

Mikael Myyrä

**Discrete Exterior Calculus and Exact Controllability for
Time-Harmonic Acoustic Wave Simulation**

Master's Thesis in Information Technology

August 23, 2023

University of Jyväskylä

Faculty of Information Technology

Author: Mikael Myyrä

Contact information: `mikael.b.myyra@jyu.fi`

Supervisor: Sanna Mönkölä, Tuomo Rossi, Jonni Lohi, Tytti Saksa

Title: Discrete Exterior Calculus and Exact Controllability for Time-Harmonic Acoustic Wave Simulation

Työn nimi: Diskreetti ulkoinen laskenta ja kontrollimenetelmä aikaharmonisessa akustiikka-simulaatiossa

Project: Master's Thesis

Study line: Mathematical modelling in science and decision analytics

Page count: 69+0

Abstract: Discrete exterior calculus (DEC) is a discretization method for differential equations which preserves important geometric properties of physical models and produces computationally efficient algorithms. We investigate ways to improve the performance of DEC in the context of time-periodic solutions to acoustic wave scattering problems by using operators purpose-built for harmonic problems and an optimization approach based on exact controllability. The harmonic operators are found to improve accuracy significantly, while the controllability method is found to be a good choice only in problems with highly non-convex geometry. Computation mesh quality is identified as a key issue in the accuracy and stability of these methods.

Keywords: discrete exterior calculus, exact controllability, time-harmonic, acoustics, simulation

Suomenkielinen tiivistelmä: Diskreetti ulkoinen laskenta (engl. discrete exterior calculus, DEC) on differentiaaliyhtälöiden ratkaisemiseen soveltuva diskreetointimenetelmä, joka säilyttää tiettyjä fysikaalisten mallien geometrisia ominaisuuksia ja tuottaa laskennallisesti tehokkaita algoritmeja. Tutkielmassa tarkastellaan keinoja tehostaa DEC:n suorituskykyä aikaharmonisten ratkaisujen selvittämisessä akustisten aaltojen sirontaa kuvaaville

tehtäville. Tähän käytetään harmonisille malleille suunniteltuja operaattoreita ja eksaktiin kontrolloitavuuteen perustuvaa optimointimenetelmää. Toteutetuissa kokeiluissa harmoniset operaattorit lisäävät menetelmän tarkkuutta huomattavasti, kun taas kontrollimenetelmä osoittautuu hyväksi valinnaksi vain tehtävissä, joiden geometria on huomattavan epäkonvek-sia. Laskentaverkon laadulla on suuri merkitys tutkittujen menetelmien tarkkuudessa ja stabiilisuudessa.

Avainsanat: diskreetti ulkoinen laskenta, kontrollimenetelmä, aikaharmoninen, akustiikka, simulointi

List of Figures

| | |
|--|----|
| Figure 1. Exterior products of 1-forms visualized as volumes. | 7 |
| Figure 2. Boundary of a triangle with inconsistently oriented faces. | 14 |
| Figure 3. Discrete exterior derivative visualized on an oriented simplicial complex. | 17 |
| Figure 4. A mesh and its dual. | 18 |
| Figure 5. 2D divergence as the exterior derivative of a rotated form. | 24 |
| Figure 6. The domain Ω and boundaries Γ_{ext} and Γ_{sca} of a scattering model. | 28 |
| Figure 7. Internal and external radii of a regular and irregular triangle. | 35 |
| Figure 8. A plane wave propagating upwards in a square domain. | 47 |
| Figure 9. Maximum and average error of the simulated plane wave in relation to maximum edge length of the mesh. | 48 |
| Figure 10. Effect of angular velocity on the exact timestep coefficient. | 49 |
| Figure 11. Mesh used to examine the CFL condition, overlaid with its refinements. | 50 |
| Figure 12. Maximum stable value of Δt plotted against the shortest edge length. | 51 |
| Figure 13. Scatterer, case 1: square. | 52 |
| Figure 14. Scatterer, case 2: eight-pointed star. | 53 |
| Figure 15. Scatterer, case 3: twelve-pointed star. | 53 |
| Figure 16. Scatterer, case 4: diamonds. | 54 |

List of Tables

| | |
|---|---|
| Table 1. Basis forms in \mathbb{R}^3 | 8 |
| Table 2. Basis forms in \mathbb{R}^2 | 8 |

Contents

| | | |
|-------|---|----|
| 1 | INTRODUCTION | 1 |
| 2 | DISCRETE EXTERIOR CALCULUS | 3 |
| 2.1 | Differential forms and the continuous exterior calculus | 3 |
| 2.1.1 | Differential forms | 3 |
| 2.1.2 | Proxy vectors | 5 |
| 2.1.3 | Basis forms | 5 |
| 2.1.4 | Exterior product | 5 |
| 2.1.5 | Higher-dimensional basis forms | 8 |
| 2.1.6 | Hodge star | 9 |
| 2.1.7 | Exterior derivative | 10 |
| 2.1.8 | Advantages of exterior calculus | 11 |
| 2.2 | Computation mesh | 12 |
| 2.2.1 | Simplices | 12 |
| 2.2.2 | Simplicial complex | 12 |
| 2.2.3 | Orientation of a simplex | 13 |
| 2.2.4 | Chains and cochains | 14 |
| 2.3 | Discrete differential forms | 14 |
| 2.3.1 | Definition of discrete forms | 15 |
| 2.3.2 | Discrete exterior derivative | 15 |
| 2.3.3 | Dual mesh | 17 |
| 2.3.4 | Discrete Hodge star | 19 |
| 2.3.5 | Interpolating discrete forms | 20 |
| 3 | MODEL | 23 |
| 3.1 | Vector equations | 23 |
| 3.2 | Exterior calculus equations | 23 |
| 3.3 | Discretization | 25 |
| 3.3.1 | Space | 25 |
| 3.3.2 | Time | 26 |
| 3.4 | Boundary conditions | 27 |
| 3.5 | Harmonic operators | 29 |
| 3.5.1 | Exact timestepping | 30 |
| 3.5.2 | Harmonic Hodge star | 31 |
| 4 | EXACT CONTROLLABILITY | 37 |
| 4.1 | Optimization problem | 37 |
| 4.2 | Computing the gradient | 38 |
| 4.2.1 | State equation | 38 |
| 4.2.2 | Gradient of control energy | 39 |
| 4.2.3 | Adjoint state method | 40 |
| 4.3 | Conjugate gradient algorithm | 42 |

| | | |
|-----|----------------------|----|
| 5 | EXPERIMENTS | 46 |
| 5.1 | Implementation | 46 |
| 5.2 | Plane wave | 46 |
| 5.3 | CFL condition | 50 |
| 5.4 | Scatterer..... | 51 |
| 5.5 | Discussion | 55 |
| 6 | CONCLUSION | 57 |
| | BIBLIOGRAPHY | 58 |

1 Introduction

Discrete exterior calculus (DEC), developed by Desbrun et al. (2005) with roots leading up to the finite-difference time domain (FDTD) method of Yee (1966) and the work of Bossavit (1998, 1999, 2000; 2001; 2005), is a discretization method for differential equations based on the exterior calculus of differential forms. It is a *mimetic* discretization method, meaning it preserves certain physically important geometric properties of the continuous model exactly (Bochev and Hyman 2006). It also provides an efficient explicit timestepping procedure requiring no solution of linear systems.

DEC has been previously applied to simulations of e.g. electromagnetics (Räbinä, Mönkölä, and Rossi 2015; Mönkölä and Rätty 2022), Darcy flow (Hirani, Nakshatrala, and Chaudhry 2015), quantum mechanics (Räbinä, Kuopanportti, et al. 2018; Kivioja et al. 2023) and fluid dynamics (Nitschke, Reuther, and Voigt 2017). Recently, it has been used to develop a theory of general classes of wave propagation problems in Räbinä, Kettunen, et al. (2018) and Rossi et al. (2021).

We apply DEC to the propagation and scattering of acoustic waves. In addition to the baseline DEC, we implement the optimized operators for harmonic simulations introduced by Räbinä (2014) in the context of electromagnetics. We also investigate ways to use DEC to obtain time-periodic solutions to scattering problems. A simple asymptotic iteration procedure is compared with an optimization approach based on the idea of exact controllability developed by Lions (1988) and applied to the computation of time-periodic solutions by Bristeau, Glowinski, and Périaux (1998). This method has been previously applied to e.g. electromagnetics (Räbinä 2014), elastodynamics (Mönkölä et al. 2008), and acoustics (Kähkönen et al. 2011). The method can be used with any forward-in-time solver, such as finite element or finite difference solvers. Räbinä (2014) is an example of usage specifically with DEC.

We implement a solver and a series of numerical experiments in the Python programming language, using the low-level DEC operations provided by the PyDEC software library (Bell and Hirani 2012) and the finite element mesh generator Gmsh (Geuzaine and Remacle 2009).

Further technical details are given in Section 5.1.

This thesis begins by explaining the mathematical theory of the methods used. An effort is made to prioritize intuitions useful in engineering applications over rigorous mathematical definitions, intended as a suitable starting point for students unfamiliar with the topic but with some experience in mathematics. Chapters 2 and 4 cover the theory of discrete exterior calculus and the exact controllability method, respectively. The simulation model used in our numerical experiments is defined in Chapter 3, and the experiments themselves are discussed in Chapter 5.

2 Discrete exterior calculus

The mathematical theory underlying discrete exterior calculus is the exterior calculus of differential forms, which provides a way to express differential equations in a way that lends itself to a straightforward and systematic method of discretization where continuous operators have direct discrete analogues. This chapter gives a brief introduction to exterior calculus followed by a description of the mesh- and cochain-based discretization of DEC. The reader is assumed to be familiar with linear algebra and vector calculus.

2.1 Differential forms and the continuous exterior calculus

This section introduces differential forms and their operations with a focus on visual intuitions and practical applications. More detail is provided in a similar presentation style by Blair Perot and Zusi (2014) and Crane et al. (2013). For a more formal and comprehensive approach to the topic, see a textbook on differential geometry such as Lee (2012) or Abraham, Marsden, and Ratiu (2012).

2.1.1 Differential forms

Notationally, a differential form looks like an *integrand*, e.g. the dx in $\int dx$ or the $5 dx dy$ in $\iint 5 dx dy$. Indeed, the things we integrate are all technically differential forms. However, this seems rather abstract. Are these objects meaningful outside of integration, and if so, what do they do? This section looks at differential forms as functions, taking inspiration from Crane et al. (2013).

Differential forms are in many ways analogous to vectors. To illustrate the similarities and differences, consider the dot product operation for two vectors in linear algebra. Expressed in matrix form, the dot product between vectors $\mathbf{a} = (a_1, \dots, a_n)^T$ and $\mathbf{b} = (b_1, \dots, b_n)^T$ is

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

The dot product measures the length of \mathbf{b} in the direction of \mathbf{a} , and to do this, \mathbf{a} is transposed into a *row vector*. This row vector can be thought of as a function that takes a vector and measures it along \mathbf{a} ,

$$\alpha(\mathbf{b}) = \mathbf{a}^T \mathbf{b}.$$

A linear function that takes a vector and produces a scalar like this is called a *covector*.

We can also measure higher-dimensional quantities. A 2-covector is a function that takes two vectors, which define a parallelogram, and computes the area of this parallelogram when projected onto a plane. Similarly, 3-covectors measure the volumes of parallelepipeds defined by three vectors, and so on for higher dimensions. Additionally, we can define 0-covectors as objects that take zero vectors as input and return a measurement, making them equivalent to scalars.

For covectors of dimension 2 and above, we also need to consider the notion of *signed* areas and volumes. Much like how the vector cross product flips its direction depending on whether its arguments come in clockwise or counterclockwise order, a covectors's output will change its sign if any two of its arguments are exchanged. The sign is a measurement of the given volume's *orientation*: clockwise or counterclockwise for 2D areas, right-handed or left-handed basis for 3D volumes, etc. This is why covectors are formally defined as *antisymmetric tensors* – they're multilinear functions on vectors (i.e. tensors) which change sign when arguments are exchanged (i.e. they are antisymmetric).

A differential k -form is a *field* of k -covectors, i.e. a function that associates a k -covector with every point in space. Somewhat confusingly, the term k -form is also commonly used to refer to individual k -covectors; a terminology we will adopt from here on. k -form in this thesis will refer to either a k -covector or a field of them depending on context.

2.1.2 Proxy vectors

There is a natural one-to-one correspondence between row and column vectors, expressed in the transpose operator. An analogous one-to-one correspondence also exists between 1-forms and vectors. The operators realizing this correspondence are called *sharp* \sharp and *flat* \flat . Sharp turns a 1-form into the corresponding vector, and flat does the reverse. In other words, given a 1-form α , $v = \alpha^\sharp$ is a vector and $v^\flat = \alpha$.

We call v a *vector proxy* of α . When performing exterior calculus computations, the inputs and results are typically interpreted and visualized as their vector proxies.

2.1.3 Basis forms

Like vectors, differential forms also form a vector space, meaning they can be expressed as linear combinations of basis vectors. In \mathbb{R}^3 , with axes labeled x , y , and z , the corresponding basis 1-forms are called dx , dy , and dz .

Higher-dimensional forms also have this vector space structure, but defining a basis for them is not quite as straightforward. The issue is that n -forms measure n -dimensional geometric objects which require n different vectors to define. Thus, to define an n -form we need a formal way to combine measurements in n different directions. The operator which does this is called the *exterior product*.

2.1.4 Exterior product

You may recall from linear algebra that the signed area of a parallelogram defined by two vectors u, v in \mathbb{R}^3 is computed by the cross product $u \times v$. Similarly, the volume of a parallelepiped defined by three vectors u, v, w is computed by the triple product $(u \times v) \cdot w$, and in general the volume spanned by n vectors v_1, \dots, v_n is the determinant

$$\det [v_1, \dots, v_n].$$

The analogous operation for differential forms is called the *exterior product* or *wedge product* \wedge , which takes a k -form α and a j -form β and produces a $(k + j)$ -form which measures

volumes spanned by α and β combined. For instance, if α and β are both 1-forms, $\alpha \wedge \beta$ is a 2-form which measures parallelograms in the plane spanned by α and β .

Depending on the dimension of forms being multiplied, the wedge product's sign may change when the order of its arguments is flipped. Specifically,

$$\alpha \wedge \beta = (-1)^{kj} \beta \wedge \alpha. \quad (2.1)$$

Additionally, the wedge product is associative, meaning

$$(\alpha \wedge \beta) \wedge \gamma = \alpha \wedge (\beta \wedge \gamma). \quad (2.2)$$

These three properties (addition of dimensions, antisymmetry, and associativity) are enough to define the wedge product abstractly in all dimensions. In practice, what this operation looks like depends on the dimension of its operands in a way that mirrors the vector operations measuring volumes. The wedge product of two 1-forms in 3D space is equivalent to the cross product on the vector proxies of the forms, and the wedge product of three 1-forms (really a 2-form and a 1-form) is equivalent to the triple product on the vector proxies. Additionally, since a 0-form is just a scalar, the wedge product of a 0-form with anything is a scalar multiplication.

To give a specific example, consider two 1-forms α, β in \mathbb{R}^3 . The wedge product $\alpha \wedge \beta$ is equivalent to $\star(\alpha^\sharp \times \beta^\sharp)^\flat$ (where the star symbol, not important yet but included for accuracy's sake, is the *Hodge star* operator covered later in Section 2.1.6). The resulting 2-form is applied to two vectors u, v as

$$\alpha \wedge \beta(u, v) = \alpha(u)\beta(v) - \alpha(v)\beta(u),$$

and the resulting value is the area of the parallelogram defined by the projections of u and v onto a plane whose basis vectors are α^\sharp and β^\sharp .

The 1-form wedge product can be visualized as combining two vectors into a single oriented parallelogram, or in a sense, two basis elements of a line into a basis element of a plane. Similarly, further wedge products add on more basis elements, producing higher-dimensional oriented volumes. This is illustrated in Figure 1.

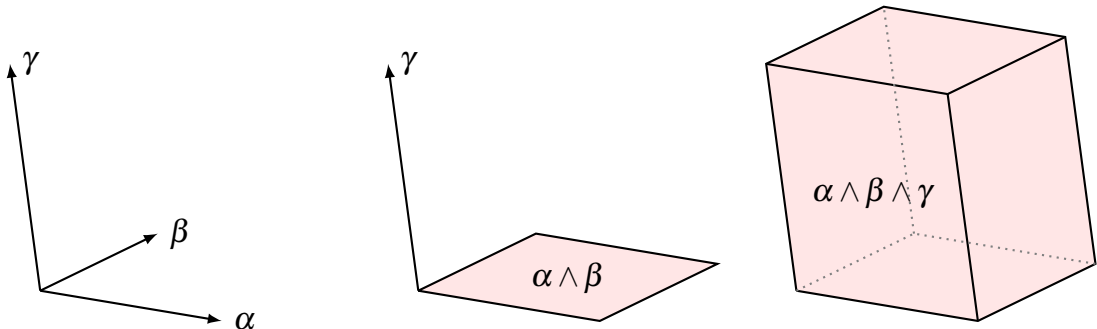


Figure 1. Exterior products of 1-forms visualized as volumes.

