\beta}^{(n-k)} \rangle_{\tilde{\mathbb{H}}^{n-k}}$$

An important aspect of inner products is related to the adjoint of linear operators. Recall from functional analysis that every linear operator on a Hilbert space comes with an adjoint operator, and they have a natural relation with respect to inner products. Let V^k and V^{k+1} be inner product vector spaces and $\mathbb{L}^k : V^k \to V^{k+1}$ be a linear operator. Then this operator induces an adjoint operator $\left(\mathbb{L}^k\right)^{\mathsf{T}} : V^{k+1} \to V^k$ in the following way

$$\langle \mathbb{L}^k \alpha^{(k)}, \beta^{(k+1)} \rangle_{\mathbb{H}^{k+1}} = \langle \alpha^{(k)}, \left(\mathbb{L}^k\right)^{\mathsf{T}} \beta^{(k+1)} \rangle_{\mathbb{H}^k}, \quad \forall \alpha^{(k)} \in V^k, \ \forall \beta^{(k+1)} \in V^{k+1}$$

with the inner products defined on the respective vector spaces.

Next, let us consider a linear map from V^k to itself, denoted $\mathbb{C}^k : V^k \to V^k$. This operator is called self-adjoint (or symmetric) if

$$\langle \mathbb{C}^k \alpha^{(k)}, \beta^{(k)} \rangle_{\mathbb{H}^k} = \langle \alpha^{(k)}, \mathbb{C}^k \beta^{(k)} \rangle_{\mathbb{H}^k}, \quad \forall \alpha^{(k)}, \beta^{(k)} \in V^k$$

implying that

$$\left(\mathbb{C}^k\right)^{\mathsf{T}} = \mathbb{C}^k$$

In addition, the operator is called skew-adjoint (or skew-symmetric) if

$$\langle \mathbb{C}^k \alpha^{(k)}, \beta^{(k)} \rangle_{\mathbb{H}^k} = -\langle \alpha^{(k)}, \mathbb{C}^k \beta^{(k)} \rangle_{\mathbb{H}^k}, \quad \forall \alpha^{(k)}, \beta^{(k)} \in V^k$$

and hence,

$$\left(\mathbb{C}^k\right)^{\mathsf{T}} = -\mathbb{C}^k$$

This operator has the special property that $\langle \alpha^{(k)}, \mathbb{C}^k \alpha^{(k)} \rangle_{\mathbb{H}^k} = 0$ for all $\alpha^{(k)} \in V^k$.

The role of discrete inner products in the discretization process becomes clear by considering the Hamiltonian of the inviscid shallow water equations (see Section 2.3). In particular, for a given discrete $k-$form $\alpha^{(k)}$, its energy norm $\langle \alpha^{(k)}, \alpha^{(k)} \rangle_{\mathbb{H}^k}$ is properly defined by the symmetric positive definite Hodge star, making it possible to derive directly an energy conserving (and thus stable) discretization. The above considerations will be useful later on in Section 2.6.

2.5.9 Discrete de Rham complexes

A cochain complex is a sequence of vector spaces and linear operators $(V^k, \mathbb{L}^k)$ such that $\mathbb{L}^{k+1}\,\mathbb{L}^k = 0$. When these spaces refer to the spaces of *discrete forms* with the same orientation while the linear operator is the *discrete exterior derivative*, then this cochain complex is called the *discrete de Rham complex*.

Using the discrete exterior derivative and discrete Hodge star, a diagram can be composed as illustrated in Figure 2.5. This diagram reflects on how the discretization process works. The lower part of the diagram is the sequence of spaces of inner-oriented discrete forms $C^k(K)$ connected with $\mathbb{D}^k$. This sequence is a cochain complex since $\mathbb{D}^{k+1}\,\mathbb{D}^k = \mathbf{0}$. The upper part of the diagram constitutes a cochain complex of outer-oriented discrete forms $C^k(\tilde{K})$ with the operator $\tilde{\mathbb{D}}^k$ that satisfies the property $\tilde{\mathbb{D}}^{k+1}\,\tilde{\mathbb{D}}^k = \mathbf{0}$. These two oriented cochain complexes are dual with respect to each other. Note that we have tacitly assumed that the primal mesh K is endowed with inner orientation and the dual mesh $\tilde{K}$ with outer orientation, but this is rather an arbitrary choice. (In Section 2.6, we will choose this the other way around.) Finally, the primal and dual complexes are connected by the Hodge star operator, which completes the double de Rham complex as shown in Figure 2.5.

In the discrete setting, the horizontal links are encoded by the incidence matrices based on the topological relations of mesh objects. The vertical links are constructed through the Hodge star matrices that are completely metric (or local) dependent. It is precisely this construction that is a determining factor in the development of the numerical framework to be discussed in Section 2.6.

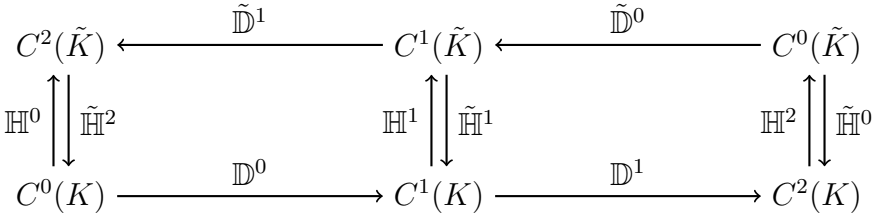


Figure 2.5: The discrete double de Rham complex in two dimensions with the lower part depicting the cochain complex of inner-oriented discrete forms and the upper part that of the outer-oriented discrete forms. Let $\mathbb{L}^k$ denotes either $\mathbb{D}^k$ or $\tilde{\mathbb{D}}^k$, then each of these cochain complexes is a sequence of linear spaces of discrete forms connected with the exterior derivative $\mathbb{L}^k$ with the property $\mathbb{L}^{k+1}\,\mathbb{L}^k = \mathbf{0}$. The cochain complexes of inner- and outer-oriented forms are linked by means of the Hodge star operators $\mathbb{H}^k$ and $\tilde{\mathbb{H}}^k$.

As a first example, the double de Rham complex can be employed to construct the Laplacian of the pressure: $\Delta p = \nabla \cdot \nabla p$. We start at the bottom left of the diagram by defining an inner $0-$form, denoted $\pi^{(0)}$. Next, we apply the matrix $\mathbb{D}^0$ (the gradient) to obtain an inner $1-$form. This is followed by the Hodge matrix $\mathbb{H}^1$ to convert the result to an outer $(n-1)-$form. (We could have written "$1-$form" since $n = 2$, but to emphasize this outer form embedded in $\mathbb{R}^n$ we use the space dimension.) Then, matrix $\tilde{\mathbb{D}}^1$ (the divergence) or $\tilde{\mathbb{D}}^{n-1}$ (again to emphasize its relation to outer forms) is applied to get an outer $n-$form. Finally, this volume form is transformed back to a point value by means of the Hodge matrix $\tilde{\mathbb{H}}^2$ or here $\tilde{\mathbb{H}}^n$. (The Laplacian itself is a scalar field function.) Thus, the discretization of the Laplace (or Poisson) operator of the pressure is given by

$$\tilde{\mathbb{H}}^n\,\tilde{\mathbb{D}}^{n-1}\,\mathbb{H}^1\,\mathbb{D}^0\,\pi^{(0)}$$

and is considered mimetic because it respects the vector calculus identities and symmetries. This discrete scalar Laplacian can be implemented on arbitrary well-centered meshes, either 2D or 3D, provided that their metric is given.

The second example demonstrates how to discretize $\Delta\mathbf{u}$. The velocity vector $\mathbf{u}$ is discretized as an inner $1-$form. We denote by $\upsilon^{(1)}$ this discrete form. We thus start at the bottom center of the diagram. We can walk through the diagram in two ways. We can first apply the matrix $\mathbb{D}^1$ (the curl), next the matrix $\mathbb{H}^2$ followed by $\tilde{\mathbb{D}}^{n-2}$ (another curl) and finally matrix $\tilde{\mathbb{H}}^{n-1}$. The other route is first $\mathbb{H}^1$, then $\tilde{\mathbb{D}}^{n-1}$ (the divergence), matrix $\tilde{\mathbb{H}}^n$ and lastly the matrix $\mathbb{D}^0$ (the gradient). Now, these two operations together make up the vector Laplacian. Hence, its discretized form reads

$$\mathbb{D}^0\,\tilde{\mathbb{H}}^n\,\tilde{\mathbb{D}}^{n-1}\,\mathbb{H}^1\,\upsilon^{(1)} + \tilde{\mathbb{H}}^{n-1}\,\tilde{\mathbb{D}}^{n-2}\,\mathbb{H}^2\,\mathbb{D}^1\,\upsilon^{(1)}$$

which is precisely the vector calculus identity $\Delta\mathbf{u} = \nabla\,(\nabla \cdot \mathbf{u}) - \nabla \times (\nabla \times \mathbf{u})$. Note, however, that the appearance of the minus sign is enforced because vector calculus does not deal with geometry and orientation.

The above examples demonstrate nicely the strength of the double de Rham complex that provides a natural way to discretize first and second order differential operators while

mimicking vector calculus identities. This promotes the physical fidelity and accuracy of the discretization.

2.5.10 Examples

In this section we provide some sample calculations to see how things work out in the case of a square cell (or a rectangle), an equilateral triangle cell and a right-angled triangle cell in $\mathbb{R}^2$ as depicted in Figure 2.6. Vertices, edges and the face (either square or triangle) are

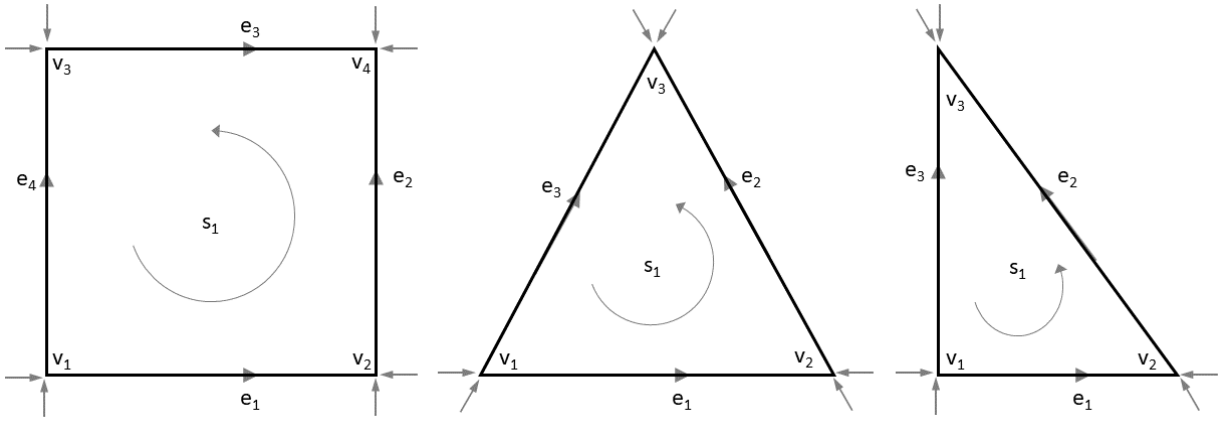


Figure 2.6: Primal computational cells with oriented $k-$cells ($k = 0, 1, 2$): (a) square, (b) equilateral triangle, and (c) right-angled triangle. Orientation is indicated by the gray arrows.

denoted as v_i, e_i and s_1, respectively, for $i = 1, \ldots, p$ with $p = 3$ in case of triangles and $p = 4$ in case of the square. The adopted convention for the inner orientation to these mesh elements is shown in the figure, with the condition that the edges are oriented such that they point toward the vertex index of greater value.

Let us consider the square first as shown in Figure 2.6a. This square is a cell complex, denoted K_{sq}. The vertices of this square, $\{v_1, v_2, v_3, v_4\}$, are the basis of the linear space of $0-$chains. Examples of $0-$chains are

v_1,

v_2,

$v_3 + v_4$,

$v_1 + v_2 + v_4$,

etc.

Similarly, the oriented edges of K_{sq}, $\{e_1, e_2, e_3, e_4\}$, are the basis of $C_1(K_{\text{sq}})$. A few examples of $1-$chains are

e_1,

$e_1 + e_2 - e_3$,

$-e_1 + e_4$

The corresponding row vectors are $\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$, $\begin{bmatrix}1 & 1 & -1 & 0\end{bmatrix}$, and $\begin{bmatrix}-1 & 0 & 0 & 1\end{bmatrix}$. Next, we can take the boundary of such chains, namely,

$\partial_1 e_1 = v_2 - v_1$,

$\partial_1(e_1 + e_2 - e_3) = \partial_1 e_1 + \partial_1 e_2 - \partial_1 e_3 = v_2 - v_1 + v_4 - v_2 + v_3 - v_4 = v_3 - v_1$,

$\partial_1(-e_1 + e_4) = -\partial_1 e_1 + \partial_1 e_4 = v_1 - v_2 - v_1 + v_3 = v_3 - v_2$

Referring to Eq. (2.13), the boundary of the second chain is made up of all $0-$chains of cell complex K_{sq} with coefficients $c^1 = 1$, $c^2 = 1$, $c^3 = -1$, and $c^4 = 0$, whereas the orientation coefficients are $o_{1,1} = -1$, $o_{1,2} = 1$, $o_{2,1} = -1$, $o_{2,2} = 1$, $o_{3,1} = -1$, $o_{3,2} = 1$, and $o_{4,1} = -1$, $o_{4,2} = 1$.

There is only one oriented $2-$chain which is s_1. Its boundary equals

$$\partial_2 s_1 = e_1 + e_2 - e_3 - e_4$$

The boundary of this boundary is zero, that is,

$$\partial_1 \partial_2 s_1 = \partial_1 e_1 + \partial_1 e_2 - \partial_1 e_3 - \partial_1 e_4 = v_2 - v_1 + v_4 - v_2 + v_3 - v_4 + v_1 - v_3 = 0$$

By virtue of Eq. (2.14), the boundary operators ∂_1 and ∂_2 are encoded, respectively, by the following incidence matrices

$$\mathbb{D}_1 = \begin{bmatrix} -1 & +1 & 0 & 0 \\ 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & +1 \\ -1 & 0 & +1 & 0 \end{bmatrix}, \qquad \mathbb{D}_2 = \begin{bmatrix} +1 & +1 & -1 & -1 \end{bmatrix}$$

For example, the boundary of the second $1-$chain from the above example can be obtained as follows

$$\begin{bmatrix}1 & 1 & -1 & 0\end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix}-1 & 0 & 1 & 0\end{bmatrix}$$

which implies $\partial_1(e_1 + e_2 - e_3) = -v_1 + v_3$. One can easily verify that the boundary operator is nilpotent, that is, $\mathbb{D}_2 \, \mathbb{D}_1 = \mathbf{0}^{\mathsf{T}}$.

Since the coboundary operator is the dual of the boundary operator, we find

$$\mathbb{D}^0 = \begin{bmatrix} -1 & +1 & 0 & 0 \\ 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & +1 \\ -1 & 0 & +1 & 0 \end{bmatrix}, \qquad \mathbb{D}^1 = \begin{bmatrix} +1 & +1 & -1 & -1 \end{bmatrix}$$

Note that these matrices are coordinate independent and hold for an arbitrary (curved) quadrilateral mesh element.

Discrete $k-$forms maps an oriented $k-$cell to a real value. For instance, a $0-$form represents a scalar function that produces its value on vertices. Let us define the pressure p on vertices v_i, denoted $\pi_i = p(v_i)$. This discrete inner $0-$form is represented as a column vector with 4 entries. We multiply this vector from the left by matrix $\mathbb{D}^0$,

$$\begin{bmatrix} -1 & +1 & 0 & 0 \\ 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & +1 \\ -1 & 0 & +1 & 0 \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{bmatrix} = \begin{bmatrix} \pi_2 - \pi_1 \\ \pi_4 - \pi_2 \\ \pi_4 - \pi_3 \\ \pi_3 - \pi_1 \end{bmatrix}$$

This result stems from the generalized Stokes' theorem, namely, evaluating the exterior derivative of p on edge e_1 is identical to evaluating p on the edge boundaries as $p(v_2) - p(v_1)$. This is simply the classical fundamental theorem of calculus for line integrals. Thus, the value of the $1-$form $\delta^{(0)}p$ is established as the integral quantity on oriented edges. Matrix $\mathbb{D}^0$ is therefore the discrete analogue of the gradient operator ∇.

As a second example, let γ_i be defined as the circulation along edge e_i. Then operator $\mathbb{D}^1$ relates this inner $1-$form to the inner $2-$form, as follows

$$\begin{bmatrix} +1 & +1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{bmatrix} = \gamma_1 + \gamma_2 - \gamma_3 - \gamma_4$$

Here, the analogy with the Stokes' curl theorem is obvious and $\mathbb{D}^1$ is thus the discrete version of the curl operator $\nabla\times$. In addition, we find that $\mathbb{D}^1\mathbb{D}^0 = \mathbf{0}^\mathsf{T}$, which is the discrete analogue of the identity $\nabla \times \nabla = \mathbf{0}$.

To summarize, the discrete operators $\mathbb{D}^0$ and $\mathbb{D}^1$ represent exactly the continuous gradient and curl operators, respectively, on a 2D primal mesh without reference to any coordinate system.

Next, we turn to the equilateral triangle shown in Figure 2.6b. From this we can deduce the following discrete version of the gradient and the curl, respectively,

$$\mathbb{D}^0 = \begin{bmatrix} -1 & +1 & 0 \\ 0 & -1 & +1 \\ -1 & 0 & +1 \end{bmatrix}, \quad \mathbb{D}^1 = \begin{bmatrix} +1 & +1 & -1 \end{bmatrix}$$

Again, we observe that $\mathbb{D}^1\mathbb{D}^0 = \mathbf{0}^\mathsf{T}$. Since the boundary and coboundary operators are defined purely topologically, the matrices found above also apply to the right-angled triangle of Figure 2.6c.