Thinking about an n -form as an oriented piece of an n -dimensional subspace like this is an intuition which will be useful in understanding the operations of exterior calculus.

Naturally, the dimension of a form cannot exceed that of the space it's embedded in. Thus, the wedge product in n -dimensional space is only defined for operands whose dimensions sum to n or less.

A consequence of the wedge product being defined to measure volumes (algebraically, a consequence of the antisymmetry equation (2.1)) is that the wedge product of a k -form with itself is zero when k is odd;

$$\alpha \wedge \beta = -\beta \wedge \alpha \implies \alpha \wedge \alpha = 0.$$

This is because, intuitively, two parallel vectors form a parallelogram with zero area, and in general a linearly dependent set of vectors forms a degenerate volume.

On the other hand, when k is even (including zero), antisymmetry does not apply. For 0-forms, whose wedge product is scalar multiplication, this means $\alpha \wedge \alpha = \alpha^2$. For higher-dimensional forms this only begins to apply in four dimensions (where 2-forms can be multiplied) and is therefore not relevant to our use cases.

Equipped with the exterior product, we can define a basis for higher-dimensional forms.

2.1.5 Higher-dimensional basis forms

Basis forms of dimension 2 and above are defined in terms of basis 1-forms and the exterior product.

In three dimensions, there are obviously three basis 1-forms. The basis of 2-forms also has three elements, or in other words, there are three linearly independent basis planes. This number can be deduced algebraically by taking the wedge product of all possible pairs of basis 1-forms and eliminating ones that are either zero or a multiple of another. A more intuitive geometric reasoning is that a plane in three dimensions is uniquely identified by its normal vector; therefore the space of all planes has the same dimension as the space of all vectors.

There is only one three-dimensional subspace of three-dimensional space, and thus only one basis 3-form. This is true for any n : there is only one basis n -form in n -dimensional space. This form is called the *volume form* of the space, and it behaves like a scalar in computations. Table 1 lists the basis forms in \mathbb{R}^3 .

| n | basis n -forms |
|-----|--|
| 0 | 1 |
| 1 | dx, dy, dz |
| 2 | $dx \wedge dy, dy \wedge dz, dz \wedge dx$ |
| 3 | $dx \wedge dy \wedge dz$ |

Table 1. Basis forms in \mathbb{R}^3 .

In two dimensions things are somewhat simpler, as there is only one plane, and the 2-form is the volume form. Table 2 lists the basis forms in \mathbb{R}^2 .

| n | basis n -forms |
|-----|------------------|
| 0 | 1 |
| 1 | dx, dy |
| 2 | $dx \wedge dy$ |

Table 2. Basis forms in \mathbb{R}^2 .

2.1.6 Hodge star

Consider the idea mentioned in Section 2.1.5 that a plane in three dimensions is uniquely identified by its normal vector. This is one manifestation of the general fact that a linear subspace is uniquely identified by its orthogonal complement. In the context of differential forms, an analogous statement is that there is an equal number of basis k -forms and $(n - k)$ -forms in n -dimensional space and the map from a k -form to its orthogonal $(n - k)$ -form is an isomorphism.

The isomorphism taking a k -form to the orthogonal $(n - k)$ -form with the same magnitude is an important operator called the *Hodge star* \star . It is a linear operator whose effect on the \mathbb{R}^3 basis forms is as follows:

$$\begin{aligned}\star 1 &= dx \wedge dy \wedge dz \\ \star dx &= dy \wedge dz \\ \star dy &= dz \wedge dx \\ \star dz &= dx \wedge dy \\ \star(dx \wedge dy) &= dz \\ \star(dy \wedge dz) &= dx \\ \star(dz \wedge dx) &= dy \\ \star(dx \wedge dy \wedge dz) &= 1\end{aligned}$$

Applying the star twice amounts to doing nothing at all, $\star\star\alpha = \alpha$.

In \mathbb{R}^2 this is complicated slightly by the fact that the orthogonal complement of a line is also a line, and the Hodge star on 1-forms amounts to a 90 degree rotation. Hence, for 1-forms in \mathbb{R}^2 , a star twice is a 180 degree rotation, $\star\star\alpha = -\alpha$. In general, for k -forms in n -dimensional space,

$$\star\star\alpha = (-1)^{k(n-k)}\alpha.$$

The effect of the Hodge star on the \mathbb{R}^2 basis forms is

$$\begin{aligned}\star 1 &= dx \wedge dy \\ \star dx &= dy \\ \star dy &= -dx \\ \star(dx \wedge dy) &= 1\end{aligned}$$

2.1.7 Exterior derivative

The final operations we need to define the *calculus* part of exterior calculus are differentiation and integration. Integration is exactly the procedure familiar from multivariate calculus (differential forms are integrands, after all). 1-forms can be integrated over curves, 2-forms over surfaces, 3-forms over volumes, and so on. Differentiation is performed by the *exterior derivative* \mathbf{d} , which, like the exterior product, is equivalent to several different vector operators depending on the dimension of its operand.

A common property of the exterior derivative regardless of dimension is that it increases the dimension of its operand by one. In other words, the exterior derivative of a k -form α is a $(k+1)$ -form. This, along with the *product rule*

$$\mathbf{d}(\alpha \wedge \beta) = \mathbf{d}\alpha \wedge \beta + (-1)^k \beta \wedge \mathbf{d}\alpha,$$

the property $\mathbf{d} \circ \mathbf{d} = 0$, and the definition of \mathbf{d} for 0-forms, can be used to recursively define the exterior derivative for all dimensions, but as with the wedge product, the formal definition is very abstract. An intuitive explanation of why \mathbf{d} is defined this way is outside the scope of this thesis, but a reasoning based on the *differential of a manifold* is given in Crane et al. 2013. For our purposes it will suffice to cover the equivalent vector proxy operations for each dimension of \mathbf{d} .

A 0-form f is a scalar function. Its exterior derivative is the gradient, also known as the differential,

$$\mathbf{d}f = \nabla f = df.$$

Omitting conversions with \sharp , \flat and \star for clarity, the exterior derivative of a 1-form α corresponds to the *curl* of the vector proxy,

$$\mathbf{d}\alpha \approx \nabla \times \alpha.$$

The exterior derivative of the volume form is not defined (as it would produce a form of higher dimension than the space), so in two dimensions this is the final \mathbf{d} . In three dimensions, the exterior derivative of a 2-form β is the *divergence* of the (perpendicular) vector proxy,

$$\mathbf{d}\beta \approx \nabla \cdot \beta.$$

These relationships give rise to *Stokes' theorem*, which states that for a k -dimensional domain Ω with boundary $\partial\Omega$ and a $(k-1)$ -form α ,

$$\int_{\Omega} \mathbf{d}\alpha = \int_{\partial\Omega} \alpha. \quad (2.3)$$

This theorem encapsulates several vector calculus identities such as Green's theorem and the divergence theorem. It is crucial in defining the discrete exterior calculus and will be covered in more detail in Section 2.3.2.

2.1.8 Advantages of exterior calculus

At this point it would be reasonable to ask what the point of using differential forms instead of vectors is. If every operation on differential forms has a vector equivalent, why the additional abstraction?

Indeed, in two- or three-dimensional Euclidean space, the differential form and vector representations of problems are largely equivalent and interchangeable. One of the main purposes of exterior calculus is the generalization of these concepts to *higher-dimensional* and *curved* spaces. Exterior calculus is readily defined for any dimension and enables the expression of problems on any *manifold* (generalization of 2D surfaces, 3D volumes etc.), incorporating the *metric* (measurement of distances between points) in a systematic way.

In this thesis, we restrict ourselves to Euclidean 2D space and thus don't make use of these

advantages, but exterior calculus still has something to offer. Namely, the DEC provides a straightforward way to discretize any equation expressed in exterior calculus.

2.2 Computation mesh

The discrete spatial structure at the root of the DEC is the *simplicial complex*: essentially a mesh (in the computer graphics sense) consisting of triangles in two dimensions, tetrahedra in three dimensions, or *k-simplices* in *k* dimensions. We will focus on the two-dimensional case here for illustration. For more details on the topic, see Desbrun, Kanso, and Tong (2006).

2.2.1 Simplices

A *k-simplex* is the simplest type of *k*-dimensional element in a polyhedral mesh, defined as the convex hull of $k + 1$ geometrically distinct points. In concrete terms, a 0-simplex is a single point, a 1-simplex is a line segment, a 2-simplex is a triangle, a 3-simplex is a tetrahedron, and so on. In the context of a mesh, a 0-simplex is often called a *vertex*, a 1-simplex an *edge*, a 2-simplex a *face*, and a *k*-simplex where $k > 2$ a *volume* or *k-volume*.

Every simplex with dimension $k > 0$ has a *boundary* consisting of $(k - 1)$ -simplices. For instance, every triangle (2-simplex) is bounded by three line segments (1-simplices), and every line segment is bounded by two vertices. Each $(k - 1)$ -simplex on the boundary of a *k*-simplex σ_k is called a $(k - 1)$ -*face* of σ_k .

We denote simplices by listing their vertices, e.g. a triangle $\sigma_2 = \{v_0, v_1, v_2\}$.

2.2.2 Simplicial complex

A simplicial complex \mathcal{K} is a set of simplices satisfying the rules

- every face of every simplex in \mathcal{K} is also in \mathcal{K}
- any two simplices in \mathcal{K} either do not intersect at all or share an entire face.

In the case of a 2D triangle mesh, this means that every edge and vertex of every triangle is

also part of the complex, and there are no overlapping triangles or edges/vertices that aren't part of a triangle's boundary.

A similar complex could be defined without requiring all its elements to be simplicial. For instance, a rectilinear grid is a common such structure. A complex like this is called a *cell complex* and its k -dimensional elements *k-cells*. A simplex is a special case of a cell, and the concepts of boundary and face apply to all cells. We will use a simplicial complex in this thesis, however, non-simplicial cells become relevant in the *dual mesh* (Section 2.3.3).

2.2.3 Orientation of a simplex

Much like how differential forms of dimension 2 and higher have a notion of two distinct orientations, which manifest themselves in the sign of measured volumes, simplices with 2 or more vertices also have two possible orientations depending on the order of the vertices. For a line segment there are two possible ways to order its two vertices, for a triangle there are three clockwise and three counterclockwise orderings, and so on.

As part of the definition of our simplicial complex, each simplex receives an orientation. For reasons that will become clear when we define the discrete exterior derivative in Section 2.3.2, every simplex also needs its boundary to match its orientation. For a counterclockwise oriented triangle, for instance, this means the boundary line segments must form a counterclockwise circulation around the triangle. Since a line segment is often part of two triangles with conflicting orientations, the defined orientations cannot be consistent with every boundary.

To resolve inconsistent face orientations, the *boundary operator* ∂ , which takes a simplex and returns its faces, must flip every face that doesn't match the desired orientation. This is denoted as a multiplication by -1. For example, the boundary of a counterclockwise oriented triangle $\{v_0, v_1, v_2\}$ with edges $\{v_0, v_1\}$, $\{v_1, v_2\}$, $\{v_0, v_2\}$ is

$$\partial\{v_0, v_1, v_2\} = \{v_0, v_1\} + \{v_1, v_2\} - \{v_0, v_2\}.$$

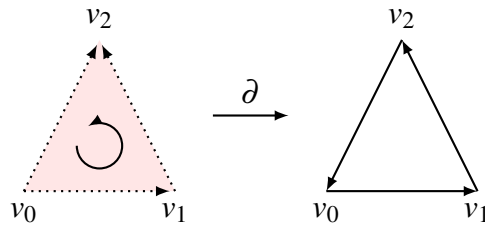


Figure 2. Boundary of a triangle with inconsistently oriented faces.

2.2.4 Chains and cochains

A k -chain on a simplicial complex \mathcal{K} is a set of real values associated with every k -simplex in \mathcal{K} . Denoting the set of all k -simplices in \mathcal{K} by \mathcal{K}^k and its cardinality by $|\mathcal{K}^k|$, a natural representation for a chain is a $|\mathcal{K}^k|$ -dimensional vector. The space of all k -chains \mathcal{C}_k is a vector space, and the set of all k -simplices can be thought of as its basis.

Whereas a chain is analogous to a vector, a *cochain* is analogous to a differential form: a linear function that takes a chain and "measures" it, producing a real number. Formally, cochains exist in the *dual space* \mathcal{C}_k^* of \mathcal{C}_k . A k -cochain $\hat{\alpha} \in \mathcal{C}_k^*$ has one linear operation for each k -simplex in \mathcal{K} , matching the dimension of the corresponding k -chain $c \in \mathcal{C}_k$, and the evaluation $\hat{\alpha}(c)$ behaves like a dot product between vectors. Thus, a natural way to represent a cochain is a $|\mathcal{K}^k|$ -dimensional *row vector*.

This analogy with differential forms will be useful when we define discrete forms in the next section.

2.3 Discrete differential forms

Equipped with an oriented simplicial complex, we can begin to map differential forms and their operations onto it. The concepts of this section are covered in more detail by Desbrun, Kanso, and Tong (2006), Crane et al. (2013), and Blair Perot and Zusi (2014).

2.3.1 Definition of discrete forms

Recalling that differential forms can be integrated over manifolds of matching dimension, a natural way to map a continuous form to a discrete value on a simplex is to integrate the form over that simplex. In other words, given a differential k -form α and k -simplex σ_k , we can get a scalar value $\hat{\alpha}$ associated with σ_k by computing the integral

$$\hat{\alpha} = \int_{\sigma_k} \alpha.$$

This is a line integral over a line segment for 1-forms, surface integral over a triangle for 2-forms, volume integral over a tetrahedron for 3-forms, and so on. Additionally, a 0-form (i.e. a scalar field) integrated over a 0-simplex (i.e. a single point) is the value of the form at that point.

Computing this integral for every k -simplex in a simplicial complex \mathcal{K} produces a set of values associated with the k -simplices, which can be interpreted as a chain or cochain. Due to the analogous nature of differential forms and cochains covered in Section 2.2.4, the most fitting choice of interpretation here is a cochain. This leads us to the following definition:

A discrete differential k -form on an oriented simplicial complex \mathcal{K} is a k -cochain obtained by integrating a continuous k -form over every k -simplex in \mathcal{K} .

The discrete equivalent of a differential form being a cochain, the discrete versions of the operations of exterior calculus are operations on cochains.

2.3.2 Discrete exterior derivative

Section 2.1.7 briefly introduced Stokes' theorem (Eq. (2.3)). In the domain of a k -simplex σ , it states that

$$\int_{\sigma} \mathbf{d}\alpha = \int_{\partial\sigma} \alpha, \quad (2.4)$$

where α is a $(k-1)$ -form.

Writing this in terms of the exterior derivative's vector analogues covered in Section 2.1.7 (∇ , $\nabla \times$, and $\nabla \cdot$) yields several well-known vector calculus identities.

When $k = 1$, Eq. (2.4) is the fundamental theorem of calculus

$$\int_{\sigma} \nabla \alpha = \alpha(v_2) - \alpha(v_1) \quad (2.5)$$

where v_1, v_2 are the endpoints of the line segment σ .

When $k = 2$, we get the curl theorem

$$\iint_{\sigma} (\nabla \times \alpha) \cdot dA = \int_{\partial\sigma} \alpha \cdot ds, \quad (2.6)$$

which is equivalent to Green's theorem in two-dimensional space.

Finally, when $k = 3$, we get the divergence theorem

$$\iiint_{\sigma} (\nabla \cdot \alpha) dV = \iint_{\partial\sigma} \alpha dA. \quad (2.7)$$

Stokes' theorem expresses and generalizes the common property between these identities, that an integral over a k -dimensional volume can be turned into an integral over the volume's $(k - 1)$ -dimensional boundary and vice versa. It is a powerful theorem with many applications. For the use case of discrete exterior calculus, it can be used as the *definition* of the discrete exterior derivative. More specifically, we can define the discrete exterior derivative as the unique operator \mathbf{d} satisfying Eq. (2.4).

To make this concrete, first note that the integrals here are exactly how cochain elements were defined in Section 2.3.1. $\int_{\sigma} \mathbf{d}\alpha$ is an element of a k -cochain, and $\int_{\partial\sigma} \alpha$ is the sum of all $(k - 1)$ -cochain elements constituting the boundary of σ . In a sense, \mathbf{d} does the opposite of the boundary operator ∂ defined in Section 2.3.1: whereas ∂ takes a simplex and produces the sum of its faces, \mathbf{d} takes the sum of a simplex's faces and produces the simplex itself. In more precise terms, \mathbf{d} is the *adjoint* of ∂ , also known as the *coboundary operator*.

Given a simplicial complex \mathcal{K} , \mathbf{d} is a linear map from \mathcal{K}^{k-1} to \mathcal{K}^k . Hence, it can be represented as a $|\mathcal{K}^k| \times |\mathcal{K}^{k-1}|$ matrix. Each row of this matrix corresponds to one k -simplex and represents a signed sum of its faces taking into account relative orientation. Each column corresponding to a boundary simplex has a coefficient of 1 or -1 depending on orientation, and the rest of the coefficients are zeroes. Graph theorists call this type of matrix an *adjacency matrix*.

Figure 3 illustrates the discrete exterior derivative on a small simplicial complex. The subscript on \mathbf{d} denotes the dimension of the input cochain, and the numbers in the diagrams represent the index of the mesh element in the cochain. The triangle faces are all oriented counterclockwise.

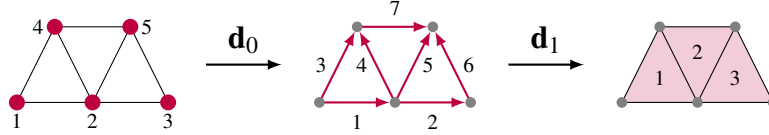


Figure 3. Discrete exterior derivative visualized on an oriented simplicial complex.

The matrix representations of \mathbf{d}_0 and \mathbf{d}_1 in Figure 3 are

$$\mathbf{d}_0 = \begin{bmatrix} -1 & 1 & & & & & & & \\ & -1 & 1 & & & & & & \\ -1 & & & 1 & & & & & \\ & -1 & & 1 & & & & & \\ & -1 & & & 1 & & & & \\ & & -1 & & & 1 & & & \\ & & & -1 & & & 1 & & \\ & & & & -1 & & & 1 & \end{bmatrix}, \quad \mathbf{d}_1 = \begin{bmatrix} 1 & & -1 & 1 & & & & & \\ & 1 & & & -1 & 1 & & & -1 \\ & & & & & & -1 & 1 & & \end{bmatrix}.$$

A remarkable property of a discrete exterior derivative defined this way is that it is *exact*. The value obtained by integrating a form over boundary simplices and taking the discrete exterior derivative is exactly the value one would obtain by first taking the continuous exterior derivative of the same form and integrating it over the bounded volume. As a result, use of the discrete exterior derivative in a numerical method introduces no additional error. This simplifies the analysis of a method's accuracy by eliminating one source of error.

2.3.3 Dual mesh

The continuous Hodge star is an isomorphism that maps a k -form to an orthogonal $(n - k)$ -form. To capture these notions of isomorphism and orthogonality in the discrete setting, a simplicial complex alone is insufficient. Some additional geometric structure is needed.

Given a simplicial complex \mathcal{K} , a secondary complex constructed for the purpose of the dis-

crete Hodge star is called a *dual mesh*, denoted \mathcal{K}^* . When the dual relationship is relevant, \mathcal{K} is in turn called the *primal mesh*.

The most important property of the dual mesh is that every k -simplex in \mathcal{K} has a corresponding $(n - k)$ -cell in \mathcal{K}^* . In two dimensions this means every primal vertex maps to a dual face, every primal edge maps to a dual edge, and every primal face maps to a dual vertex. This is illustrated in Figure 4, with the purple dotted lines representing the dual mesh.

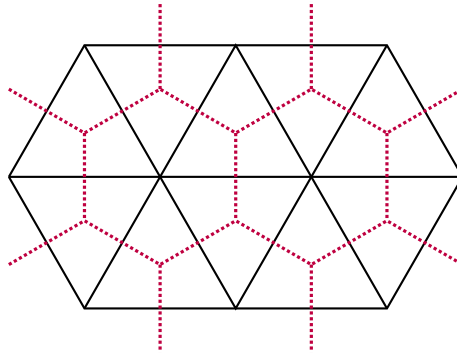


Figure 4. A mesh and its dual.

Due to this correspondence between elements of the primal and dual meshes, there is also a correspondence between their boundaries. Each primal face becomes a dual vertex that is on the boundary of every dual edge corresponding to the primal face's boundary edges, and so on. The way this manifests in the adjacency matrix representation of \mathbf{d} is that the k -dimensional exterior derivative on the dual mesh is the *transpose* of the $(n - k - 1)$ -dimensional exterior derivative on the primal mesh,

$$\mathbf{d}_k^* = \mathbf{d}_{n-k-1}^T.$$

In a sense, this is the $(n - k)$ -dimensional primal coboundary backwards. This way we only need to compute the exterior derivative matrices for the primal mesh, and their transposes can be used on the dual.

Notably, a dual mesh is generally not a simplicial one. For instance, a regular triangular grid in 2D, such as the one in Figure 4, will have hexagonal dual faces. However, the boundary operator and discrete exterior derivative are defined in a way that can handle arbitrary polytopes. The number of faces on a cell merely affects the sparsity pattern of the matrix \mathbf{d} .

A somewhat more challenging property of the dual mesh are its incomplete faces on the boundary. For instance, the dual mesh of Figure 4 has two hexagonal faces and eight edges pointing outside the primal mesh, connected to nothing at the outer end. Care must be taken to account for this when defining boundary conditions. An example of appropriately defined boundary conditions will be given in Section 3.4.

There are many possible ways to place the dual elements relative to the primal ones. In this thesis, *circumcentric* dual cells as implemented by the PyDEC software library (Bell and Hirani 2012) are used, meaning the dual vertex of a primal n -simplex is placed at the center of the smallest n -ball containing the simplex, or in other words, at the point that is equidistant from all vertices of the simplex. An important property of a dual mesh defined this way is that primal and dual elements are orthogonal to one another.

2.3.4 Discrete Hodge star

The discrete Hodge star is a mapping between k -cochains on the primal mesh and $(n - k)$ -cochains on the dual mesh. By construction of the dual mesh, this mapping in matrix form is a square ($|\mathcal{K}^k| \times |\mathcal{K}^k|$)-matrix. For efficient computation, it is desirable for this matrix to be diagonal, as diagonal matrices are fast to multiply and trivial to invert.

The Hodge star is diagonal when each element of the resulting dual cochain only depends on its corresponding primal element. The value of the diagonal elements is based on the idea that the Hodge star produces a form of equal magnitude. As we don't generally know the exact form the star will be applied to, this necessarily involves approximation.

Given a continuous k -form α , primal k -simplex σ , and its dual $(n - k)$ -cell σ^* , a simple approximation for the discrete Hodge star is obtained by assuming α is constant over σ and σ^* . Combined with the fact that α and $\star\alpha$ have equal magnitude, this approximation leads to the idea that the cochains $\hat{\alpha} = \int_{\sigma} \alpha$ and $\star\hat{\alpha} = \int_{\sigma^*} \star\alpha$ should have equal magnitude *per unit of volume*,

$$\frac{1}{|\sigma|} \hat{\alpha} = \frac{1}{|\sigma^*|} \star\hat{\alpha} \iff \star\hat{\alpha} = \frac{|\sigma^*|}{|\sigma|} \hat{\alpha}.$$

A discrete Hodge star defined using this approximation is a square diagonal matrix whose

elements are

$$\star_{ii} = \frac{|\sigma_i^*|}{|\sigma_i|}, \quad i = 1, \dots, |\mathcal{K}^k|. \quad (2.8)$$

A matrix \star_k defined this way takes k -cochains from the primal mesh to $(n - k)$ -cochains on the dual mesh. The \star_{n-k}^* taking $(n - k)$ -cochains from the dual mesh back to the primal mesh is the inverse of this, $\star_{n-k}^* = \star_k^{-1}$. Knowing this, we only need to compute the Hodge star matrices in the direction from the primal mesh to the dual, and their inverses (which are trivial to compute thanks to diagonality) can be used for the opposite direction.

This approximation, which we call *Yee's Hodge star* due to its early appearance in Yee (1966), is one of two Hodge star operators used in this thesis. A different approximation is obtained by assuming spatially harmonic forms instead of locally constant ones. The details of this will be covered in Section 3.5.

Because the discrete exterior derivative is exact, the Hodge star is the only source of approximation error in methods based on these two operators. As such, variants of the discrete Hodge star directly lead to DEC variants with different accuracy and efficiency characteristics. Examples of this are the harmonic Hodge star used in this thesis and the higher order approximations studied by Lohi (2023).

2.3.5 Interpolating discrete forms

All values in DEC computations are cochains and thus exist only on elements of a mesh. However, it is sometimes useful to get the value of a form at a single point. In this thesis, for instance, we would like to visualize the 1-form velocity as arrows pointing in the direction of velocity, but a 1-cochain has no notion of direction. It only has magnitudes over entire mesh edges. Thus, direction must be reconstructed by interpolating between multiple nearby edges. This is accomplished using *Whitney forms*, first introduced by Whitney (1957). This section gives a short summary of their definition; see e.g. Lohi and Kettunen (2021) for a thorough treatment of the topic.

Whitney forms are defined individually for every simplex in terms of *barycentric coordinates*: for a k -simplex with vertices v_0, \dots, v_k , any point within the simplex can be expressed

as

$$\sum_{i=0}^k \lambda_i v_i$$

where the coefficients λ_i are called barycentric coordinates. If $\sum_{i=0}^k \lambda_i = 1$, these coordinates are called *normalized*. Equivalently to normalized barycentric coordinates, the *barycentric function* λ_i for a vertex v_i is a piecewise linear function whose value is 1 on v_i and zero on all other vertices.

The Whitney form $\mathcal{W}(v_i)$ associated with a 0-simplex v_i is its barycentric function,

$$\mathcal{W}(v_i) = \lambda_i.$$

For a 1-simplex $\sigma = [v_0, v_1]$, it is

$$\mathcal{W}(\sigma) = \lambda_0 d\lambda_1 - \lambda_1 d\lambda_0$$

or equivalently in vector calculus notation,

$$\mathcal{W}(\sigma) = \lambda_0 \nabla \lambda_1 - \lambda_1 \nabla \lambda_0,$$

and generally for a k -simplex $\sigma = [v_0, \dots, v_k]$,

$$\mathcal{W}(\sigma) = k! \sum_{i=0}^k (-1)^i \lambda_i d\lambda_1 \wedge \dots \wedge \widehat{d\lambda_i} \wedge \dots \wedge d\lambda_k \quad (2.9)$$

where $\widehat{}$ denotes that the term is omitted.

The \mathcal{W} in the above expressions is called the *Whitney map*, which is a map from a simplicial complex \mathcal{K} to a Whitney form. Given a k -cochain $\widehat{\alpha}$ on \mathcal{K} whose element a_i is associated with the k -simplex σ_i , the Whitney map can be evaluated on $\widehat{\alpha}$ as

$$\mathcal{W}(\widehat{\alpha}) = \mathcal{W}\left(\sum_{\sigma_i \in \mathcal{K}^k} a_i \sigma_i\right) = \sum_{\sigma_i \in \mathcal{K}^k} a_i \mathcal{W}(\sigma_i). \quad (2.10)$$

The output of this map is a Whitney form, which can then be evaluated at any point x in space. To do this, one would first find the n -simplex x lies in (since all the barycentric functions and hence the Whitney forms for all simplices outside of it are zero), where n is the dimension of the working space, find all the k -simplices that are faces of this n -simplex, and sum their

respective Whitney forms at x . The use case in this thesis is interpolation of 1-forms in \mathbb{R}^2 , in which case one would find the triangle x lies in with sides $\sigma_0, \sigma_1, \sigma_2$ and associated cochain elements a_0, a_1, a_2 , and take the sum

$$a_0(\mathcal{W}(\sigma_0))(x) + a_1(\mathcal{W}(\sigma_1))(x) + a_2(\mathcal{W}(\sigma_2))(x).$$

3 Model

The physical phenomenon we apply the DEC to is the propagation and scattering of acoustic waves in a two-dimensional domain. This chapter will cover the mathematical models used; the details of numerical experiments performed and their results are described in Chapter 5.

3.1 Vector equations

We consider waves described by the differential equation

$$\frac{\partial^2 \phi}{\partial t^2} - c^2 \nabla^2 \phi = 0 \quad (3.1)$$

where ϕ is a velocity potential, t is time and c is the speed of wave propagation. This equation is equivalent to the first-order system

$$\begin{cases} \frac{\partial p}{\partial t} - c^2 \nabla \cdot \mathbf{v} = 0 \\ \frac{\partial \mathbf{v}}{\partial t} - \nabla p = 0 \end{cases} \quad (3.2)$$

where $p = \frac{\partial \phi}{\partial t}$ is the scalar-valued acoustic pressure and $\mathbf{v} = \nabla \phi$ is the vector-valued velocity of particles perturbed by the wave.

The system (3.2) applies within the spatial domain Ω and the time domain $[0, T]$.

3.2 Exterior calculus equations

The first step to translating the vector system (3.2) to exterior calculus notation is the selection of appropriate differential forms to represent the variables p and \mathbf{v} . Because a scalar can be represented either by a 0-form or a volume form, and a vector by a 1-form or $(n-1)$ -form, there are two possible choices for each variable, both of which can be made equivalent to the vector equations by the application of Hodge stars when necessary. The physical properties of a variable can hint at the most appropriate choice, but one can also begin with 0- and

1-forms and pick the most convenient representation for the end result.

We begin by representing pressure with the 0-form p and velocity with the 1-form $v = \mathbf{v}^\flat$. The exterior derivative of a 1-form corresponds to the curl $\nabla \times$, but we need the divergence of v in the first equation of (3.2). In three dimensions, divergence is the exterior derivative of a 2-form, meaning computing the divergence of a 1-form involves first converting it to a 2-form with a Hodge star, $\nabla \cdot \mathbf{v} \approx \mathbf{d} \star v$. The equivalent turns out to be true in 2D as well.

To illustrate why, Figure 5 invokes Stokes' theorem. Divergence integrated over an area is equal to the boundary curve integral in the outward-pointing normal direction. Curl, on the other hand, is the boundary curve integral in the tangent direction. Thus, we can compute divergence by first rotating such that the boundary normal points in the tangent direction, which is exactly the effect of the 1-form Hodge star, and then taking the curl, i.e. the exterior derivative.

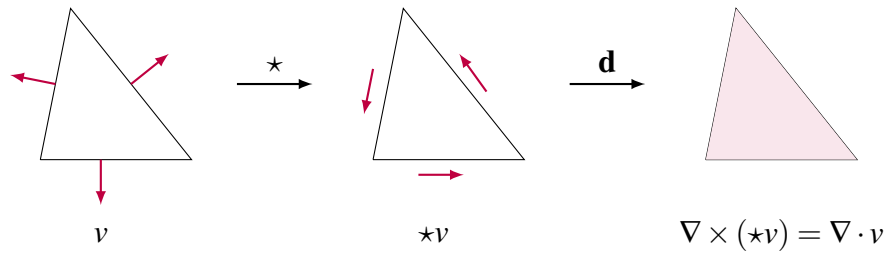


Figure 5. 2D divergence as the exterior derivative of a rotated form.

Note that divergence expressed this way is a 2-form, and we've chosen to represent p as a 0-form. To match the dimension of p , an additional Hodge star needs to be applied to the divergence. The first equation in 3.2 thus becomes

$$\frac{\partial p}{\partial t} - c^2 \star \mathbf{d} \star v = 0. \quad (3.3)$$

The second equation is more straightforward, as the exterior derivative of a 0-form directly corresponds to the gradient. In exterior calculus notation, it becomes

$$\frac{\partial v}{\partial t} - \mathbf{d}p = 0. \quad (3.4)$$

One more modification is needed to make these equations useful to us. The implementation of boundary conditions, which will be covered in Section 3.4, requires information about the

flux of velocity across edges. The flux is the component of velocity normal to the boundary, $\mathbf{v} \cdot \mathbf{n}$. If we represent v as a 1-cochain, integrating v over boundary edges in the tangent direction, information about the normal direction is lost. Thus, to have access to flux in the discretized equations, the equations must be expressed in terms of flux instead of velocity.

In exterior calculus notation the flux is $\star v$. The reason for this is the same as the reason why divergence can be computed as the curl of $\star v$. In a sense, the star turns a line integrated in the tangent direction into a line integrated in the normal direction, analogously to the 3D case where a 1-form (line) turns into a 2-form (surface) integrated in the normal direction.

The final choices for the unknowns in our equations are the 0-form pressure p and the 1-form flux $q = \star v$. Substituting these variables into Eq. (3.3) and (3.4) and applying one more Hodge star to Eq. (3.4) yields the system

$$\begin{cases} \frac{\partial p}{\partial t} - c^2 \star \mathbf{d}q = 0 \\ \frac{\partial q}{\partial t} - \star \mathbf{d}p = 0 \end{cases} \quad (3.5)$$

3.3 Discretization

Discretization of the model into a computer-solvable form is performed separately for space and time. The procedures defined by DEC are used for the spatial part, and time discretization is performed using a staggered finite difference scheme known as the leapfrog method.