In the following, we consider the dual of the square and the triangles as shown in Figure 2.7. The labeling of the mesh elements are now indicated by a tilde. This time we only have one vertex and multiple faces for each mesh cell. Moreover, all the $1-$cells and

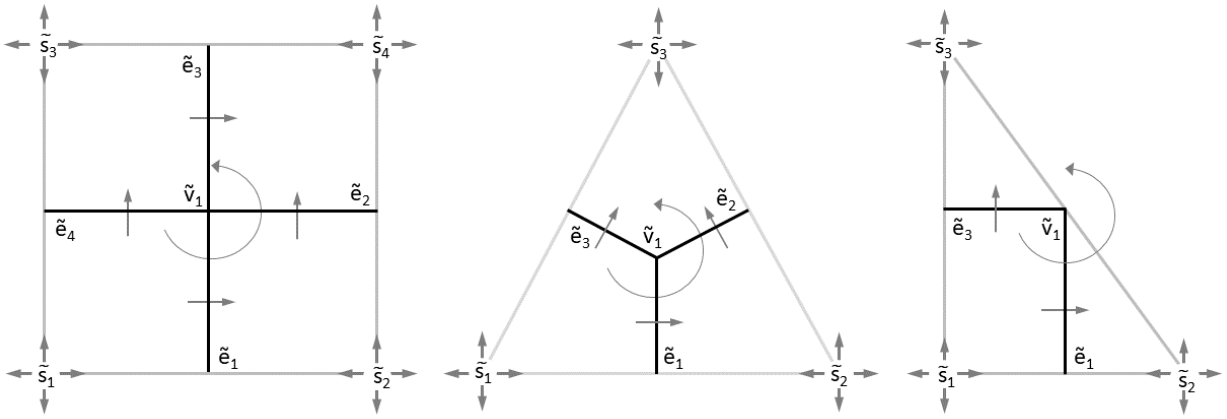


Figure 2.7: Dual computational cells with oriented $(2-k)-$cells ($k = 0, 1, 2$): (a) square, (b) equilateral triangle, and (c) right-angled triangle. Orientation is indicated by the gray arrows.

$2-$cells are open ended, and therefore the considered mesh cells are not a cell complex. We also notice that one edge is "missing" (its length is zero) in the right triangle cell (Fig. 2.7c).

The outer orientation on the dual cells is depicted in Figure 2.7. For the first example, the dual of the square, the boundary operators acting on dual $1-$cells and $2-$cells are given by, respectively,

$$\tilde{\partial}_1 \begin{bmatrix} \tilde{e}_1 \\ \tilde{e}_2 \\ \tilde{e}_3 \\ \tilde{e}_4 \end{bmatrix} = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_1 \\ -\tilde{v}_1 \\ -\tilde{v}_1 \end{bmatrix}, \qquad \tilde{\partial}_2 \begin{bmatrix} \tilde{s}_1 \\ \tilde{s}_2 \\ \tilde{s}_3 \\ \tilde{s}_4 \end{bmatrix} = \begin{bmatrix} \tilde{e}_1 + \tilde{e}_4 \\ \tilde{e}_2 - \tilde{e}_1 \\ \tilde{e}_3 - \tilde{e}_4 \\ -\tilde{e}_2 - \tilde{e}_3 \end{bmatrix}$$

Subsequently, the coboundary operators can be found as follows

$$\tilde{\mathbb{D}}^0 = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix}, \qquad \tilde{\mathbb{D}}^1 = \begin{bmatrix} +1 & 0 & 0 & +1 \\ -1 & +1 & 0 & 0 \\ 0 & 0 & +1 & -1 \\ 0 & -1 & -1 & 0 \end{bmatrix}$$

Note that $\tilde{\mathbb{D}}^1$ is the discrete analogue of the divergence operator $\nabla\cdot$. We also observed that $\tilde{\mathbb{D}}^1$ is the negative transpose of $\mathbb{D}^0$, that is, $\tilde{\mathbb{D}}^1 = -(\mathbb{D}^0)^\mathsf{T}$, which is the discrete version of the antisymmetry relation $\nabla\cdot = -(\nabla)^\mathsf{T}$.

As illustrated by Figure 2.7a, operator $\tilde{\mathbb{D}}^0$ is identified with the curl operator. Moreover, we also see that $\tilde{\mathbb{D}}^0 = (\mathbb{D}^1)^\mathsf{T}$, which implies $\nabla\times = (\nabla\times)^\mathsf{T}$. Finally, we find that $\tilde{\mathbb{D}}^1\tilde{\mathbb{D}}^0 = \mathbf{0}$ which is the discrete representation of $\nabla \cdot \nabla\times = \mathbf{0}$.

In the same vein, we can derive the coboundary operators for the triangles of Figure 2.7b and 2.7c, which are

$$\tilde{\mathbb{D}}^0 = \begin{bmatrix} +1 \\ +1 \\ -1 \end{bmatrix}, \quad \tilde{\mathbb{D}}^1 = \begin{bmatrix} +1 & 0 & +1 \\ -1 & +1 & 0 \\ 0 & -1 & -1 \end{bmatrix}$$

Despite the missing edge $\tilde{e}_2$ in the right triangle of Figure 2.7c, one should remember that the coboundary operators are topological, that is, independent of the shape of mesh elements. Finally, one can observe that the above findings related to symmetries and identities remain valid for triangular cells.

Thus far, we have seen that the discrete exterior derivative represents the exact discretization of the differential operators `grad`, `curl`, and `div` in the form of matrices that are purely topological. Such matrices act on coordinate-free variables or physical quantities (discrete forms) defined on vertices, edges and faces. Since the discrete exterior derivative obeys the generalized Stokes' theorem by construction, the resulting discrete `grad`, `curl`, and `div` operators naturally mimic the vector calculus identities $\texttt{curl grad} = 0$ and $\texttt{div curl} = 0$ and the symmetry relations $\texttt{div} = -\texttt{grad}^\mathsf{T}$ and $\texttt{curl} = \texttt{curl}^\mathsf{T}$.

The discrete Hodge star operators, on the contrary, do not depend on the mesh topology, but only on the metric (lengths, areas and volumes) of the various mesh elements. A discrete Hodge star maps between variables living on an inner-oriented mesh and variables living on an outer-oriented mesh. This map always involves some form of approximation. Basically, a choice of discrete Hodge star is much like a choice of dual mesh. Here, we adopted the circumcentric dual like in the examples above (cf. Figure 2.7) which is desired when considering the stability of a numerical scheme.

We now return to the examples to demonstrate how the circumcentric Hodge star matrices are computed. Recall the square of Figure 2.7a. The size of this square is 1 and the circumcenter is $(\frac{1}{2}, \frac{1}{2})$. Hence, the intrinsic volume of the primal edges is $|e_i| = 1$, $i = 1, \ldots, 4$, and the primal face is $|s_1| = 1$. Note that $|v_i| = 1$ by definition. Furthermore, we have for the dual edges and faces *inside* the square, $|\tilde{e}_i| = \frac{1}{2}$ and $|\tilde{s}_i| = \frac{1}{4}$, $i = 1, \ldots, 4$. The Hodge star matrix that acts on $0-$, $1-$ and $2-$forms is then given by, respectively,

$$\mathbb{H}^0 = \frac{1}{4}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbb{H}^1 = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbb{H}^2 = \begin{bmatrix} 1 \end{bmatrix}$$

We can conclude that all the three matrices are diagonal and positive definite.

Next, consider the equilateral triangle of Figure 2.7b. Let $v_1 = (-\frac{1}{2}\sqrt{3}, 0)$, $v_2 = (\frac{1}{2}\sqrt{3}, 0)$ and $v_3 = (0, \frac{3}{2})$ be vertices of the triangle. The circumcenter is then $(0, \frac{1}{2})$ and the area of the triangle is $\frac{3}{4}\sqrt{3}$. Furthermore, $|e_i| = \sqrt{3}$, $|\tilde{e}_i| = \frac{1}{2}$, $|s_1| = \frac{3}{4}\sqrt{3}$ and $|\tilde{s}_i| = \frac{1}{4}\sqrt{3}$, $i = 1, 2, 3$. The corresponding Hodge star matrices are then given by

$$\mathbb{H}^0 = \frac{\sqrt{3}}{4}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbb{H}^1 = \frac{\sqrt{3}}{6}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbb{H}^2 = \frac{4\sqrt{3}}{9}\begin{bmatrix} 1 \end{bmatrix}$$

which are again symmetric positive definite.

For the final example the vertices of the right triangle of Figure 2.7c are $v_1 = (0, 0)$,

$v_2 = (1, 0)$ and $v_3 = (0, 1)$. Thus, with $|s_1| = \frac{1}{2}$, $|\tilde{s}_1| = \frac{1}{4}$, $|\tilde{s}_2| = |\tilde{s}_3| = \frac{1}{8}$, we have

$$\mathbb{H}^0 = \frac{1}{8}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbb{H}^2 = \begin{bmatrix} 2 \end{bmatrix}$$

which are positive definite. Now, the triangle is not well centered since the circumcenter $(\frac{1}{2}, \frac{1}{2})$ lies exactly on the longest side of the triangle, meaning that $|\tilde{e}_2| = 0$. This implies that matrix $\mathbb{H}^1$ is not invertible because it is a singular matrix, namely,

$$\mathbb{H}^1 = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In SWASH, this is remedied by choosing the barycenter (the mean of the three vertices) instead. In the current example, this center is $(\frac{1}{3}, \frac{1}{3})$. Consequently, $|e_1| = |e_3| = 1$, $|e_2| = \sqrt{2}$, $|\tilde{e}_1| = |\tilde{e}_3| = \frac{\sqrt{5}}{6}$, $|\tilde{e}_2| = \frac{\sqrt{2}}{6}$ and $|\tilde{s}_i| = \frac{1}{6}$, $i = 1, 2, 3$, which yields

$$\mathbb{H}^0 = \frac{1}{6}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbb{H}^1 = \frac{1}{6}\begin{bmatrix} \sqrt{5} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{5} \end{bmatrix}, \quad \mathbb{H}^2 = \begin{bmatrix} 2 \end{bmatrix}$$

In particular, the use of this adapted matrix $\mathbb{H}^1$ gives rise to a locally inconsistent calculation of a dual $1-$form from a primal $1-$form and vice versa. (This also holds true for matrix $\mathbb{H}^0$, but as will become apparent, it will not be used in our discretization method; see Section 2.6.) Yet, in practice we see that the impact of the discretization error induced by this local inconsistency is typically very limited. Note that, in this context, generation of well-centered meshes is *not strictly* required for our discretization method (see also [39]).

2.6 Mimetic framework for the inviscid shallow water equations on general meshes

2.6.1 Introduction

Section 2.5 addressed some key ideas that are invaluable for the discretization process, namely, the discrete forms, the generalized Stokes' theorem, the discrete exterior derivative and the primal-dual meshes. The reason is threefold.

First, unlike vectors, discrete forms are coordinate and metric free and therefore have the same form and properties in all coordinate systems. Since discrete forms are defined by their values at discrete mesh elements, this also highlights their different roles in the spatial discretization (e.g., mass circulation vs mass flux while both representing the velocity vector).

Secondly, the main application of the generalized Stokes' theorem is to construct the discrete exterior derivative and, in turn, to derive discrete counterparts of the continuous

differential operators, viz. `grad`, `curl` and `div`. Like discrete forms, the associated discrete operators are independent of the coordinate system. Moreover, they have an intrinsically discrete nature that allows their exact representation in the numerical framework, including the vector calculus identities $\texttt{curl grad} = 0$ and $\texttt{div curl} = 0$.

Finally, the two types of orientation (inner and outer) of the various mesh elements reveal the primal-dual grid structure of the discretization. This naturally induces the layout of staggered grids that lies at the root of the Arakawa C-grid finite difference method. More importantly, the primal-dual framework enables to construct *exact* discrete expressions for symmetry relations like $\texttt{div} = -\texttt{grad}^{\mathsf{T}}$, which is required to prevent nonlinear computational instability.

The main benefits of obeying identities and symmetry relations at the discrete level include the compatibility with physics and conservation of energy. In general, mimetic methods for spatial discretization yield physically reliable results, even at relatively coarse grid spacings, have favorable stability properties, and do not suffer from spurious modes. Especially, the Arakawa C-grid method is best known for its stability and efficiency and has attracted great attention of various researchers and engineers in the last five decades.

This section is devoted to deriving a general mimetic framework for the numerical solution of the inviscid shallow water equations (2.1) and (2.2) on general meshes by utilizing the concepts of algebraic topology. This derived framework forms the basis for the classical Arakawa C-grid for rectilinear grids in Chapter 3 and for curvilinear grids in Chapter 4, and the covolume method using unstructured triangular meshes in Chapter 5.

2.6.2 General mimetic framework

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain on which Eqs. (2.1)−(2.2) are given. This domain is discretized by a well-centered polygonal mesh, either triangular or rectangular. Since the mesh is generated by a mesh generator, its boundary edges are aligned with the domain boundaries. Hence, this mesh is a cell complex and, for this reason, it is considered as the primal mesh. On the other hand, its circumcentric dual is not a cell complex because the boundary of dual cells is missing near the domain boundary. Figure 2.8 shows an example of a triangular mesh and its dual.

The computational mesh is composed of N_v vertices, N_e edges/faces and N_c cells. Here, we make an explicit distinction between edges as lines (1−cells) on the one hand, and faces as planes ($(n-1)$−cells) on the other, even though they are coincident lines in two dimensions ($n = 2$). It should be noted that the orientation of primal faces induces the orientation of dual edges. Furthermore, by duality, N_c, N_e and N_v also give the number of vertices, edges and cells, respectively, in the dual mesh.

Discretization of continuity equation

We start with the semi-discretization of the continuity equation (2.1). To enforce mass conservation in *every* cell of the mesh, we choose the primal mesh to which the discretization of Eq. (2.1) will be associated. This equation contains two unknowns, namely, h and $\mathbf{q}$. The

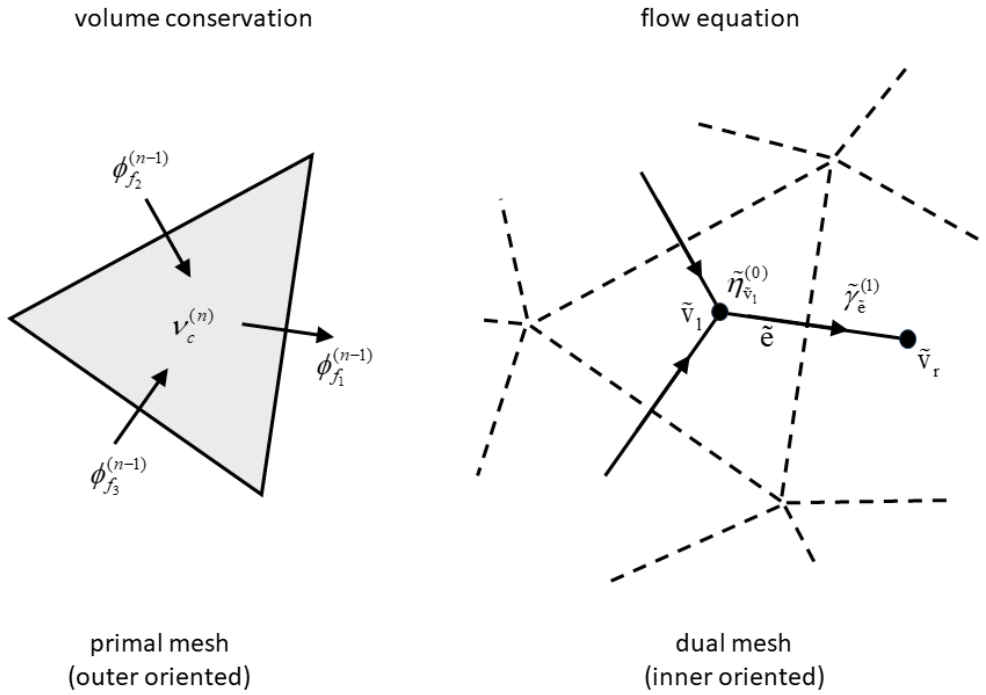


Figure 2.8: Parts of staggered orthogonal triangular mesh and the primary unknowns involved for the shallow water equations. Left panel depicts the primal mesh (cell complex shown as solid lines) with outer discrete forms $\nu^{(n)}$ and $\phi^{(n-1)}$, and right panel shows the dual mesh (solid lines indicating not a cell complex) with inner discrete forms $\tilde{\eta}^{(0)}$ and $\tilde{\gamma}^{(1)}$. Definitions of discrete forms and mesh elements are provided in the text.

mass flux $\mathbf{q}$ is encoded by a discrete outer $(n-1)-$form. It is represented by its surface integrals over the outer-oriented faces of the mesh. Each integral value is constant per planar face. For instance, there are three face values for each primal triangular cell (see left panel of Figure 2.8). Also note that the outer orientation of the face defines a direction of positive flux; see Figure 2.3b. The set of the integral values on faces provides a metric-free representation of the flux field. This set is arranged as a column vector with N_e entries with each entry assigned to a mesh face. We denote this vector by $\phi^{(n-1)}$.

In the context of incompressible shallow water flows with free surface, mass is usually expressed in terms of the volume of the water column, or the water height $h(\mathbf{x}, t)$, while the water density is assumed constant throughout the flow field. Therefore, by mass we refer to the area-integrated height and is treated as an outer discrete $n-$form. Its discretization is thus defined on primal $n-$cells (here, computational mesh cells). It is an *outer-oriented* volume form because the net change in the water column is due to the net outflow through the boundaries of the $n-$cell (cf. Figure 2.3b). The associated discrete values are stored as elements of a column vector of size N_c, denoted $\nu^{(n)}$. (Each entry is given as $\nu_c^{(n)}$, see Figure 2.8.)

Clearly, the primal mesh is outer oriented; see left panel of Figure 2.8. Furthermore,

when we denote the discrete analogue of the divergence operator by the incidence matrix $\mathbb{D}^{n-1}$ of size $N_c \times N_e$, then the semi-discrete version of the continuity equation is given by

$$\frac{d\nu^{(n)}}{dt} + \mathbb{D}^{n-1}\phi^{(n-1)} = 0 \tag{2.19}$$

which exactly conserves mass (or volume) in each cell of the mesh. Note that the action of $\mathbb{D}^{n-1}$ on $\phi^{(n-1)}$ results in an $n-$form so that Eq. (2.19) consistently contains only outer $n-$forms.