3.3.1 Space

Spatial discretization in DEC is straightforward: given a continuous model in exterior calculus notation, a simplicial complex \mathcal{K} approximating the domain Ω and its dual, replace k -forms in the model with k -cochains, \mathbf{d} with the coboundary operator, and \star with the discrete Hodge star, and you're most of the way done. Denoting the 0-cochain equivalent of p by P and the 1-cochain equivalent of q by Q , this yields the semi-discrete equations

$$\begin{cases} \frac{\partial P}{\partial t} - c^2 \star \mathbf{d}Q = 0 \\ \frac{\partial Q}{\partial t} - \star \mathbf{d}P = 0 \end{cases} \quad (3.6)$$

(only *semi*-discrete at this point because they are still continuous in time).

The only remaining question is whether the cochains should be placed on the primal mesh or the dual. In our case, the implementation of boundary conditions demands Q be available on the boundary edges, which means Q must be on primal edges (as dual edges are orthogonal to the boundary instead). From there, P should be placed such that the terms in Eq. (3.6) are in the same cochain space. In the first equation, $\star \mathbf{d}Q$ takes Q from primal edges to primal faces and from there to dual vertices, suggesting that P should be a **dual** 0-cochain. Checking that $\star \mathbf{d}P$ in the second equation is a primal 1-cochain confirms this as the correct choice.

With these choices, we can pin down which forms of \star and \mathbf{d} are used in the system (3.6). Recall that given the matrices \star_k and \mathbf{d}_k defined in terms of the primal mesh, the corresponding operators on the dual mesh are \star_k^{-1} and \mathbf{d}_{n-k-1} . In terms of these operators, the system (3.6) becomes

$$\begin{cases} \frac{\partial P}{\partial t} - c^2 \star_1 \mathbf{d}_1 Q = 0 \\ \frac{\partial Q}{\partial t} - \star_1^{-1} \mathbf{d}_1^T P = 0. \end{cases} \quad (3.7)$$

To produce the cochain elements P_i and Q_i on the dual vertices v_i^* and primal edges \mathcal{E}_i from the continuous variables p and \mathbf{v} , we get the formulas

$$P_i = p(v_i^*) \quad (3.8)$$

$$Q_i = \int_{\mathcal{E}_i} (\mathbf{v} \cdot \mathbf{n}) ds. \quad (3.9)$$

Note the term $\mathbf{v} \cdot \mathbf{n}$, where \mathbf{n} is the unit-length normal of the edge, computing the flux of velocity in Eq. (3.9).

3.3.2 Time

For time discretization we use the leapfrog method, which is a second-order finite difference scheme commonly used in "Yee-like schemes" (as termed by Bossavit and Kettunen (1999)) such as DEC, including Yee (1966) itself. It is simple and efficient method, requiring only one value of each variable to be computed per timestep. See the lecture notes of Young (2014) for an explanation of some of its desirable properties.

In the leapfrog method, time advances in steps of Δt seconds, and values of P and Q are computed at separate time instances offset from each other by $\frac{\Delta t}{2}$. Let $t^k = k\Delta t$ denote the

value of time after k timesteps and P^k the value of P at this time. The next value of Q is computed at the time $t^k + \frac{\Delta t}{2}$, which we denote $Q^{k+\frac{1}{2}}$. The time derivatives of P and Q are then approximated with

$$\frac{\partial P}{\partial t}(t^{k+\frac{1}{2}}) = \frac{P^{k+1} - P^k}{\Delta t} \quad (3.10)$$

$$\frac{\partial Q}{\partial t}(t^k) = \frac{Q^{k+\frac{1}{2}} - Q^{k-\frac{1}{2}}}{\Delta t} \quad (3.11)$$

Substituting Eq. (3.10) and (3.11) into Eq. (3.7) yields

$$\begin{aligned} \frac{P^{n+1} - P^n}{\Delta t} - c^2 \star_1 \mathbf{d}_1 Q^{n+\frac{1}{2}} &= 0 \\ \frac{Q^{n+\frac{3}{2}} - Q^{n+\frac{1}{2}}}{\Delta t} - \star_1^{-1} \mathbf{d}_1^T P^{n+1} &= 0 \end{aligned}$$

Multiplying by Δt and moving terms to the right-hand side produces the timestep formulas

$$P^{n+1} = P^n + \Delta t c^2 \star_1 \mathbf{d}_1 Q^{n+\frac{1}{2}} \quad (3.12)$$

$$Q^{n+\frac{3}{2}} = Q^{n+\frac{1}{2}} + \Delta t \star_1^{-1} \mathbf{d}_1^T P^{n+1} \quad (3.13)$$

This is an *explicit* method: the timestep formulas can be applied as matrix multiplications to the values of P and Q at the beginning of a timestep, without being required to solve any linear systems. This makes the method computationally efficient and highly parallelizable, but only conditionally stable. There is an upper limit to the timestep Δt , known as the *Courant-Friedrichs-Lewy condition* or CFL condition (Courant, Friedrichs, and Lewy 1967), above which the system will begin to gain energy and "blow up" towards infinity. Determining the exact value of this condition analytically is a challenging problem. For our purposes it is sufficient to find an approximate value via numerical experiments.

3.4 Boundary conditions

Our main test case models the scattering of acoustic waves on impact with a sound-soft obstacle. For this purpose we define two boundaries, Γ_{sca} for the scattering obstacle and Γ_{ext} for the outer boundary of the computational domain, illustrated in Figure 6.

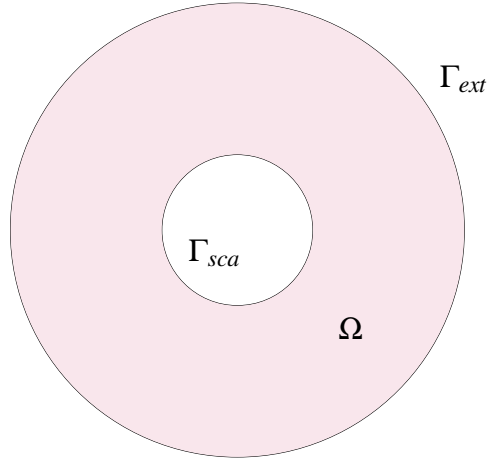


Figure 6. The domain Ω and boundaries Γ_{ext} and Γ_{sca} of a scattering model.

Given an incident wave with velocity potential ϕ_{inc} , the scattering obstacle is modelled by setting ϕ_{inc} on Γ_{sca} as a Dirichlet boundary condition

$$\mathbf{v} = \nabla \phi_{inc} \quad \text{in } \Gamma_{sca} \times [0, T] \quad (3.14)$$

where $[0, T]$ is the simulated time period. The boundary condition is only applied to \mathbf{v} and not p because p is placed on the dual mesh, which does not have elements at the boundary.

The domain should be as small as possible to minimize the amount of computation required, but we wish to simulate the scattering object sitting in empty space. Placing the outer boundary Γ_{ext} close to the scatterer requires the modelling of waves passing through the boundary uninterrupted, without any reflections occurring. This is accomplished using a first-order absorbing Engqvist-Majda boundary condition (Engqvist and Majda 1977)

$$\frac{1}{c} \frac{\partial \phi}{\partial t} + \mathbf{n} \cdot \nabla \phi = 0 \quad \text{in } \Gamma_{ext} \times [0, T].$$

Substituting the variables $p = \frac{\partial \phi}{\partial t}$ and $\mathbf{v} = \nabla \phi$ yields

$$\frac{1}{c} p + \mathbf{n} \cdot \mathbf{v} = 0,$$

which translates to the differential form expression

$$\frac{1}{c} p + \star v = 0.$$

Finally, we can substitute in our model variable $q = \star v$, producing the boundary conditions

$$q = \star(\nabla\phi_{inc})^b \quad \text{in } \Gamma_{ext} \times [0, T] \quad (3.15)$$

$$\frac{1}{c}p + q = 0 \quad \text{in } \Gamma_{sca} \times [0, T] \quad (3.16)$$

An issue with the absorbing boundary condition (Eq. (3.16)) is that p and q exist on different mesh elements in our DEC model. An approximation is needed to turn one of p or q into a form compatible with the other. We use the same approximation used to define the discrete (Yee's) Hodge star, the assumption of locally constant forms.

Specifically, we assume p is constant over the primal face corresponding to each dual vertex P is stored on, and therefore constant over the nearest boundary edge when the primal face is adjacent to the boundary. Denoting a boundary edge by \mathcal{E}_i , the corresponding flux cochain element by Q_i and the pressure of the adjacent face's dual vertex by $P_{\mathcal{E}_i}$, Eq. (3.16) is approximated with

$$Q_i + \frac{1}{c} \int_{\mathcal{E}_i} P_{\mathcal{E}_i} ds = 0,$$

leading to the flux update formula for boundary edges

$$Q_i = -\frac{1}{c} |\mathcal{E}_i| P_{\mathcal{E}_i}, \quad (3.17)$$

where $|\mathcal{E}_i|$ is the length of the edge.

3.5 Harmonic operators

The time discretization of Section 3.3.2 and the discrete Hodge star of Section 2.3.4 are based on linear and locally constant approximations. However, in our case we know we're simulating harmonic waves, which is a fact that can be used to derive more accurate approximations of the time derivative and Hodge star.

A similar approximation for the time derivative has been used in the context of electromagnetics by Ma and Chen (2005), Rabinä (2014), and Peng and Appelö (2022). The harmonic Hodge star was derived for the three-dimensional edge-to-face case by Rabinä (2014). In this section we use the same ideas to derive two-dimensional edge-to-edge and face-to-vertex Hodge star operators.

3.5.1 Exact timestepping

The state of a time-harmonic wave at timestep k and time t can be expressed in terms of complex exponentiation as

$$P(t) = \text{Re}(\hat{P}_0 \exp(-i\omega t)) \quad (3.18)$$

$$Q(t + \frac{\Delta t}{2}) = \text{Re}(\hat{Q}_0 \exp(-i\omega(t + \frac{\Delta t}{2}))) \quad (3.19)$$

where \hat{P}_0 and \hat{Q}_0 are complex initial values for P and Q , ω is the angular velocity of the wave, and i is the imaginary unit.

Substituting Eq. (3.18) into the numerator of Eq. (3.10) where $t^k = t + \frac{\Delta t}{2}$, we get

$$\begin{aligned} & P(t^k + \frac{\Delta t}{2}) - P(t^k - \frac{\Delta t}{2}) \\ &= \text{Re} \left[\hat{P}_0 \exp(-i\omega(t^k + \frac{\Delta t}{2})) \right] - \text{Re} \left[\hat{P}_0 \exp(-i\omega(t^k - \frac{\Delta t}{2})) \right] \\ &= \text{Re} \left[\hat{P}_0 (\exp(-i\omega(t^k + \frac{\Delta t}{2})) - \exp(-i\omega(t^k - \frac{\Delta t}{2}))) \right] \\ &= \text{Re} \left[\hat{P}_0 \exp(-i\omega t^k) \left(\exp(-i\omega \frac{\Delta t}{2}) - \exp(i\omega \frac{\Delta t}{2}) \right) \right] \\ &= \text{Re} \left[-i\omega \hat{P}_0 \exp(-i\omega t^k) \left(\frac{\exp(-i\omega \frac{\Delta t}{2}) - \exp(i\omega \frac{\Delta t}{2})}{-i\omega} \right) \right] \\ &= \text{Re} \left[\frac{\partial P}{\partial t}(t^k) \left(\frac{\exp(-i\omega \frac{\Delta t}{2}) - \exp(i\omega \frac{\Delta t}{2})}{-i\omega} \right) \right] \\ &= -\frac{1}{\omega} \frac{\partial P}{\partial t}(t^k) \cdot \text{Im} \left[\exp(-i\omega \frac{\Delta t}{2}) - \exp(i\omega \frac{\Delta t}{2}) \right] \\ &= \frac{\partial P}{\partial t}(t^k) \left(\frac{2}{\omega} \sin\left(\frac{\omega \Delta t}{2}\right) \right). \end{aligned}$$

The same consideration for the flux (Eq. (3.19)) produces the same coefficient.

With this, the time derivative approximations (Eq. (3.10) and (3.11)) become

$$\frac{\partial P}{\partial t} = \frac{P^{n+1} - P^n}{\frac{2}{\omega} \sin \frac{\omega \Delta t}{2}} \quad (3.20)$$

$$\frac{\partial Q}{\partial t} = \frac{Q^{n+\frac{3}{2}} - Q^{n+\frac{1}{2}}}{\frac{2}{\omega} \sin \frac{\omega \Delta t}{2}} \quad (3.21)$$

and the timestep equations (3.12) and (3.13)

$$P^{n+1} = P^n + \frac{2}{\omega} \sin \frac{\omega \Delta t}{2} (c^2 \star_2 d_1 Q^{n+\frac{1}{2}}) \quad (3.22)$$

$$Q^{n+\frac{3}{2}} = Q^{n+\frac{1}{2}} + \frac{2}{\omega} \sin \frac{\omega \Delta t}{2} (\star_1^{-1} d_1^T P^{n+1}). \quad (3.23)$$

3.5.2 Harmonic Hodge star

The harmonic properties of acoustic waves also apply to space. In a small neighborhood around any given point, it can be assumed that the solution behaves like a complex plane wave

$$\hat{u}(\mathbf{x}) = \hat{u}_0 \exp(i\vec{\kappa} \cdot \mathbf{x}) \quad (3.24)$$

where \hat{u}_0 is a complex initial value, and $\vec{\kappa}$ is the *wave vector* $\kappa \mathbf{d}$, where in turn \mathbf{d} is the unit-length direction of wave propagation and κ is the *wavenumber*. The wavenumber is the number of wave periods per unit of distance, related to the wavelength λ by $\kappa = \frac{2\pi}{\lambda}$ and the angular velocity ω and propagation speed c by $\kappa = \frac{\omega}{c}$.

Edge to edge

In the case of \star_1 taking a 1-cochain from a flux on the primal edge \mathcal{E} to a velocity on the dual edge \mathcal{E}^* , we would like to minimize the error

$$\left\| \int_{\mathcal{E}} \hat{u} \cdot \mathbf{n} ds - \star_1^{-1} \int_{\mathcal{E}^*} \hat{u} \cdot \mathbf{t} ds \right\|^2 \quad (3.25)$$

where \mathbf{n} is the unit normal vector to \mathcal{E} and \mathbf{t} is the unit tangent vector to \mathcal{E}^* . Denoting $\hat{u}_{\mathcal{E}} = \int_{\mathcal{E}} \hat{u} \cdot \mathbf{n} ds$ and $\hat{u}_{\mathcal{E}^*} = \int_{\mathcal{E}^*} \hat{u} \cdot \mathbf{t} ds$, this gives the quadratic equation

$$\hat{u}_{\mathcal{E}}^2 - 2 \star_1^{-1} \hat{u}_{\mathcal{E}} \hat{u}_{\mathcal{E}^*} + (\star_1^{-1})^2 \hat{u}_{\mathcal{E}^*}^2,$$

the derivative of which with respect to \star_1 is

$$2 \star_1^{-1} \hat{u}_{\mathcal{E}^*}^2 - 2 \hat{u}_{\mathcal{E}} \hat{u}_{\mathcal{E}^*},$$

which is zero and thus the error is minimized when

$$\star_1^{-1} = \frac{\hat{u}_{\mathcal{E}} \hat{u}_{\mathcal{E}^*}}{\hat{u}_{\mathcal{E}^*}^2} \iff \star_1 = \frac{\hat{u}_{\mathcal{E}^*}^2}{\hat{u}_{\mathcal{E}} \hat{u}_{\mathcal{E}^*}}.$$

However, this only applies when the propagating direction \mathbf{d} is a known constant, which is generally not true. For practical cases, we need to minimize this error over all possible propagating directions. This is accomplished by integrating \mathbf{d} over the unit circle in polar coordinates, where $\mathbf{d} = (\cos \theta, \sin \theta)$, resulting in the formula

$$\star_1 = \frac{\int_0^{2\pi} \hat{u}_{\mathcal{E}^*}^2 d\theta}{\int_0^{2\pi} \hat{u}_{\mathcal{E}} \hat{u}_{\mathcal{E}^*} d\theta}. \quad (3.26)$$

For the following computation, we assume \mathcal{E} and \mathcal{E}^* are orthogonal and centered at the origin. Due to the symmetry of the unit circle, orientation of the edges has no effect and thus we can also assume \mathcal{E} is oriented along the x-axis and \mathcal{E}^* along the y-axis. Using the Taylor series $\exp(ax) = \sum_{n=0}^{\infty} \frac{a^n x^n}{n!}$ and the auxiliary variable $\alpha = i\kappa \cos \theta$, we get

$$\begin{aligned} \hat{u}_{\mathcal{E}} &= \int_{\mathcal{E}} \hat{u} \cdot \mathbf{n} ds \\ &= (\hat{u}_0)_y \int_{-\frac{l}{2}}^{\frac{l}{2}} \exp(i\kappa x \cos \theta) dx \\ &= (\hat{u}_0)_y \int_{-\frac{l}{2}}^{\frac{l}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n x^n}{n!} dx \\ &= (\hat{u}_0)_y \left[\sum_{n=0}^{\infty} \frac{1}{n} \frac{\alpha^n x^{n+1}}{n!} \right]_{x=-\frac{l}{2}}^{x=\frac{l}{2}} \\ &= (\hat{u}_0)_y \sum_{n=0}^{\infty} \frac{\alpha^n \left(\left(\frac{l}{2}\right)^{n+1} - \left(-\frac{l}{2}\right)^{n+1} \right)}{(n+1)!} \\ &\text{(terms where } n \text{ is odd are eliminated)} \\ &= (\hat{u}_0)_y \sum_{n=0}^{\infty} \frac{\alpha^{2n} 2 \left(\frac{l}{2}\right)^{2n+1}}{(2n+1)!} \\ &= (\hat{u}_0)_y \sum_{n=0}^{\infty} \frac{\alpha^{2n} l^{2n+1}}{2^{2n} (2n+1)!} \\ &= (\hat{u}_0)_y l \sum_{n=0}^{\infty} \frac{(\alpha l)^{2n}}{2^{2n} (2n+1)!} \end{aligned}$$

where l is the length of \mathcal{E} and $(\hat{u}_0)_y$ is the y component of \hat{u}_0 .

Similarly, the exponential term for $\hat{u}_{\mathcal{E}^*}$ and the same integration produces

$$\hat{u}_{\mathcal{E}^*} = (\hat{u}_0)_y l^* \sum_{n=0}^{\infty} \frac{(\beta l^*)^{2n}}{2^{2n} (2n+1)!}.$$

where l^* is the length of \mathcal{E}^* .

The first few terms of this series are

$$\hat{u}_{\mathcal{E}} = (\hat{u}_0)_y l \left(1 + \frac{(\alpha l)^2}{24} + \frac{(\alpha l)^4}{1920} + \dots \right)$$

and the y component of \hat{u}_0 is $(u_0)_y = u_0 \sin \theta$, where $u_0 = |\hat{u}_0|$ is the magnitude of \hat{u}_0 .

With these series expressions we can build the products in the optimized Hodge star as series whose first few terms are

$$\begin{aligned} \hat{u}_{\mathcal{E}^*}^2 &= u_0^2 (\sin^2 \theta) (l^*)^2 \left(1 + \frac{(\beta l^*)^2}{12} + \frac{(\beta l^*)^4}{360} + \dots \right) \\ \hat{u}_{\mathcal{E}} \hat{u}_{\mathcal{E}^*} &= u_0^2 (\sin^2 \theta) l l^* \left(1 + \frac{(\alpha l)^2}{24} + \frac{(\beta l^*)^2}{24} + \frac{(\alpha l \beta l^*)^2}{576} + \dots \right). \end{aligned}$$

Now we need to integrate these expressions around the unit circle. First, we can get rid of the imaginary terms in α and β by evaluating the even powers. Switching notation to denote the edge lengths by $|\mathcal{E}|$ and $|\mathcal{E}^*|$, we get

$$\begin{aligned} (\alpha l)^2 &= -\kappa^2 |\mathcal{E}|^2 \cos^2 \theta \\ (\beta l^*)^2 &= -\kappa^2 |\mathcal{E}^*|^2 \sin^2 \theta. \end{aligned}$$

Substituting these values we get the integrals

$$\begin{aligned} \int_0^{2\pi} \hat{u}_{\mathcal{E}^*}^2 d\theta &= u_0^2 |\mathcal{E}^*|^2 \int_0^{2\pi} \left(\sin^2 \theta - \frac{\kappa^2 |\mathcal{E}^*|^2 \sin^4 \theta}{12} + \frac{\kappa^4 |\mathcal{E}^*|^4 \sin^6 \theta}{360} + \dots \right) d\theta \\ &= u_0^2 |\mathcal{E}^*|^2 \pi \left(1 - \frac{\kappa^2 |\mathcal{E}^*|^2}{16} \right) + \mathcal{O}(\kappa^4 |\mathcal{E}^*|^4) \\ \int_0^{2\pi} \hat{u}_{\mathcal{E}} \hat{u}_{\mathcal{E}^*} d\theta &= u_0^2 |\mathcal{E}| |\mathcal{E}^*| \int_0^{2\pi} \left(\sin^2 \theta - \frac{\kappa^2 |\mathcal{E}|^2 \sin^2 \cos^2 \theta}{24} - \frac{\kappa^2 |\mathcal{E}^*|^2 \sin^4 \theta}{24} \right. \\ &\quad \left. + \frac{\kappa^4 |\mathcal{E}|^2 |\mathcal{E}^*|^2 \sin^4 \cos^2 \theta}{576} + \dots \right) d\theta \\ &= u_0^2 |\mathcal{E}| |\mathcal{E}^*| \pi \left(1 - \frac{\kappa^2 |\mathcal{E}|^2}{96} - \frac{\kappa^2 |\mathcal{E}^*|^2}{32} \right) + \mathcal{O}(\kappa^4 |\mathcal{E}|^2 |\mathcal{E}^*|^2) \end{aligned}$$

Dividing these according to Eq. (3.26) yields

$$\star_1 \approx \frac{|\mathcal{E}^*|}{|\mathcal{E}|} \left(\frac{1 - \frac{\kappa^2 |\mathcal{E}^*|^2}{16}}{1 - \frac{\kappa^2 |\mathcal{E}|^2}{96} - \frac{\kappa^2 |\mathcal{E}^*|^2}{32}} \right). \quad (3.27)$$

Face to vertex

The same consideration for \star_2 taking a 2-cochain on a primal face \mathcal{F} to a 0-cochain on a dual vertex v^* yields

$$\star_2 = \frac{\int_0^{2\pi} \hat{u}(v^*)^2 d\theta}{\int_0^{2\pi} \hat{u}_{\mathcal{F}} \hat{u}(v^*) d\theta} \quad (3.28)$$

where $\hat{u}_{\mathcal{F}}$ is the wave equation (3.24) integrated over the face and $\hat{u}(v^*)$ is the wave evaluated at the vertex. We can assume for now that the vertex is at the origin and the face is a circle with radius r and area πr^2 . A generalization for polygon faces will be performed later.

The vertex being at the origin implies $\hat{u}(v^*) = \hat{u}(0) = \exp(0) = 1$.

The integral of \hat{u} over the circular face \mathcal{F} is symmetric for all angles θ , so we may assume $\theta = \frac{\pi}{2}$, i.e. that the wave is propagating along the y-axis. We then have

$$\begin{aligned} \hat{u}_{\mathcal{F}} &= \int_{\mathcal{F}} \hat{u} \\ &= \int_{-r}^r \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} \exp(i\kappa y) dy dx \\ &= \int_{-r}^r \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} \sum_{n=0}^{\infty} \frac{(i\kappa)^n y^n}{n!} dy dx \\ &= \int_{-r}^r \sum_{n=0}^{\infty} \frac{(i\kappa)^n \left((\sqrt{r^2-x^2})^{n+1} - (-\sqrt{r^2-x^2})^{n+1} \right)}{(n+1)!} dx \\ &\quad \text{(terms where } n \text{ is odd are eliminated)} \\ &= \int_{-r}^r \sum_{n=0}^{\infty} \frac{2(i\kappa)^{2n} (\sqrt{r^2-x^2})^{2n+1}}{(2n+1)!} dx. \end{aligned}$$

Here we change the variable to t such that $x = r \sin t$, for which $dx = r \cos t dt$ and $\sqrt{r^2-x^2} = r \cos t$, and continue to integrate the first few terms of the series:

$$\begin{aligned} \hat{u}_{\mathcal{F}} &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \sum_{n=0}^{\infty} \frac{2(i\kappa)^{2n} (r \cos t)^{2n+1}}{(2n+1)!} r \cos t dt \\ &= \sum_{n=0}^{\infty} \frac{2(i\kappa)^{2n} r^{2n+2}}{(2n+1)!} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} (\cos t)^{2n+2} dt \\ &= \pi r^2 \left(1 + \frac{(i\kappa r)^2}{8} + \frac{(i\kappa r)^4}{192} + \dots \right) \end{aligned}$$

Integrating $\hat{u}_{\mathcal{F}}\hat{u}(v^*)$ over the unit circle, evaluating the i^2 terms, and substituting $|\mathcal{F}| = \pi r^2$ gives

$$\begin{aligned} \int_0^{2\pi} \hat{u}_{\mathcal{F}}\hat{u}(v^*) d\theta &\approx |\mathcal{F}| \int_0^{2\pi} \left(1 - \frac{(\kappa r)^2}{8} + \frac{(\kappa r)^4}{192} + \dots\right) d\theta \\ &= 2\pi |\mathcal{F}| \left(1 - \frac{(\kappa r)^2}{8}\right) + \mathcal{O}\left(\frac{(\kappa r)^4}{192}\right). \end{aligned}$$

Because $\hat{u}(v^*)^2$ is the constant 1, its integral over the unit circle is 2π . It then follows from Eq. (3.28) that

$$\star_2 \approx \frac{1}{|\mathcal{F}|} \left(\frac{1}{1 - \frac{(\kappa r)^2}{8}} \right). \quad (3.29)$$

Polygon faces

The preceding derivation was based on circular faces, but the faces in practical applications are polygons. We can make use of results obtained from the circle case by approximating polygons with circles.

Given a regular polygon with maximum internal sphere radius r and minimum external sphere radius R , performing the integrations of Eq. (3.28) over a triangle or square using the approximation $r_{\mathcal{F}}^2 = \frac{1}{3}(2r^2 + R^2)$ matches the first three terms of the Taylor polynomial obtained from the circle case.

With irregular convex polygons, each edge defines a different value for r and each vertex a different value for R , as illustrated in Figure 7. r_k is the perpendicular distance from the origin to the edge k and R_k is the distance to the vertex k .

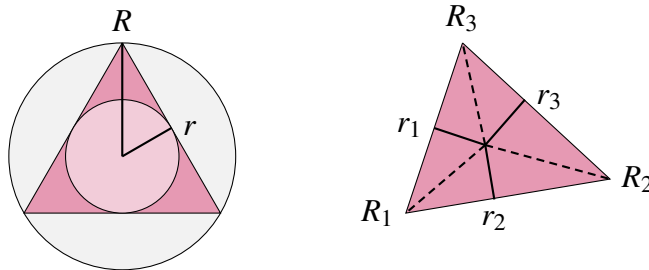


Figure 7. Internal and external radii of a regular and irregular triangle.

The approximation we use for irregular polygons with n edges is the average of the regular

polygon approximations corresponding to each edge

$$r_{\mathcal{F}}^2 \approx \frac{1}{3n} \sum_{k=1}^n (2r_k^2 + R_k^2). \quad (3.30)$$

Substituting this expression for r^2 in Eq. (3.29) produces the \star_2 operator we use in our experiments.

4 Exact controllability

We use a solver based on discrete exterior calculus to simulate a wave system forward in time. In addition to this, we're interested in finding a state that is *time-periodic* in the entire domain Ω , meaning that the state of the simulation repeats at intervals of T seconds. Under the right circumstances, this can be achieved with our time-dependent solver by simply running it for a long enough time period. This is called the *asymptotic iteration method* because the time-periodic state is approached over time as $t \rightarrow \infty$.

A solution becomes time-periodic when all waves produced by a source have reached an absorbing material or boundary. This can only happen if all wave sources are time-periodic with period T and all waves propagating from the sources are able to be absorbed by something. Even when these conditions are fulfilled, convergence of the asymptotic iteration can be slow or unreliable in cases where waves spend a long time reflecting off surfaces before being absorbed, as we find in the experiments of Chapter 5.

To accelerate convergence towards a time-periodic state, we employ the concept of exact controllability developed by Lions (1988) and first applied to the computation of time-periodic solutions by Bristeau, Glowinski, and P eriaux (1998). Our controllability method formulates the difference between the initial state at $t = 0$ and the state at $t = T$ as a least-squares optimization problem with the initial state as the variable to be optimized, which is solved using the *conjugate gradient* (CG) algorithm. The gradients needed for this algorithm are obtained using the *adjoint state* method.

4.1 Optimization problem

Let $U^n = (P^n, Q^{n+1})^T$ be the state of the simulation at timestep n , $\mathbf{e} = U^0$ the initial state, and N the number of timesteps required to simulate the time period T . We want to find a state where $U^N - \mathbf{e} = 0$, which is equivalent to minimizing the quadratic function

$$J(\mathbf{e}) = \frac{1}{2}(U^N - \mathbf{e})^T(U^N - \mathbf{e}). \quad (4.1)$$

We call $J(\mathbf{e})$ the *control energy*. To solve this problem, we first need the gradient of J .

4.2 Computing the gradient

To compute the gradient $\nabla J(\mathbf{e})$, we first need an expression for U^N . This is accomplished by expressing the timestepping procedure as a matrix multiplication and repeatedly applying it until the end of the time window T . This expression is used to expand the product in Eq. (4.1) into a standard quadratic equation and find its gradient by inspection.

4.2.1 State equation

Inside of the domain Ω , a single timestep using the formulas of Eq. (3.12) and (3.13) can be expressed as the block matrix equation

$$\begin{bmatrix} I & \Delta t c^2 \star_2 d_1 & & -I \\ & I & \Delta t \star_1^{-1} d_1^T & -I \end{bmatrix} \begin{bmatrix} P^n \\ Q^{n+\frac{1}{2}} \\ P^{n+1} \\ Q^{n+\frac{3}{2}} \end{bmatrix} = 0.$$

This can be extended to the entire time window $[0, T]$ by applying it repeatedly for N timesteps, producing the equation

$$S(\mathbf{e}) = \begin{bmatrix} I & & & & & & \\ \mathcal{C} & \mathcal{D} & & & & & \\ & \mathcal{C} & \mathcal{D} & & & & \\ & & \ddots & \ddots & & & \\ & & & \mathcal{C} & \mathcal{D} & & \\ & & & & \mathcal{C} & \mathcal{D} & \end{bmatrix} \begin{bmatrix} U^0 \\ U^1 \\ \vdots \\ U^{N-1} \\ U^N \end{bmatrix} - \begin{bmatrix} \mathbf{e} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathcal{F}^0 \\ \vdots \\ \mathcal{F}^{N-2} \\ \mathcal{F}^{N-1} \end{bmatrix} = 0 \quad (4.2)$$

where

$$\mathcal{C} = \begin{bmatrix} I & \Delta t c^2 \star_2 d_1 \\ 0 & I \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} -I & 0 \\ \Delta t \star_1^{-1} d_1^T & -I \end{bmatrix},$$

and the vectors \mathcal{F}^n contain source terms, which in our case only include the values of the Dirichlet boundary condition (Eq. (3.15)) on the boundary Γ_{sca} . Equation (4.2) is called the *state equation*.

The domain boundaries need some special attention here, as the equation as stated above does not apply to them. The details are left out of Eq. (4.2) for clarity, but it should be un-

derstood that rows of this equation corresponding to mesh edges on the scattering obstacle's boundary Γ_{sca} are replaced with rows that only have (-1) on the diagonal. Similarly, rows corresponding to edges on the absorbing boundary Γ_{ext} are replaced with the matrix form of Eq. (3.16).

Denoting $Q = -\mathcal{D}^{-1}\mathcal{C}$, a single timestep in this matrix notation consists of the formula

$$U^{k+1} = -\mathcal{D}^{-1}(\mathcal{C}U^k + \mathcal{F}^k). \quad (4.3)$$

Applying this N times to the initial conditions \mathbf{e} gives the expression for U^N

$$U^N = Q^N \mathbf{e} - \sum_{i=0}^{N-1} Q^i \mathcal{D}^{-1} \mathcal{F}^{(N-1)-i} \quad (4.4)$$

where Q^n denotes the n th power of the square matrix Q .