Discretization of momentum equation

By duality, the spatial discretization of the momentum equation (2.2) is implemented on the inner-oriented dual mesh. Starting with the time derivative term, the quantity $h\mathbf{u}$ is discretized as an inner $1-$form defined on the dual edges which measures the total mass circulation along the edges. It is denoted by $\tilde{\gamma}^{(1)}$ which is a column vector of length N_e. Since the dual edges are straight lines, each entry is constant per edge and also defines the flow along the edge (as indicated by $\tilde{\gamma}^{(1)}_{\tilde{e}}$ in right panel of Figure 2.8). Accordingly, it describes the flow field on the dual mesh, independent of the coordinate system.

The right-hand side of Eq. (2.2) is the pressure gradient which represents a driving force along the flow direction. For consistency reasons, the discretization of the term $h\nabla\zeta$ must be an inner $1-$form evaluated on the dual edges. Here, we show how to derive its discretization in a mimetic way.

Let $\tilde{e}$ be a dual edge, $\tilde{v}_{\text{l}}$ be its left vertex and $\tilde{v}_{\text{r}}$ its right vertex, see right panel of Figure 2.8. According to the inner orientation of $1-$cells (cf. Figure 2.3a), the boundary of $\tilde{e}$ is given by $\tilde{\partial}_1\tilde{e} = \tilde{v}_{\text{r}} - \tilde{v}_{\text{l}}$. Next, let us define the grid functions (discrete $0-$forms) for the free surface, the bed level and the water depth by ζ_i, d_i and h_i, respectively, with i the index of dual vertex $\tilde{v}_i$. By noting that

$$\nabla\, \tfrac{1}{2}gh^2 = gh\nabla\zeta + gh\nabla d$$

the application of the dual coboundary operator $\tilde{\delta}^0$ to $\frac{1}{2}gh^2$ on $\tilde{e}$ yields

$$\frac{1}{2}gh_{\text{r}}^2 - \frac{1}{2}gh_{\text{l}}^2 = \frac{1}{2}g\,(h_{\text{r}} + h_{\text{l}})\,(h_{\text{r}} - h_{\text{l}}) = g\overline{h}_{\tilde{e}}\,(\zeta_{\text{r}} - \zeta_{\text{l}}) + g\overline{h}_{\tilde{e}}\,(d_{\text{r}} - d_{\text{l}})$$

where $\overline{h}_{\tilde{e}}$ is the arithmetic mean of the two water depths, each on one side of the edge,

$$\overline{h}_{\tilde{e}} \coloneqq \frac{1}{2}\,(h_{\text{l}} + h_{\text{r}})$$

Note that this average value, although associated with the dual edge, is a $0-$form. (The sum of two $k-$forms is a $k-$form.)

The above discretization is exact and provides a discrete expression for the product term $gh\nabla\zeta$ in terms of inner $0-$forms. We first consider both operands separately and then look into the product term.

First, let us collect all the point values of the water depth h_i into a column vector $\tilde{\eta}^{(0)}$ of size N_c. We compute the arithmetic mean of this inner 0−form on the dual edges as explained above. We denote this mean by $\overline{\tilde{\eta}^{(0)}}$. We observe that this action of averaging returns a column vector of length N_e. It is important to note that the *arithmetic* averaging of an arbitrary discrete form is completely unrelated to the metric.

Next, we sample the discrete values of the water level at the vertices of the dual mesh. We thus obtain a column vector with N_c elements which is represented by $\tilde{\zeta}^{(0)}$. The discretization of $\nabla\zeta$ on the dual mesh is then given by $\tilde{\mathbb{D}}^0\tilde{\zeta}^{(0)}$ which is the inner 1−form defined on the dual edges. Here, the incidence matrix $\tilde{\mathbb{D}}^0$ of size $N_e \times N_c$ represents the discrete gradient operator $\tilde{\delta}^0$ on the dual mesh. Note that by construction $\tilde{\mathbb{D}}^0 = -\left(\mathbb{D}^{n-1}\right)^{\mathsf{T}}$, which is required to ensure energy conservation (see below). It should also be highlighted that $\tilde{\mathbb{D}}^1\,\tilde{\mathbb{D}}^0 = \mathbf{0}^{\mathsf{T}}$ which implies that the discrete pressure gradient is curl free. Its practical importance is most evident for depth-averaged rotating flow conditions, as the pressure gradient cannot act as a spurious source of vorticity $\nabla \times h\mathbf{u}$.

Finally, the mimetic discretization of the pressure gradient $gh\nabla\zeta$ is given by

$$g\,\overline{\tilde{\eta}^{(0)}} \odot \tilde{\mathbb{D}}^0\tilde{\zeta}^{(0)}$$

where $\odot$ symbolizes the element-wise multiplication of two vectors of the same dimension. This binary operation returns an inner 1−form of the same length. (The multiplication of any k−form by a 0−form is a k−form.)

With respect to the second term in the left-hand side of Eq. (2.2), the conditions imposed on the discretization of the term $A\mathbf{u} = \nabla\cdot(\mathbf{q}\otimes\mathbf{u})$ are discussed in Section 2.3. In particular, its discretization must result in a matrix with skew-symmetric off-diagonal part, as given by Eq. (2.10). This will be addressed in detail in Chapters 3, 4 and 5, but for now it is designated as $\mathbb{A}\tilde{v}^{(1)}$ which, for consistency, must be an inner 1−form defined on the dual edges. Here, $\mathbb{A}$ is the discrete version of A and is encoded as a square matrix of dimension $N_e \times N_e$. Note that $\mathbb{A}$ is not a topological operator and can be depended on the metric. Furthermore, the inner 1−form $\tilde{v}^{(1)}$ is the discrete representation of the depth-averaged velocity vector field $\mathbf{u}$ integrated along the dual edges and is given as a column vector of size N_e.

Putting this all together, the semi-discretization of Eq. (2.2) reads

$$\frac{d\tilde{\gamma}^{(1)}}{dt} + \mathbb{A}\tilde{v}^{(1)} = -g\,\overline{\tilde{\eta}^{(0)}} \odot \tilde{\mathbb{D}}^0\tilde{\zeta}^{(0)} \tag{2.20}$$

At this point we observe that Eqs. (2.19) and (2.20), with the exception of the second term $\mathbb{A}\tilde{v}^{(1)}$, are *exact* in the sense that they are independent of the metric. However, there are more unknowns than equations. Five discrete forms can be distinguished of which two are designated as the primary unknowns of the governing equations, namely, the pressure $\tilde{\eta}^{(0)}$ on the dual vertices and the mass flux $\phi^{(n-1)}$ on the primal faces. (Note that the water level $\tilde{\zeta}^{(0)}$ can immediately be derived from $\tilde{\eta}^{(0)}$.) The other three discrete forms are $\nu^{(n)}$, $\tilde{\gamma}^{(1)}$ and $\tilde{v}^{(1)}$. To close the system of equations it is necessary to relate the latter three discrete forms to the prognostic variables. Such relations are commonly called constitutive

relations and are established by making use of the Hodge star matrices. Note that they require the notion of metric. (In physics, the concept of constitutive relations provides a hypothesized relationship between two physical quantities related to a substance and so to make equations governing physical laws solvable. Constitutive equations generally depend on the physical behavior of a particular material and are therefore approximative in nature.)

Application of the Hodge stars

Based on the de Rham complex diagram of Figure 2.5 we can relate discrete forms defined on the dual mesh to those on the primal mesh, or vice versa, using the Hodge star matrices. We treat the discrete forms $\tilde{\gamma}^{(1)}$, $\nu^{(n)}$ and $\tilde{v}^{(1)}$ in turn.

Using the matrix $\mathbb{H}^{n-1}$ we define the constitutive mapping from the volumetric (or mass) flux per unit cross area (given in units of $\mathrm{m}^3{\cdot}\mathrm{s}^{-1}{\cdot}\mathrm{m}^{-2}$) integrated over a primal face (in m^2) to the depth-integrated velocity circulation along the dual edge (with units of $\mathrm{m}{\cdot}\mathrm{m}{\cdot}\mathrm{s}^{-1}{\cdot}\mathrm{m}$) as

$$\tilde{\gamma}^{(1)} = \mathbb{H}^{n-1}\, \phi^{(n-1)}$$

Recall that matrix $\mathbb{H}^{n-1}$ is invertible if the computational mesh is well centered.

Another discrete constitutive equation takes the following form

$$\nu^{(n)} = \tilde{\mathbb{H}}^{0}\, \tilde{\eta}^{(0)}$$

with $\tilde{\mathbb{H}}^{0}$ relating the pressure head (measured in units of meters) to the volume of the water column (expressed in m^3). Since this matrix is always invertible (each polygonal cell has non-zero area), we will use its inverse, that is, the primal Hodge matrix $\mathbb{H}^{n}$.

Next, we continue by specifying the depth-averaged velocity $\tilde{v}^{(1)}$. We first define an outer depth-averaged velocity $(n-1)-$form $v^{(n-1)}$ according to

$$v^{(n-1)} = \phi^{(n-1)} \odot \left(\overline{\tilde{\eta}^{(0)}}\right)^{-1} \tag{2.21}$$

and its dual is then given by

$$\tilde{v}^{(1)} = \mathbb{H}^{n-1}\, v^{(n-1)}$$

As will become apparent later on, Eq. (2.21) is a requirement to fulfill discrete energy conservation.