4.2.2 Gradient of control energy

Equation (4.4) allows us to formulate a matrix expression for the gradient $\nabla J(\mathbf{e})$. Let $\mathbf{g} = \sum_{i=0}^{N-1} Q^i \mathcal{D}^{-1} \mathcal{F}^{(N-1)-i}$, which is a vector of the same dimension as \mathbf{e} . An expanded form of the control energy is then

$$\begin{aligned} J(\mathbf{e}) &= \frac{1}{2} (U^N - \mathbf{e})^T (U^N - \mathbf{e}) \\ &= \frac{1}{2} ((Q^N - I)\mathbf{e} - \mathbf{g})^T ((Q^N - I)\mathbf{e} - \mathbf{g}) \\ &= \frac{1}{2} \mathbf{e}^T (Q^N - I)^T (Q^N - I) \mathbf{e} \\ &\quad - \frac{1}{2} (\mathbf{e}^T (Q^N - I)^T \mathbf{g} - \mathbf{g}^T (Q^N - I) \mathbf{e}) \\ &\quad + \frac{1}{2} \mathbf{g}^T \mathbf{g} \\ &= \frac{1}{2} \mathbf{e}^T (Q^N - I)^T (Q^N - I) \mathbf{e} - \mathbf{g}^T (Q^N - I) \mathbf{e} + \frac{1}{2} \mathbf{g}^T \mathbf{g} \\ &=: \frac{1}{2} \mathbf{e}^T \mathbf{A} \mathbf{e} - \mathbf{b}^T \mathbf{e} + c, \end{aligned}$$

which gives us the gradient

$$\nabla J(\mathbf{e}) = \mathbf{A} \mathbf{e} - \mathbf{b} \quad (4.5)$$

where

$$\mathbf{A} = (Q^N - I)^T (Q^N - I), \quad \mathbf{b} = \mathbf{g}^T (Q^N - I).$$

4.2.3 Adjoint state method

The obvious way to compute $\nabla J(\mathbf{e})$ would be either by explicitly computing the matrix A in Eq. (4.5) or by approximating it with finite differences. However, these are both highly expensive procedures, requiring N large matrix multiplications in the first case and $\dim(\mathbf{e})$ simulations up to time T in the second. The adjoint state method provides an efficient alternative, requiring only one forward simulation to time T and one *backward* simulation back from time T to time zero. It states that

$$\frac{dJ}{d\mathbf{e}} = \frac{\partial J}{\partial \mathbf{e}} - \mathbf{z}^T \frac{\partial S}{\partial \mathbf{e}} \quad (4.6)$$

where S is the state equation (4.2) and \mathbf{z} is the solution to the *adjoint equation*

$$\left(\frac{\partial S}{\partial \mathbf{u}}\right)^T \mathbf{z} = \left(\frac{\partial J}{\partial \mathbf{u}}\right)^T. \quad (4.7)$$

$\mathbf{z} = (Z^0, \dots, Z^N)^T$ is called the *adjoint state variable* and has the same dimension as the state variable $\mathbf{u} = (U^0, \dots, U^N)^T$ in the state equation.

See Bradley (2019) for a brief tutorial on the topic and proof of the preceding statements, or Givoli (2021) for a more comprehensive introduction.

Because U^n is the only state variable that exists in the control energy J ,

$$\left(\frac{\partial J}{\partial \mathbf{u}}\right)^T = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ U^N - \mathbf{e} \end{bmatrix}.$$

Similarly, because \mathbf{e} is only present in the first row of the state equation,

$$\frac{\partial S}{\partial \mathbf{e}} = (-I, 0, 0, \dots, 0)^T$$

and thus

$$\mathbf{z}^T \frac{\partial S}{\partial \mathbf{e}} = -Z^0.$$

We also know that

$$\frac{\partial J}{\partial \mathbf{e}} = -(U^N - \mathbf{e}).$$

It follows from substituting these values into Eq. (4.6) that the gradient we're looking for is

$$\frac{dJ}{d\mathbf{e}} = -(U^N - \mathbf{e}) + Z^0. \quad (4.8)$$

To compute this we need the value of Z^0 , which can be obtained using the adjoint equation (4.7). The matrix form of this equation looks very similar to the state equation:

$$\begin{bmatrix} I & \mathcal{C}^T & & & & & & \\ & \mathcal{D}^T & \mathcal{C}^T & & & & & \\ & & \mathcal{D}^T & \mathcal{C}^T & & & & \\ & & & \ddots & \ddots & & & \\ & & & & \mathcal{D}^T & \mathcal{C}^T & & \\ & & & & & \mathcal{D}^T & & \end{bmatrix} \begin{bmatrix} Z^0 \\ Z^1 \\ \vdots \\ Z^{N-1} \\ Z^N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ U^N - \mathbf{e} \end{bmatrix} \quad (4.9)$$

where \mathcal{C} and \mathcal{D} are the block matrices from the state equation (4.2).

This equation can be solved with a backward procedure starting with

$$Z^N = (\mathcal{D}^T)^{-1}(U^N - \mathbf{e}), \quad (4.10)$$

followed by repeating

$$Z^k = -(\mathcal{D}^T)^{-1}\mathcal{C}^T Z^{k+1} \quad (4.11)$$

for $k = N - 1, \dots, 1$ and finally

$$Z^0 = -\mathcal{C}^T Z^1. \quad (4.12)$$

Written out explicitly, a single timestep according to Eq. (4.11) looks like

$$\begin{aligned} Z^k &= -(\mathcal{D}^T)^{-1}\mathcal{C}^T Z^{k+1} \\ &= - \begin{bmatrix} -I & (\Delta t \star_1^{-1} d_1^T)^T \\ 0 & -I \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ (\Delta t c^2 \star_2 d_1)^T & I \end{bmatrix} \begin{bmatrix} z_0^{k+1} \\ z_1^{k+1} \end{bmatrix}, \end{aligned}$$

which is equivalent to solving the equation

$$\begin{bmatrix} -I & (\Delta t \star_1^{-1} d_1^T)^T \\ 0 & -I \end{bmatrix} \begin{bmatrix} z_0^k \\ z_1^k \end{bmatrix} = - \begin{bmatrix} I & 0 \\ (\Delta t c^2 \star_2 d_1)^T & I \end{bmatrix} \begin{bmatrix} z_0^{k+1} \\ z_1^{k+1} \end{bmatrix},$$

producing the backward timestep formulas

$$z_1^k = z_1^{k+1} + (\Delta t c^2 \star_2 d_1)^T z_0^{k+1} \quad (4.13)$$

$$z_0^k = z_0^{k+1} + (\Delta t \star_1^{-1} d_1^T)^T z_1^k. \quad (4.14)$$

Because the state equation is solved forward in time and the adjoint state equation backward, these equations are often called the *forward problem* and *backward problem* in the context of the adjoint state method.

The above formulation again omits boundary conditions for clarity, so it is not obvious how boundary conditions should be handled in the backward problem. For Dirichlet boundary conditions (Eq. (3.15)), the lack of source terms in the adjoint equation means all Dirichlet boundaries are set to zero instead of their value in the state equation. The absorbing boundary (Eq. (3.16)), on the other hand, has a sign flipped,

$$\frac{1}{c} p - q = 0. \quad (4.15)$$

An intuitive reasoning for this is that the condition of Eq. (3.16) only allows waves propagating outwards to "pass through" it, whereas the backward problem is integrated backwards in time, requiring the boundary to instead allow through waves travelling into the domain. This is accounted for by flipping the sign (Givoli 2021).

4.3 Conjugate gradient algorithm

Equipped with the control energy (Eq. (4.1)) and an efficient procedure for computing its gradient, we're ready to solve the optimization problem minimizing the energy. This is done using the conjugate gradient (CG) algorithm, an iterative method specialized for symmetric positive definite systems, such as the ones which arise from quadratic optimization problems like this one.

Let $\mathbf{Ae} = \mathbf{b}$ be an n -dimensional symmetric positive definite linear system. The CG algorithm is based on searching along an A -conjugate sequence of directions

$$\{\mathbf{p}_0, \dots, \mathbf{p}_{n-1} : \mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0 \text{ for all } i \neq j\} \quad (4.16)$$

and making the residuals $\mathbf{r}_i = \mathbf{b} - A\mathbf{e}_i$ mutually orthogonal. The significance of these properties comes from the properties of *Krylov subspaces*, covered in detail in Saad (2003). Here we take them as granted and provide a derivation based on Saad (2003, p. 199-200).

The A -conjugate set of Eq. (4.16) is linearly independent and therefore a basis of n -dimensional space. Thus, given this set and an initial guess \mathbf{e}_0 , the exact solution \mathbf{e}_* can be expressed as a linear combination

$$\mathbf{e}_* = \mathbf{e}_0 + \sum_{i=0}^{n-1} \alpha_i \mathbf{p}_i = \mathbf{e}_j + \sum_{i=j}^{n-1} \alpha_i \mathbf{p}_i.$$

where

$$\mathbf{e}_j = \mathbf{e}_0 + \sum_{i=0}^{j-1} \alpha_i \mathbf{p}_i = \mathbf{e}_{j-1} + \alpha_{j-1} \mathbf{p}_{j-1} \quad (4.17)$$

is the estimated solution at step j and α_i is some real coefficient. It follows that the residuals follow a similar recurrence

$$\begin{aligned} \mathbf{r}_{j+1} - \mathbf{r}_j &= (\mathbf{b} - A\mathbf{e}_{j+1}) - (\mathbf{b} - A\mathbf{e}_j) \\ &= A(\mathbf{e}_j - \mathbf{e}_{j+1}) \\ &= A(-\alpha_j \mathbf{p}_j) \\ \implies \mathbf{r}_{j+1} &= \mathbf{r}_j - \alpha_j A\mathbf{p}_j. \end{aligned} \quad (4.18)$$

For the residuals to be orthogonal, we must impose

$$\langle \mathbf{r}_{j+1}, \mathbf{r}_j \rangle = \langle \mathbf{r}_j - \alpha_j A\mathbf{p}_j, \mathbf{r}_j \rangle = 0,$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product. The solution of this equation is

$$\alpha_j = \frac{\langle \mathbf{r}_j, \mathbf{r}_j \rangle}{\langle A\mathbf{p}_j, \mathbf{r}_j \rangle}. \quad (4.19)$$

α_j obtained from this equation can be applied to Eq. (4.17) and (4.18) to obtain the next approximate solution \mathbf{e}_{j+1} and residual \mathbf{r}_{j+1} . The residual can in turn be used to construct the next search direction \mathbf{p}_{j+1} as

$$\mathbf{p}_{j+1} = \mathbf{r}_{j+1} + \beta_j \mathbf{p}_j, \quad (4.20)$$

where β_j is obtained by requiring \mathbf{p}_{j+1} and \mathbf{p}_j to be A -conjugate,

$$\langle \mathbf{p}_{j+1}, A\mathbf{p}_j \rangle = \langle \mathbf{r}_{j+1} + \beta_j \mathbf{p}_j, A\mathbf{p}_j \rangle = 0,$$

implying

$$\beta_j = -\frac{\langle \mathbf{r}_{j+1}, A\mathbf{p}_j \rangle}{\langle \mathbf{p}_j, A\mathbf{p}_j \rangle}. \quad (4.21)$$

There are also the relations

$$\langle A\mathbf{p}_j, \mathbf{r}_j \rangle = \langle A\mathbf{p}_j, \mathbf{p}_j - \beta_{j-1}\mathbf{p}_{j-1} \rangle = \langle A\mathbf{p}_j, \mathbf{p}_j \rangle$$

following from Eq. (4.20) and the orthogonality of $A\mathbf{p}_j$ and \mathbf{p}_{j-1} ,

$$A\mathbf{p}_j = -\frac{1}{\alpha_j}(\mathbf{r}_{j+1} - \mathbf{r}_j)$$

following from Eq. (4.18), and finally,

$$\langle \mathbf{r}_{j+1}, A\mathbf{p}_j \rangle = \langle \mathbf{r}_{j+1}, -\frac{1}{\alpha_j}(\mathbf{r}_{j+1} - \mathbf{r}_j) \rangle = -\frac{1}{\alpha_j} \langle \mathbf{r}_{j+1}, \mathbf{r}_{j+1} \rangle$$

and

$$\langle A\mathbf{p}_j, \mathbf{p}_j \rangle = \langle -\frac{1}{\alpha_j}(\mathbf{r}_{j+1} - \mathbf{r}_j), \mathbf{r}_j \rangle = \frac{1}{\alpha_j} \langle \mathbf{r}_j, \mathbf{r}_j \rangle$$

following from the preceding two relations and the orthogonality of \mathbf{r}_{j+1} and \mathbf{r}_j . Applying these to Eq. (4.19) and (4.21) produces the coefficients

$$\alpha_j = \frac{\langle \mathbf{r}_j, \mathbf{r}_j \rangle}{\langle A\mathbf{p}_j, \mathbf{p}_j \rangle} \quad (4.22)$$

$$\beta_j = \frac{\langle \mathbf{r}_{j+1}, \mathbf{r}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{r}_j \rangle}. \quad (4.23)$$

Note also that the residual $\mathbf{r} = \mathbf{b} - A\mathbf{e}$ is the opposite of the gradient $\nabla J(\mathbf{e}) = A\mathbf{e} - \mathbf{b}$.

Combining these facts into an iterative algorithm computing successive values of $\mathbf{p}_j, \mathbf{r}_j, \mathbf{e}_j$ produces the CG algorithm. Many equivalent variations are found in literature; the following form is taken from Saad (2003). Given a linear system $A\mathbf{e} = \mathbf{b}$ and an initial guess \mathbf{e}_0 ,

- 1: $\mathbf{r}_0 \leftarrow \mathbf{b} - A\mathbf{e}_0$ ▷ Initial residual
- 2: $\mathbf{p}_0 \leftarrow \mathbf{r}_0$ ▷ Initial search direction
- 3: **for** $j = 0, 1, \dots$ **until convergence do**
- 4: $\alpha_j \leftarrow \frac{\langle \mathbf{r}_j, \mathbf{r}_j \rangle}{\langle A\mathbf{p}_j, \mathbf{p}_j \rangle}$ ▷ Coefficient to enforce orthogonality of residuals
- 5: $\mathbf{e}_{j+1} \leftarrow \mathbf{e}_j + \alpha_j \mathbf{p}_j$ ▷ Next approximate solution
- 6: $\mathbf{r}_{j+1} \leftarrow \mathbf{r}_j - \alpha_j A\mathbf{p}_j$ ▷ Next residual

- 7: $\beta_j \leftarrow \frac{\langle \mathbf{r}_{j+1}, \mathbf{r}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{r}_j \rangle}$ \triangleright Coefficient to enforce A -conjugacy of search directions
8: $\mathbf{p}_{j+1} \leftarrow \mathbf{r}_{j+1} + \beta_j \mathbf{p}_j$ \triangleright Next search direction
9: **end for**

Recall that the adjoint state method produces a value for $\nabla J(\mathbf{e}) = A\mathbf{e} - \mathbf{b}$ (Eq. (4.5)), where \mathbf{b} is a vector that depends on source terms and A a matrix that does not. This gives us a way to compute the terms in the algorithm where multiplication by A occurs, namely \mathbf{r}_0 and $A\mathbf{p}_j$.

First, $\mathbf{r}_0 = \mathbf{b} - A\mathbf{e}_0 = -\nabla J(\mathbf{e}_0)$, which is given directly by the adjoint state method. Secondly, $A\mathbf{p}_j$ is $\nabla J(\mathbf{p}_j)$ without the vector \mathbf{b} . Because \mathbf{b} depends on the source terms and A does not, we can obtain this value by running the adjoint state method with source terms set to zero, causing \mathbf{b} to be multiplied by zero and leaving A unchanged.

These procedures allow us to perform the CG algorithm by computing one forward simulation and one backward simulation per iteration.

5 Experiments

A number of numerical experiments were performed to test the effectiveness of the techniques covered in preceding chapters. The accuracy of DEC with and without harmonic timestep operators was investigated using a simple plane wave with a known analytic solution, the CFL condition was estimated by progressively refining a mesh, and the convergence rate of the controllability method was compared with the asymptotic iteration method in various geometries.

5.1 Implementation

The programming language of choice for the experiments' implementation was Python due to the availability of the PyDEC software library (Bell and Hirani 2012). This library handles the generation of a dual mesh with circumcentric dual vertices as well as the exterior derivative and Hodge operators, allowing us to focus on implementing our specific acoustics model and the controllability method.

PyDEC can generate a dual mesh when given a primal mesh, but it does not provide a way to generate the primal mesh. To this end, the finite element mesh generator Gmsh (Geuzaine and Remacle 2009) was used via its programmatic Python interface.

The source code of the experiments is available in a public GitHub repository (Myyrä 2023). Instructions for running the code and details of the code structure are provided in the repository, as well as animated versions of some of the figures in this chapter. Additional experiments built for the purpose of learning to use DEC and verifying parts of the implementation are also present.