With the above discussed approximations, we arrive at the following semi-discrete system of equations written in terms of the unknowns $\tilde{\eta}^{(0)}$ and $\phi^{(n-1)}$

$$\frac{d\tilde{\eta}^{(0)}}{dt} + \mathbb{H}^{n}\, \mathbb{D}^{n-1}\phi^{(n-1)} = 0 \tag{2.22}$$

$$\mathbb{H}^{n-1}\frac{d\phi^{(n-1)}}{dt} + \mathbb{A}\, \left(\overline{\tilde{\eta}^{(0)}}\right)^{-1} \odot \mathbb{H}^{n-1}\, \phi^{(n-1)} = -g\, \overline{\tilde{\eta}^{(0)}} \odot \tilde{\mathbb{D}}^{0}\tilde{\zeta}^{(0)} \tag{2.23}$$

This obtained result is essentially a manifestation of the staggered Arakawa C-grid type discretization. Eqs. (2.22)$-$(2.23) represent a suitable basis for the development of the

various SWASH discretization methods. As such, it can be applied to simplicial meshes including triangular meshes (see Chapter 5) and to cubical meshes including rectilinear grids (Chapter 3) and curvilinear grids (Chapter 4).

The compatible discretizations of the `grad` and `div` operators on primal-dual meshes is key to developing a physically consistent and stable method. With this unique feature, the conservation of the mass and the total energy are satisfied within round-off errors (see below).

As for the discretization of the divergence (or advective) transport term $\nabla \cdot (\mathbf{q} \otimes \mathbf{u})$, that is, $\mathbb{A}\tilde{v}^{(1)}$, it is useful to note that it necessarily constitutes a discretization error. Although it is desirable to make this discretization energy conserving, as will be discussed below, there are applications that require some form of dissipation such as, for example, the propagation of bores and wave breaking. The usual approach is to introduce energy dissipation implicitly through the upwind approximation of the divergence term. As demonstrated by e.g., [73, 75], this numerical treatment allows an accurate regularization of the shock waves while stabilizing the semi-discretization. This will be further discussed in Chapter 3.

Energy conservation

Here we proof that the system of equations (2.22)−(2.23) is a Hamiltonian system. The discrete version of the total energy of the system is given by

$$\mathcal{H} = \mathcal{H}_{\text{kin}} + \mathcal{H}_{\text{pot}} = \frac{1}{2}\langle \upsilon^{(n-1)}, \phi^{(n-1)} \rangle_{\mathbb{H}^{n-1}} + \frac{1}{2} g \langle \tilde{\eta}^{(0)}, \tilde{\eta}^{(0)} \rangle_{\tilde{\mathbb{H}}^0}$$

and is well defined, provided that the discrete inner products are positive definite and symmetric. We will regularly use the algebraic properties of these inner products as outlined in Section 2.5.8. Let us consider the two contributions of the discrete Hamiltonian separately.

By analogy with its continuous equivalent, the rate of change of the discrete $\mathcal{H}_{\text{kin}}$ reads

$$\frac{d\mathcal{H}_{\text{kin}}}{dt} = -\langle \kappa^{(n)}, \frac{d\tilde{\eta}^{(0)}}{dt} \rangle_{\mathbb{H}^n} + \langle \upsilon^{(n-1)}, \frac{d\tilde{\gamma}^{(1)}}{dt} \rangle_{\mathbb{H}^{n-1}}$$

with

$$\kappa^{(n)} = \frac{1}{2}\langle \tilde{v}^{(1)}, \upsilon^{(n-1)} \rangle_{\mathbb{H}^{n-1}}$$

defined as the discrete kinetic energy (as volume form). Now, one can deduce the following result from the first term of the right-hand side

$$\langle \kappa^{(n)}, \frac{d\tilde{\eta}^{(0)}}{dt} \rangle_{\mathbb{H}^n} = \frac{1}{2}\langle \upsilon^{(n-1)}, \tilde{v}^{(1)} \odot \frac{d\tilde{\eta}^{(0)}}{dt} \rangle_{\mathbb{H}^{n-1}}$$

By substituting this expression and Eq. (2.20), we obtain the following result

$$\frac{d\mathcal{H}_{\text{kin}}}{dt} = -\frac{1}{2}\langle \upsilon^{(n-1)}, \tilde{v}^{(1)} \odot \frac{d\tilde{\eta}^{(0)}}{dt} \rangle_{\mathbb{H}^{n-1}} - \langle \upsilon^{(n-1)}, \mathbb{A}\tilde{v}^{(1)} \rangle_{\mathbb{H}^{n-1}} - \langle \upsilon^{(n-1)}, g\,\overline{\tilde{\eta}^{(0)}} \odot \tilde{\mathbb{D}}^0 \tilde{\zeta}^{(0)} \rangle_{\mathbb{H}^{n-1}}$$

and subsequently using Eq. (2.21), we have

$$\frac{d\mathcal{H}_{\text{kin}}}{dt} = -\frac{1}{2}\langle \upsilon^{(n-1)}, \tilde{v}^{(1)} \odot \frac{d\tilde{\eta}^{(0)}}{dt}\rangle_{\mathbb{H}^{n-1}} - \langle \upsilon^{(n-1)}, \mathbb{A}\tilde{v}^{(1)}\rangle_{\mathbb{H}^{n-1}} - \langle \phi^{(n-1)}, g\,\tilde{\mathbb{D}}^0\tilde{\eta}^{(0)}\rangle_{\mathbb{H}^{n-1}} \quad (2.24)$$

Here, we have assumed a uniform bed. Note that the definition of $\upsilon^{(n-1)}$ given by Eq. (2.21) is required to arrive at the last term of Eq. (2.24).

Next, we aim to expand the second term of Eq. (2.24). Let us denote the off-diagonal part of matrix $\mathbb{A}$ by $\mathbb{C}$. We assume a proper discretization of $\mathbb{A}$ so that matrix $\mathbb{C}$ is skew-symmetric. Based on Eq. (2.10) the discretization of the second term is given by

$$\langle \upsilon^{(n-1)}, \mathbb{A}\tilde{v}^{(1)}\rangle_{\mathbb{H}^{n-1}} = \frac{1}{2}\langle \upsilon^{(n-1)}, \mathbb{C}\tilde{v}^{(1)}\rangle_{\mathbb{H}^{n-1}} - \frac{1}{2}\langle \upsilon^{(n-1)}, \mathbb{C}^{\mathsf{T}}\tilde{v}^{(1)}\rangle_{\mathbb{H}^{n-1}} - \frac{1}{2}\langle \upsilon^{(n-1)}, \frac{d\tilde{\eta}^{(0)}}{dt} \odot \tilde{v}^{(1)}\rangle_{\mathbb{H}^{n-1}}$$

The sum of the first two terms of the right-hand side reduces to

$$\langle \upsilon^{(n-1)}, \mathbb{C}\tilde{v}^{(1)}\rangle_{\mathbb{H}^{n-1}} = \langle \upsilon^{(n-1)}, \mathbb{C}\upsilon^{(n-1)}\rangle_{\mathbb{H}^{n-1}} = 0$$

by virtue of the skew-symmetry property of $\mathbb{C}$, so that

$$\langle \upsilon^{(n-1)}, \mathbb{A}\tilde{v}^{(1)}\rangle_{\mathbb{H}^{n-1}} = -\frac{1}{2}\langle \upsilon^{(n-1)}, \frac{d\tilde{\eta}^{(0)}}{dt} \odot \tilde{v}^{(1)}\rangle_{\mathbb{H}^{n-1}}$$

Then substitution of this result into Eq. (2.24) yields

$$\frac{d\mathcal{H}_{\text{kin}}}{dt} = -\langle \phi^{(n-1)}, g\,\tilde{\mathbb{D}}^0\tilde{\eta}^{(0)}\rangle_{\mathbb{H}^{n-1}}$$

For the second contribution, the discrete analogue of the rate of change of $\mathcal{H}_{\text{pot}}$ is given by

$$\frac{d\mathcal{H}_{\text{pot}}}{dt} = \langle g\tilde{\eta}^{(0)}, \frac{d\nu^{(n)}}{dt}\rangle_{\tilde{\mathbb{H}}^0} = -\langle g\tilde{\eta}^{(0)}, \mathbb{D}^{n-1}\phi^{(n-1)}\rangle_{\tilde{\mathbb{H}}^0}$$

Finally, the rate of change of the discrete Hamiltonian reads

$$\begin{aligned}
\frac{d\mathcal{H}}{dt} = \frac{d\mathcal{H}_{\text{kin}}}{dt} + \frac{d\mathcal{H}_{\text{pot}}}{dt} &= -\langle \phi^{(n-1)}, g\,\tilde{\mathbb{D}}^0\tilde{\eta}^{(0)}\rangle_{\mathbb{H}^{n-1}} - \langle g\tilde{\eta}^{(0)}, \mathbb{D}^{n-1}\phi^{(n-1)}\rangle_{\tilde{\mathbb{H}}^0} \\
&= -\left[\langle \phi^{(n-1)}, g\,\tilde{\mathbb{D}}^0\tilde{\eta}^{(0)}\rangle_{\mathbb{H}^{n-1}} + \langle g\,\left(\mathbb{D}^{n-1}\right)^{\mathsf{T}}\tilde{\eta}^{(0)}, \phi^{(n-1)}\rangle_{\tilde{\mathbb{H}}^1}\right] \\
&= -\left[\langle \phi^{(n-1)}, g\,\tilde{\mathbb{D}}^0\tilde{\eta}^{(0)}\rangle_{\mathbb{H}^{n-1}} - \langle g\,\tilde{\mathbb{D}}^0\tilde{\eta}^{(0)}, \phi^{(n-1)}\rangle_{\tilde{\mathbb{H}}^1}\right] \\
&= 0
\end{aligned}$$

where we have use the fact that the discrete gradient is minus the adjoint of the discrete divergence.

Chapter 3

Mimetic discretizations of inviscid shallow water equations for Cartesian meshes

This chapter is yet empty. The following link is left here to give an idea of what the scope of this chapter will look like: SWASH − 2D structured grid.

Chapter 4

Mimetic discretizations of inviscid shallow water equations for curvilinear grids

This chapter is under preparation.

Chapter 5

Covolume method for discretization of inviscid shallow water equations on unstructured triangular meshes

This chapter is yet empty. The following link is left here to give an idea of what the content of this chapter will look like: SWASH − 2D unstructured triangular mesh.

Chapter 6

Three-dimensional shallow water equations

This chapter is yet empty. The following link is left here to give an idea of what the content of this material will look like: SWASH − sigmal layers.

Chapter 7

Numerical approaches

This chapter is under preparation.

Chapter 8

Implementation of boundary conditions

This chapter is under preparation.

Chapter 9

Iterative solvers

9.1 Strongly Implicit Procedure (SIP)

We want to solve the following linear system of equations

$$A\,\vec{N} = \vec{b} \tag{9.1}$$

where A is some non-symmetric penta-diagonal matrix, $\vec{N}$ is the wave action vector to be solved and $\vec{b}$ contains source terms and boundary values.

The basis for the SIP method (Stone, 1968; Ferziger and Perić, 1999) lies in the observation that an LU decomposition is an excellent general purpose solver, which unfortunately cannot take advantage of the sparseness of a matrix. Secondly, in an iterative method, if the matrix $M = LU$ is a good approximation to the matrix A, rapid convergence results. These observations lead to the idea of using an approximate LU factorization of A as the iteration matrix M, i.e.:

$$M = L\,U = A + K \tag{9.2}$$

where L and U are both sparse and K is small. For non-symmetric matrices the incomplete LU (ILU) factorisation gives such an decomposition but unfortunately converges rather slowly. In the ILU method one proceeds as in a standard LU decomposition. However, for every element of the original matrix A that is zero the corresponding elements in L or U is set to zero. This means that the product of LU will contain more nonzero diagonals than the original matrix A. Therefore the matrix K must contain these extra diagonals as well if Eq. (9.2) is to hold.

Stone reasoned that if the equations approximate an elliptic partial differential equation the solution can be expected to be smooth. This means that the unknowns corresponding to the extra diagonals can be approximated by interpolation of the surrounding points. By allowing K to have more non zero entries on all seven diagonals and using the interpolation mentioned above the SIP method constructs an LU factorization with the property that for a given approximate solution ϕ the product $K\phi \approx 0$ and thus the iteration matrix M is close to A by relation (9.2).

To solve the system of equations the following iterations is performed, starting with an initial guess for the wave action vector $\vec{N}^0$ an iteration is performed solving:

$$U\,\vec{N}^{s+1} = L^{-1}\,K\,\vec{N}^{s} + L^{-1}\,\vec{b} \tag{9.3}$$

Since the matrix U is upper triangular this equation is efficiently solved by back substitution. An essential property which makes the method feasible is that the matrix L is easily invertible. This iterative process is repeated $s = 0, 1, 2, ...$ until convergence is reached.

Chapter 10

Parallel implementation aspects

This chapter is under preparation.

Bibliography

[1] A. Arakawa. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I. *J. Comput. Phys.*, 1:119–143, 1966.

[2] A. Arakawa and V. R. Lamb. Computational design of the basic dynamical processes of the UCLA general circulation model. *Methods in Computational Physics: Advances in Research and Applications*, 17:173–265, 1977.

[3] A. Arakawa and V. R. Lamb. A potential enstrophy and energy conserving scheme for the shallow water equations. *Mon. Weather Rev.*, 109:18–36, 1981.

[4] P. B. Bochev and J. M. Hyman. Principles of mimetic discretizations of differential operators. In D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, and M. Shashkov, editors, *Compatible Spatial Discretizations. The IMA Volumes in Mathematics and its Applications, vol 142*, pages 89–120, New York, NY, 2006. Springer.

[5] L. Bonaventura and T. Ringler. Analysis of discrete shallow-water models on geodesic delaunay grids with C-type staggering. *Monthly Weather Review*, pages 2351–2373, 2005.

[6] W. Boscheri, M. Dumbser, M. Ioriatti, I. Peshkov, and E. Romenski. A structure-preserving staggered semi-implicit finite volume scheme for continuum mechanics. *J. Comput. Phys.*, 424, 2021. Article 109866.

[7] V. Casulli. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *J. Comput. Phys.*, 86:56–74, 1990.

[8] V. Casulli and R. A. Walters. An unstructured grid, three-dimensional model based on the shallow water equations. *Int. J. Numer. Meth. Fluids*, 32:331–348, 2000.

[9] V. Casulli and P. Zanolli. Semi-implicit numerical modeling of nonhydrostatic free-surface flows for environmental problems. *Math. Comput. Modell.*, 36:1131–1149, 2002.

[10] J. C. Cavendish, C. A. Hall, and T. A. Porsching. A complementary volume approach for modeling three-dimensional Navier-Stokes equations using dual Delaunay/Voronoi

tessellations. *International Journal of Numerical Methods for Heat and Fluid Flow*, 4:329–345, 1994.

[11] C. J. Cotter and J. Shipton. Mixed finite elements for numerical weather prediction. *J. Comput. Phys.*, 231:7076–7091, 2012.

[12] C. J. Cotter and J. Thuburn. A finite element exterior calculus framework for the rotating shallow-water equations. *J. Comput. Phys.*, 257:1506–1526, 2014.

[13] P. J. Dellar. Common Hamiltonian structure of the shallow water equations with horizontal temperature gradients and magnetic fields. *Phys. Fluids*, 15:292–297, 2003.

[14] M. Desbrun, A. N. Hirani, M. Leok, and J. E. Marsden. Discrete exterior calculus. *arXiv:math/0508341v2 [math.DG]*, pages 1–53, 2005.

[15] M. Desbrun, E. Kanso, and Y. Tang. Discrete differential forms for computational modeling. In A. I. Bobenko, P. Schröder, J. M. Sullivan, and G. M. Ziegler, editors, *Discrete Differential Geometry*, volume 38, pages 287–323. Birkhäuser Verlag, Basel, Switzerland, 2008.

[16] F. N. Felten and T. S. Lund. Kinetic energy conservation issues associated with the collocated mesh scheme for incompressible flow. *J. Comput. Phys.*, 215:465–484, 2006.

[17] O. B. Fringer, M. Gerritsen, and R. L. Street. An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal ocean simulator. *Ocean Modell.*, 14:139–173, 2006.

[18] C. A. Hall, J. C. Cavendish, and W. H. Frey. The dual variable method for solving fluid flow difference equations on Delaunay triangulations. *Comput. Fluids*, 20:145–164, 1991.

[19] A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2001.

[20] M. Herzfeld, D. Engwirda, and F. Rizwi. A coastal unstructured model using Voronoi meshes and C-grid staggering. *Ocean Modell.*, 148, 2020. Article 101599.

[21] A. N. Hirani. *Discrete Exterior Calculus*. Ph.D. thesis, California Institute of Technology, Pasadena, California, USA, 2003.

[22] A. N. Hirani, K. B. Nakshatrala, and J. H. Chaudhry. Numerical method for Darcy flow derived using discrete exterior calculus. *Int. J. Comput. Meth. Engng. Sci. Mech.*, 16:151–169, 2015.

[23] J. M. Hyman and M. Shashkov. Natural discretizations for the divergence, gradient, and curl on logically rectangular grids. *Computers Math. Applic.*, 33:81–104, 1997.

[24] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. In *AIAA 14th Fluid and Plasma Dynamic Conference*, pages 1981–1259, Palo Alto, CA, 1981.

[25] H. W. J. Kernkamp, A. van Dam, G. S. Stelling, and E. D. de Goede. Efficient scheme for the shallow water equations on unstructured grids with application to the Continental Shelf. *Ocean Dyn.*, 61:1175–1188, 2011.

[26] P. Korn and S. Danilov. Elementary dispersion analysis of some mimetic discretizations on triangular C-grids. *J. Comput. Phys.*, 330:156–172, 2017.

[27] P. Korn and L. Linardakis. A conservative discretization of the shallow-water equations on triangular grids. *J. Comput. Phys.*, 375:871–900, 2018.

[28] S. C. Kramer and G. S. Stelling. A conservative unstructured scheme for rapidly varied flows. *Int. J. Numer. Meth. Fluids*, 58:183–212, 2008.

[29] J. Kreeft and M. Gerritsma. Mixed mimetic spectral element method for Stokes flow: a pointwise divergence-free solution. *J. Comput. Phys.*, 240:284–309, 2013.

[30] J. J. Leendertse. *Aspects of a computational model for long-period water-wave propagation.* Ph.D. thesis, RM 5294-PR, Rand Corporation, Santa Monica, USA, 1967.