5.2 Plane wave

In the first experiment, we simulate a plane wave

$$\phi_{inc} = -\cos(\omega t - \vec{\mathbf{K}} \cdot \mathbf{x}), \quad (5.1)$$

where ω is the angular velocity of the wave, t is time, $\vec{\kappa}$ is the wave vector (see Section 3.5.2), and \mathbf{x} is the spatial position. This wave propagates unimpeded through a square domain Ω represented as an unstructured triangle mesh. The scenario is illustrated in Figure 8, where the color of a face represents pressure at the dual vertex, and the arrows represent velocity interpolated from the primal edge fluxes using Whitney forms (see Section 2.3.5).

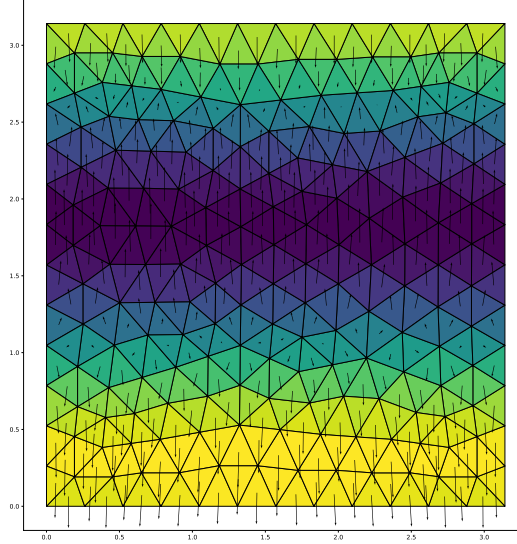


Figure 8. A plane wave propagating upwards in a square domain.

It can be shown by substituting the incident wave equation (5.1) into the second-order acoustic wave equation (3.1) that it is a solution of the wave equation. This knowledge is applied to test the accuracy of our simulation. We apply the incident wave equation (5.1) on the boundary Γ as a Dirichlet boundary condition

$$q = \star(\nabla\phi_{inc})^{\flat} \quad \text{in } \Gamma \times [0, T] \quad (5.2)$$

(as derived in Section 3.4), set the initial conditions to

$$p_0 = \frac{\partial\phi_{inc}}{\partial t} \Big|_{t=0} \quad (5.3)$$

$$q_0 = \star(\nabla\phi_{inc})^{\flat} \Big|_{t=0}, \quad (5.4)$$

and simulate the propagation of the wave in the interior of the domain with DEC.

This simulation is performed several times with progressively reducing edge lengths on the mesh to investigate the effect of mesh element size on the accuracy. Error is measured

by comparing the simulated values P and Q to the exact cochains P_* and Q_* obtained by integrating $p = \frac{\partial \phi_{inc}}{\partial t}$ and $q = \star(\nabla \phi_{inc})^b$ over the corresponding mesh elements. Because the flux Q is integrated over mesh edges, the error in Q is divided by edge length to normalize the effect of varying edge lengths on the result.

This experiment is repeated twice, swapping Yee's Hodge star for the harmonic operators (Eq. (3.27) and (3.29)) derived in Section 3.5.2 and the timestep equations (3.12) and (3.13) for the exact equivalents (Eq. (3.22) and (3.23)) derived in Section 3.5.1. The maximum elements of error in pressure and flux are plotted in Figure 9. The results presented are obtained with the wave parameters $\omega = 2$ and $\vec{k} = (0, 2)$.

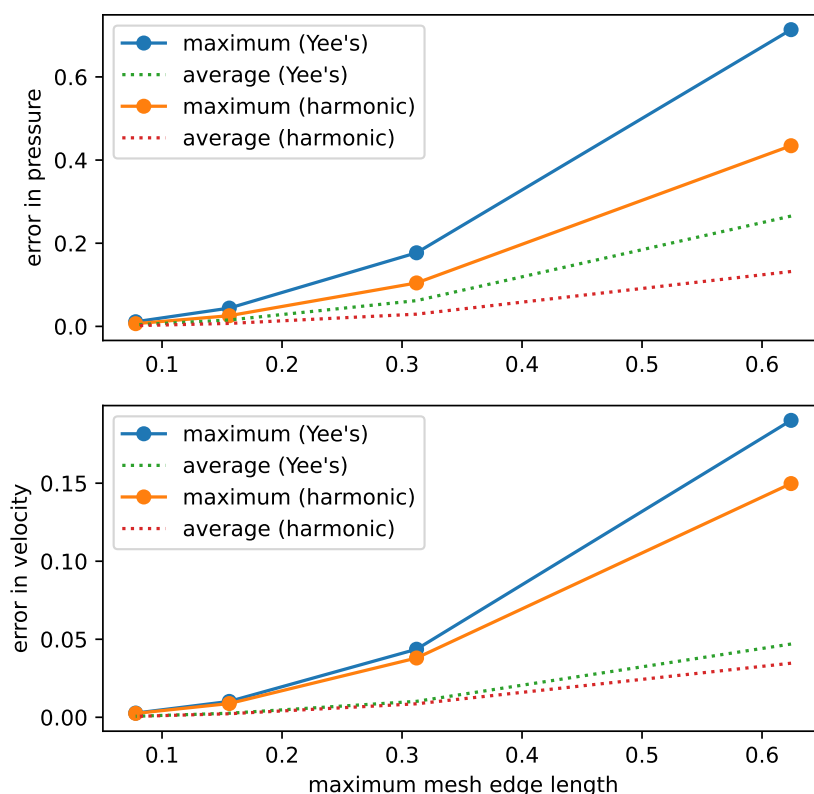


Figure 9. Maximum and average error of the simulated plane wave in relation to maximum edge length of the mesh.

The results show a reduction in error of approximately 40% in pressure and varying between

8% and 20% in velocity for the harmonic operators. The harmonic operators add no additional computational cost to the timestepping procedure, since the Hodge star is still diagonal and the exact timestep coefficient is still a scalar. Their only cost is a small amount of extra computation beforehand, which can be performed offline alongside mesh generation if desired. Thus, they are a highly advantageous addition to the method.

To examine the proportional effects of the harmonic Hodge star and the exact timestep, note that Eq. (3.12) and (3.22) only differ by a scalar coefficient. The timestep Δt in Eq. (3.12) is replaced by $\tau_{ex} = \frac{2}{\omega} \sin \frac{\omega \Delta t}{2}$. Figure 10 plots this value as a function of the angular velocity ω with a few selected values of Δt . We find that the difference between Δt and τ_{ex} can be significant at high values of Δt and ω , but quickly becomes negligible as Δt becomes smaller.

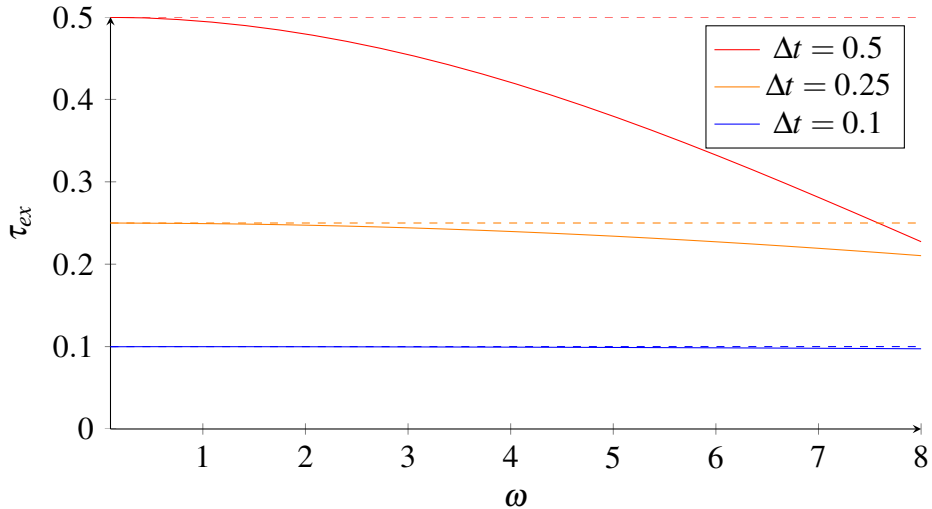


Figure 10. Effect of angular velocity on the exact timestep coefficient.

In our experiments stability generally requires values of Δt below 0.1, making the effect of the exact timestep minimal. However, as it requires practically no extra computation, there is no penalty to implementing it, and it is potentially beneficial in cases where stability permits large timesteps.

The CFL condition limits the magnitude of the term $\frac{\Delta t}{|\mathcal{E}|}$, where $|\mathcal{E}|$ is the length of the shortest edge in the computation mesh. This term can be kept constant by increasing the timestep and minimum edge length in the same proportion. In other words, larger timesteps can be used if a coarser mesh is also used. Thus, in scenarios without complex geometry or

short wavelenghts demanding detailed meshes, exact timestepping may enable one to reduce mesh detail and increase timestep size without a significant loss of accuracy, resulting in more efficient simulations.

5.3 CFL condition

If the CFL condition is known, a stable value of Δt can be selected automatically as a multiple of the shortest edge length, $\Delta t = c|\mathcal{E}|$. To find an estimation for the coefficient c numerically, a mesh was constructed and then refined several times by cutting each edge in half and connecting the midpoints. Figure 11 illustrates the initial mesh in black and the refinements in progressively lighter shades of blue.

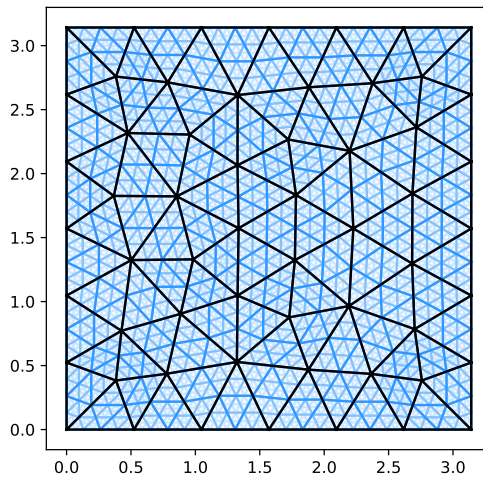


Figure 11. Mesh used to examine the CFL condition, overlaid with its refinements.

On each of these meshes, a plane wave was applied on the boundary in the same way as in the previous experiment. A binary search was performed on the timestep Δt to find the highest value where the simulation remains stable (i.e. the maximum absolute values of the simulation variables do not increase). The results are plotted in Figure 12, where h denotes the shortest edge length.

The line $\Delta t = 0.3h$ fits safely below the upper limit for every refinement tested, which suggests that we can use $\Delta t = 0.3h$ in our other experiments. However, in practice this coeffi-

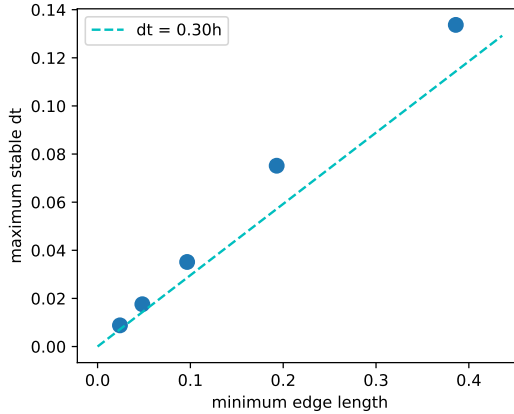


Figure 12. Maximum stable value of Δt plotted against the shortest edge length.

cient was found to depend significantly on the mesh geometry, and a much lower value of $\Delta t = 0.1h$ was selected.

5.4 Scatterer

Our main experiment uses the entirety of the model defined in Chapter 3, as well as the controllability method covered in Chapter 4. The computation mesh is constructed with two boundaries as illustrated in Figure 6, with the inner boundary Γ_{sca} representing a scattering obstacle and absorbing outer boundary Γ_{ext} . The incident wave, which we set to the plane wave equation (3.24), is applied on Γ_{sca} as a Dirichlet boundary condition.

The resulting simulation represents the scattered component ϕ_{sca} of the total wave $\phi_{inc} + \phi_{sca}$. We can reconstruct the total wave by adding ϕ_{inc} to the result. By omitting the incident wave from the simulation, we gain accuracy by eliminating numerical error from the component of the wave whose exact solution is known.

It is difficult to come up with initial conditions that fulfill both boundary conditions. If we set $p_0 = 0$, $q_0 = 0$, the incident wave is not matched on Γ_{sca} . If we set $p_0 = \frac{\partial \phi_{inc}}{\partial t} \Big|_{t=0}$, $q_0 = \star(\nabla \phi_{inc})^b \Big|_{t=0}$ like in the previous experiment, the absorbing condition is not fulfilled on Γ_{ext} . This has the effect of a discontinuity on the first timestep of simulation, which may cause stability issues in particularly unfortunate cases.

To avoid this discontinuity, we use an easing procedure proposed by Mur (1993). Initial conditions are set to $p_0 = 0$, $q_0 = 0$, and source terms (i.e. values on Γ_{sca}) are also initially set to zero. Source terms are then eased in over a time period t_{tr} by multiplying them by

$$f_{tr}(t) = (2 - \sin(\frac{\pi t}{2 t_{tr}})) \sin(\frac{\pi t}{2 t_{tr}}). \quad (5.5)$$

This way we get a smooth solution in the entire space and time domain. To apply this to the controllability method, we simulate forward in time until $t = t_{tr}$ and use the result as the initial guess for the CG algorithm.

We investigate the convergence of the controllability method and the asymptotic forward-in-time iteration method with different scatterer geometries by measuring the control energy (Eq. (4.1)) over time. Simulating forward one time period T is considered equivalent to a single step of the controllability method. We also compare the performance of the Yee's and harmonic operators in both iteration algorithms. The scatterer geometries and time-periodic scattered waves obtained from the controllability method are visualized alongside their respective convergence results in figures 13–16. The twelve-pointed star case (fig. 15) has wavenumber $\kappa = 2$, while $\kappa = 1$ in the rest of the cases.