[31] D. Y. LeRoux, V. Rostand, and B. Pouliot. Analysis of numerically induced oscillations in 2D finite-element shallow-water models. Part I: inertia-gravity waves. *SIAM J. Sci. Comput.*, 29:331–360, 2007.

[32] D. K. Lilly. On the computational stability of numerical solutions of time-dependent non-linear geophysical fluid dynamics problems. *Mon. Weather Rev.*, 93:11–26, 1965.

[33] K. Lipnikov, G. Manzini, and M. Shashkov. Mimetic finite difference method. *J. Comput. Phys.*, 257:1163–1227, 2014.

[34] T. A. Manteuffel and A. B. White Jr. The numerical solution of second-order boundary value problems on nonuniform meshes. *Math. Comp.*, 47:511–535, 1986.

[35] C. Mattiussi. An analysis of finite volume, finite element, and finite difference methods using some concepts from algebraic topology. *J. Comput. Phys.*, 133:289–309, 1997.

[36] C. Mattiussi. A reference discretization strategy for the numerical solution of physical field problems. *Advances in Imaging and Electron Physics*, 121:143–279, 2002.

[37] M. S. Mohamed, A. N. Hirani, and R. Samtaney. Comparison of discrete Hodge star operators for surfaces. *Computer-Aided Design*, 78:118–125, 2016.

[38] M. S. Mohamed, A. N. Hirani, and R. Samtaney. Discrete exterior calculus discretization of incompressible Navier-Stokes equations over surface simplicial meshes. *J. Comput. Phys.*, 312:175–191, 2016.

[39] M. S. Mohamed, A. N. Hirani, and R. Samtaney. Numerical convergence of discrete exterior calculus on arbitrary surface meshes. *Int. J. Comput. Meth. Engng. Sci. Mech.*, 19:194–206, 2018.

[40] Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin. Fully conservative higher order finite difference schemes for incompressible flow. *J. Comput. Phys.*, 143:90–124, 1998.

[41] P. J. Morrison. Poisson brackets for fluids and plasmas. In M. Tabor and Y. M. Treve, editors, *Mathematical Methods in Hydrodynamics and Integrability in Dynamical Systems*, pages 13–46, 1982. AIP Conf. Proc., no. 88.

[42] J. R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley Publishing Company, California, USA, 1984.

[43] T. Needham. *Visual Differential Geometry and Forms*. Princeton University Press, 2021.

[44] R. A. Nicolaides. Direct discretization of planar div-curl problems. *SIAM J. Numer. Anal.*, 29:32–56, 1992.

[45] R. A. Nicolaides. The covolume approach to computing incompressible flow. In M. D. Gunzburger and R. A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics*, page 295, Cambridge, UK, 1993. Cambridge Univ. Press.

[46] A. Palha and M. Gerritsma. A mass, energy, enstrophy and vorticity conserving (MEEVC) mimetic spectral element discretization for the 2D incompressible Navier-Stokes equations. *J. Comput. Phys.*, 328:200–220, 2017.

[47] B. Perot. Conservation properties of unstructured staggered mesh schemes. *J. Comput. Phys.*, 159:58–89, 2000.

[48] N. A. Phillips. An example of non-linear computational instability. In B. Bolin, editor, *The Atmosphere and the Sea in Motion*, pages 501–504, New York, 1959. Rockefeller Institute Press.

[49] S. A. Piacsek and G. P. Williams. Conservative properties of convection difference schemes. *J. Comput. Phys.*, 6:392–405, 1970.

[50] T. Ringler, J. Thuburn, J. Klemp, and W. Skamarock. A unified approach to energy conservation and potential vorticity dynamics on arbitrarily structured C-grids. *J. Comput. Phys.*, 229:3065–3090, 2010.

[51] T. G. Shepherd. Symmetries, conservation laws, and Hamiltonian structure in geophysical fluid dynamics. *Advances in Geophysics*, 32:287–338, 1990.

[52] A. Staniforth and J. Thuburn. Horizontal grids for global weather and climate prediction models: a review. *Q. J. R. Meteorol. Soc.*, 138:1–26, 2012.

[53] P. K. Stansby. Semi-implicit finite volume shallow-water flow and solute transport solver with k-ε turbulence model. *Int. J. Numer. Meth. Fluids*, 25:285–313, 1997.

[54] S. Steinberg. The accuracy of numerical models for continuum problems. In H. Bulgak and C. Zenger, editors, *Error Control and Adaptivity in Scientific Computing*, pages 299–323, Dordrecht, The Netherlands, 1999. Springer.

[55] G. S. Stelling. *On the construction of computational methods for shallow water flow problems*. Ph.D. thesis, Rijkswaterstaat communications no. 35, 1984.

[56] G. S. Stelling and S. P. A. Duinmeijer. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. *Int. J. Numer. Meth. Fluids*, 43:1329–1354, 2003.

[57] G.S. Stelling and M. Zijlema. Numerical modeling of wave propagation, breaking and run-up on a beach. In B. Koren and C. Vuik, editors, *Advanced computational methods in science and engineering*, volume 71, pages 373–401, Springer, Heidelberg, 2010. Lecture Notes in Computational Science and Engineering.

[58] B. Strand. Summation by parts for finite difference approximations for d/dx. *J. Comput. Phys.*, 110:47–67, 1994.

[59] G. R. Stuhne and W. R. Peltier. A robust unstructured grid discretization for 3-dimensional hydrostatic flows in spherical geometry: A new numerical structure for ocean general circulation modeling. *J. Comput. Phys.*, 213:704–729, 2006.

[60] J. Thuburn. Numerical wave propagation on the hexagonal C-grid. *J. Comput. Phys.*, 227:5836–5858, 2008.

[61] J. Thuburn and C. J. Cotter. A framework for mimetic discretization of the rotating shallow-water equations on arbitrary polygonal grids. *SIAM J. Sci. Comput.*, 34:B203–B225, 2012.

[62] J. Thuburn, T. Ringler, J. Klemp, and W. Skamarock. Numerical representation of geostrophic modes on arbitrarily structured C-grids. *J. Comput. Phys.*, 228:8321–8335, 2009.

[63] E. Tonti. Why starting from differential equations for computational physics? *J. Comput. Phys.*, 257:1260–1290, 2014.

[64] P. van Beek, R.R.P. van Nooyen, and P. Wesseling. Accurate discretization of gradients on non-uniform curvilinear staggered grids. *J. Comput. Phys.*, 117:364–367, 1995.

[65] A. E. P. Veldman and K.-W. Lam. Symmetry-preserving upwind discretization of convection on non-uniform grids. *Appl. Numer. Math.*, 58:1881–1891, 2008.

[66] R. W. C. P. Verstappen and A. E. P. Veldman. Spectro-consistent discretization of Navier-Stokes: a challenge to RANS and LES. *J. Engng. Math.*, 34:163–179, 1998.

[67] R. W. C. P. Verstappen and A. E. P. Veldman. Symmetry-preserving discretization of turbulent flow. *J. Comput. Phys.*, 187:343–368, 2003.

[68] J. Von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *J. Appl. Phys.*, 21:232–237, 1950.

[69] P. Wesseling, A. Segal, J.J.I.M. van Kan, C.W. Oosterlee, and C.G.M. Kassels. Finite volume discretization of the incompressible Navier-Stokes equations in general coordinates on staggered grids. *Comput. Fluid Dyn. J.*, 1:27–33, 1992.

[70] H. C. Yee and P. K. Sweby. Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations. II. Global asymptotic behavior of time discretizations. *Int. J. Comput. Fluid Dyn.*, 4:219–283, 1995.

[71] H. C. Yee, P. K. Sweby, and D. F. Griffiths. Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations. I. The dynamics of time discretization and its implications for algorithm development in computational fluid dynamics. *J. Comput. Phys.*, 97:249–310, 1991.

[72] M. Zijlema. TRIWAQ - three-dimensional shallow water flow model. Version 1.1. Technical documentation RKZ-438, National Institute for Coastal and Marine Management, The Hague, The Netherlands, 1998. SIMONA 99-01.

[73] M. Zijlema. The role of the Rankine-Hugoniot relations in staggered finite difference schemes for the shallow water equations. *Comput. Fluids*, 192, 2019. Article 104274.

[74] M. Zijlema. Computation of free surface waves in coastal waters with SWASH on unstructured grids. *Comput. Fluids*, 213, 2020. Article 104751.

[75] M. Zijlema. On the efficiency of staggered C-grid discretization for the inviscid shallow water equations from the perspective of nonstandard calculus. *Mathematics*, 10:1387–, 2022.

[76] M. Zijlema, A. Segal, and P. Wesseling. Finite volume computation of incompressible turbulent flows in general co-ordinates on staggered grids. *Int. J. Numer. Meth. Fluids*, 20:621–640, 1995.

[77] M. Zijlema and G. S. Stelling. Further experiences with computing non-hydrostatic free-surface flows involving water waves. *Int. J. Numer. Meth. Fluids*, 48:169–197, 2005.

[78] M. Zijlema and G. S. Stelling. Efficient computation of surf zone waves using the nonlinear shallow water equations with non-hydrostatic pressure. *Coast. Engng.*, 55:780–790, 2008.

[79] M. Zijlema, G. S. Stelling, and P. B. Smit. SWASH: an operational public domain code for simulating wave fields and rapidly varied flows in coastal waters. *Coast. Engng.*, 58:992–1012, 2011.