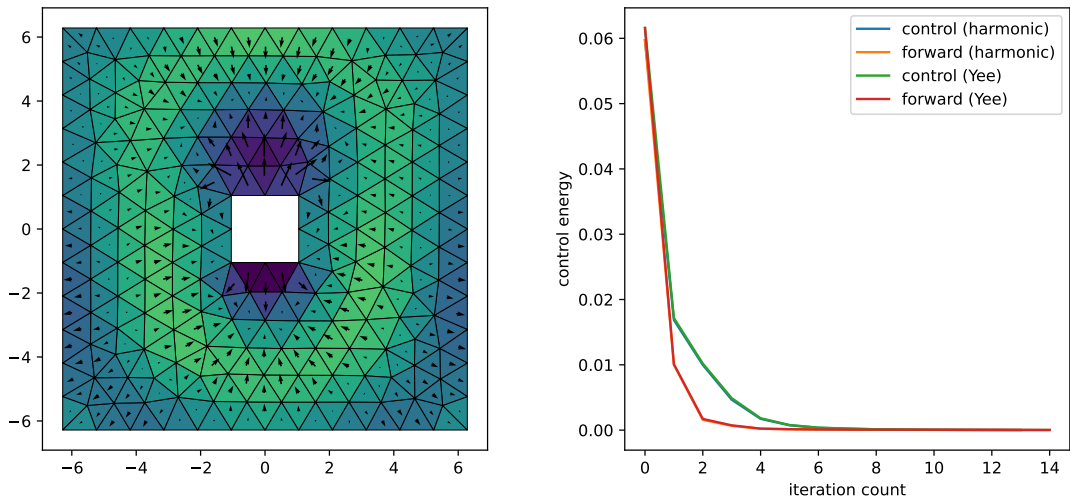


Figure 13. Scatterer, case 1: square

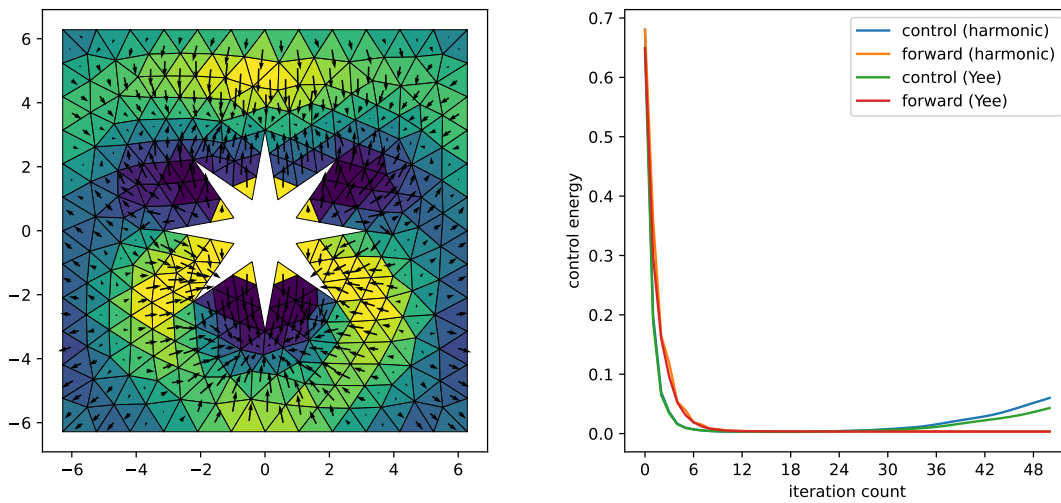


Figure 14. Scatterer, case 2: eight-pointed star

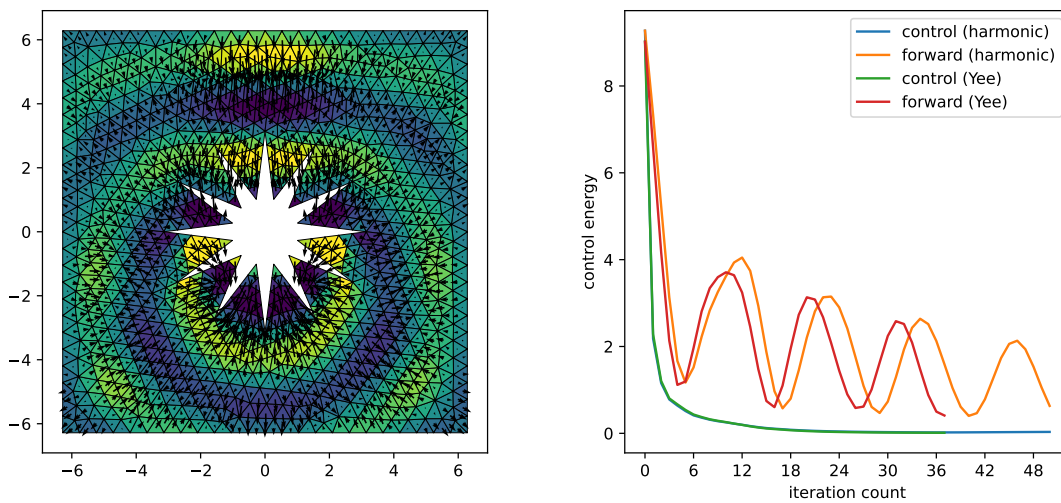


Figure 15. Scatterer, case 3: twelve-pointed star

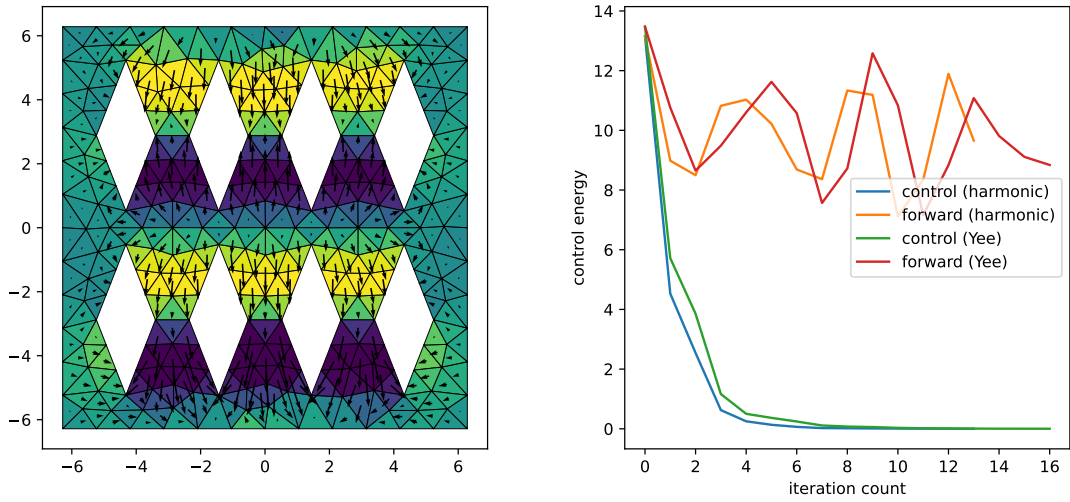


Figure 16. Scatterer, case 4: diamonds

In the square case we find that the asymptotic iteration method outperforms the controllability method, and in the eight-pointed star case it has no strong advantage. The square is entirely convex, so the scattered wave has no obstacles to slow it down and reaches the absorbing boundary quickly. The star is not fully convex, but still allows waves to exit with little obstruction. In these cases, the optimization approach is not efficient enough to improve on the rate of convergence.

On the other hand, in the twelve-pointed star and diamonds cases, where a higher degree of nonconvexity is present and waves must navigate narrow passages to find the absorbing boundary, the asymptotic iteration method's performance degrades significantly. In the diamonds case it fails to reduce energy at all. The controllability method, on the other hand, loses little efficiency in these scenarios.

The effects of the harmonic operators on convergence are inconclusive. They lead to slightly faster convergence in some cases, but have no advantage in others. However, it should be noted that the computation of the control energy (Eq. (4.1)) involves simulating forward in time, which depends on the timestep operators used. This means that while both operators tend to converge to the same level of energy, the solutions they converge to are different. Based on the results of the plane wave experiment, it can be assumed the harmonic solution

is more accurate.

We also note that in the eight-pointed star case, the controllability method's solution fails to converge to the desired level and the energy begins to increase. This seems to occur when the computation mesh is of poor quality. In these cases the mesh produces a high amount of error in the adjoint state solution when the gradient is close to zero, causing the CG algorithm's invariants of orthogonal residuals and A -conjugate search directions to be violated. Unless the mesh quality is poor enough to cause instability, this does not occur in the asymptotic iteration.

The quality of a mesh can be roughly estimated by investigating the diagonal values of the Hodge star matrices. In particular, extremely low values of \star_1 arising from short dual edges generate large numerical errors. Values of \star_1 can even become negative when a dual vertex lies outside its corresponding primal face, and meshes where this happens were found to give rise to unstable simulations.

5.5 Discussion

The purpose of these experiments was to investigate the effectiveness of the harmonic time-step operators and the controllability method. The harmonic timestep operators, especially the Hodge star, offer a notable advantage in accuracy over Yee's equivalents at a small computational cost, so their usage can be recommended in all harmonic problems.

The controllability method's results are less clear-cut. It has a significant advantage in cases with highly nonconvex geometry, but suffers compared to the asymptotic iteration method when the geometry is mostly convex. It also seems to be more sensitive to poor mesh quality than the asymptotic method. Its usage should therefore be considered on a case by case basis.

Mesh quality was found to be a major factor in the performance of DEC-based simulations, and our results likely suffer from the use of a mesh generator not specifically designed for DEC. The circumcentric dual mesh is particularly sensitive to narrow triangles with obtuse angles, which cause the circumcenter to lie outside the triangle. This produces very small or negative values in the Hodge star matrix, causing inaccuracy at best and instability at

worst. In future work, the regular crystal meshes of Rabinä (2014) and Hodge-optimized triangulations of Mullen et al. (2011) should be investigated.

While these experiments were performed in \mathbb{R}^2 , one of the advantages of DEC is straightforward generalization to higher dimensions. The main challenges of generalization come from generating the primal mesh and deriving the harmonic Hodge operators, which needs to be done for every distinct pair of k - and $(n - k)$ -cells. We expect these methods to have similar characteristics and tradeoffs in higher dimensions.

6 Conclusion

In this thesis, a numerical method based on discrete exterior calculus (DEC) and exact controllability was developed for the purpose of finding time-periodic solutions to acoustic wave scattering problems. Additionally, a timestepping procedure and Hodge operator optimized for harmonic problems were investigated. The harmonic operators were found highly effective, while the controllability method was only more efficient than the simpler asymptotic iteration method in cases with highly nonconvex geometry. To improve the performance and reliability of the methods, future development should focus on the quality of the computation mesh, as the generation methods employed here are prone to inaccuracies and instabilities.

Bibliography

Abraham, Ralph, Jerrold E. Marsden, and Tudor Ratiu. 2012. *Manifolds, Tensor Analysis, and Applications*. Springer Science & Business Media, December 6, 2012. ISBN: 978-1-4612-1029-0.

Bell, Nathan, and Anil N. Hirani. 2012. “PyDEC: Software and Algorithms for Discretization of Exterior Calculus.” *ACM Transactions on Mathematical Software* 39, no. 1 (November 1, 2012): 3:1–3:41. ISSN: 0098-3500, accessed December 4, 2022. <https://doi.org/10.1145/2382585.2382588>.

Blair Perot, J., and Christopher J. Zusi. 2014. “Differential forms for scientists and engineers.” *Journal of Computational Physics, Physics-compatible numerical methods*, 257 (January 15, 2014): 1373–1393. ISSN: 0021-9991, accessed October 10, 2022. <https://doi.org/10.1016/j.jcp.2013.08.007>.

Bochev, Pavel B., and James M. Hyman. 2006. “Principles of Mimetic Discretizations of Differential Operators.” In *Compatible Spatial Discretizations*, edited by Douglas N. Arnold, Pavel B. Bochev, Richard B. Lehoucq, Roy A. Nicolaides, and Mikhail Shashkov, 89–119. The IMA Volumes in Mathematics and its Applications. New York, NY: Springer. ISBN: 978-0-387-38034-6. https://doi.org/10.1007/0-387-38034-5_5.

Bossavit, A. 2001. “‘Generalized Finite Differences’ in Computational Electromagnetics.” *Progress In Electromagnetics Research* 32:45–64. ISSN: 1559-8985, accessed July 6, 2023. <https://doi.org/10.2528/PIER00080102>.

Bossavit, A., and L. Kettunen. 1999. “Yee-like schemes on a tetrahedral mesh, with diagonal lumping.” *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 12 (1): 129–142. ISSN: 1099-1204, accessed July 6, 2023. [https://doi.org/10.1002/\(SICI\)1099-1204\(199901/04\)12:1/2<129::AID-JNM327>3.0.CO;2-G](https://doi.org/10.1002/(SICI)1099-1204(199901/04)12:1/2<129::AID-JNM327>3.0.CO;2-G).

Bossavit, Alain. 1998. “On the geometry of electromagnetism.” *J. Japan Soc. Appl. Electromagn. & Mech* 6:17–28.

Bossavit, Alain. 1999. "Computational electromagnetism and geometry: Building a finite-dimensional" Maxwell's house"-1-Network equations." *Journal of the Japan Society of Applied Electromagnetics and Mechanics* 7 (2): 150–159.

———. 2000. "Computational electromagnetism and geometry (5) The" Galerkin hodge"." *Journal of the Japan Society of Applied Electromagnetics and Mechanics* 8 (2): 203–209.

———. 2005. "Discretization of Electromagnetic Problems: The "Generalized Finite Differences" Approach." In *Handbook of Numerical Analysis*, 13:105–197. Elsevier. ISBN: 978-0-444-51375-5, accessed February 8, 2023. [https://doi.org/10.1016/S1570-8659\(04\)13002-0](https://doi.org/10.1016/S1570-8659(04)13002-0). <https://linkinghub.elsevier.com/retrieve/pii/S1570865904130020>.

Bradley, Andrew M. 2019. "PDE-constrained optimization and the adjoint method," October 15, 2019. Accessed September 23, 2022. https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf.

Bristeau, M. O., R. Glowinski, and J. Périaux. 1998. "Controllability Methods for the Computation of Time-Periodic Solutions; Application to Scattering." *Journal of Computational Physics* 147, no. 2 (December 10, 1998): 265–292. ISSN: 0021-9991, accessed February 18, 2023. <https://doi.org/10.1006/jcph.1998.6044>.

Courant, R., K. Friedrichs, and H. Lewy. 1967. "On the Partial Difference Equations of Mathematical Physics." Conference Name: IBM Journal of Research and Development, *IBM Journal of Research and Development* 11, no. 2 (March): 215–234. ISSN: 0018-8646. <https://doi.org/10.1147/rd.112.0215>.

Crane, Keenan, Fernando de Goes, Mathieu Desbrun, and Peter Schröder. 2013. "Digital geometry processing with discrete exterior calculus." In *ACM SIGGRAPH 2013 Courses*, 1–126. SIGGRAPH '13. New York, NY, USA: Association for Computing Machinery, July 21, 2013. ISBN: 978-1-4503-2339-0, accessed November 5, 2022. <https://doi.org/10.1145/2504435.2504442>.

Desbrun, Mathieu, Anil N. Hirani, Melvin Leok, and Jerrold E. Marsden. 2005. *Discrete Exterior Calculus*. Number: arXiv:math/0508341, arXiv:math/0508341, August 18, 2005. Accessed September 13, 2022. <https://doi.org/10.48550/arXiv.math/0508341>. arXiv:math/0508341.

Desbrun, Mathieu, Eva Kanso, and Yiyong Tong. 2006. “Discrete Differential Forms for Computational Modeling.” *Discrete Differential Geometry*.

Engquist, Bjorn, and Andrew Majda. 1977. “Absorbing boundary conditions for the numerical simulation of waves.” *Mathematics of Computation* 31 (139): 629–651. ISSN: 0025-5718, 1088-6842, accessed January 1, 2023. <https://doi.org/10.1090/S0025-5718-1977-0436612-4>.

Geuzaine, Christophe, and Jean-Francois Remacle. 2009. “Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities.” *International Journal for Numerical Methods in Engineering* 79:1309–1331. ISSN: 00295981. http://gmsh.info/doc/preprints/gmsh_paper_preprint.pdf.

Givoli, Dan. 2021. “A tutorial on the adjoint method for inverse problems.” *Computer Methods in Applied Mechanics and Engineering* 380 (July 1, 2021): 113810. ISSN: 0045-7825, accessed March 17, 2023. <https://doi.org/10.1016/j.cma.2021.113810>.

Hirani, A. N., K. B. Nakshatrala, and J. H. Chaudhry. 2015. “Numerical Method for Darcy Flow Derived Using Discrete Exterior Calculus.” *International Journal for Computational Methods in Engineering Science and Mechanics* 16, no. 3 (May 4, 2015): 151–169. ISSN: 1550-2287, accessed January 22, 2023. <https://doi.org/10.1080/15502287.2014.977500>.

Kähkönen, Sami, Roland Glowinski, Tuomo Rossi, and Raino a. E. Mäkinen. 2011. “Solution of time-periodic wave equation using mixed finite elements and controllability techniques.” Publisher: World Scientific Publishing Co. *Journal of Computational Acoustics* 19, no. 4 (December): 335–352. ISSN: 0218-396X, accessed June 18, 2023. <https://doi.org/10.1142/S0218396X11004377>.

Kivioja, Markus, Roberto Zamora-Zamora, Alina Blinova, Sanna Mönkölä, Tuomo Rossi, and Mikko Möttönen. 2023. “Evolution and decay of an Alice ring in a spinor Bose-Einstein condensate.” Publisher: American Physical Society, *Physical Review Research* 5, no. 2 (May 15, 2023): 023104. Accessed July 6, 2023. <https://doi.org/10.1103/PhysRevResearch.5.023104>.

Lee, John M. 2012. *Introduction to Smooth Manifolds*. Vol. 218. Graduate Texts in Mathematics. New York, NY: Springer. ISBN: 978-1-4419-9981-8 978-1-4419-9982-5, accessed January 15, 2023. <https://doi.org/10.1007/978-1-4419-9982-5>. <http://link.springer.com/10.1007/978-1-4419-9982-5>.

Lions, J. L. 1988. “Exact Controllability, Stabilization and Perturbations for Distributed Systems.” Publisher: Society for Industrial and Applied Mathematics, *SIAM Review* 30, no. 1 (March): 1–68. ISSN: 0036-1445, accessed February 18, 2023. <https://doi.org/10.1137/1030001>.

Lohi, Jonni. 2023. “Higher order approximations in discrete exterior calculus.” Accepted: 2023-05-22T13:54:14Z ISBN: 9789513996130 Publisher: Jyväskylän yliopisto, *JYU Dissertations*, accessed June 14, 2023. <https://jyx.jyu.fi/handle/123456789/87097>.

Lohi, Jonni, and Lauri Kettunen. 2021. “Whitney forms and their extensions.” *Journal of Computational and Applied Mathematics* 393 (September 1, 2021): 113520. ISSN: 0377-0427, accessed February 1, 2023. <https://doi.org/10.1016/j.cam.2021.113520>.

Ma, Changning, and Zhizhang Chen. 2005. “Stability and numerical dispersion analysis of CE-FDTD method.” Conference Name: IEEE Transactions on Antennas and Propagation, *IEEE Transactions on Antennas and Propagation* 53, no. 1 (January): 332–338. ISSN: 1558-2221. <https://doi.org/10.1109/TAP.2004.838760>.

Mönkölä, Sanna, Erkki Heikkola, Anssi Pennanen, and Tuomo Rossi. 2008. “Time-harmonic elasticity with controllability and higher-order discretization methods.” Accepted: 2012-10-11T11:02:30Z Publisher: Elsevier, *Journal of Computational Physics* 227 (11). Accessed June 18, 2023. <https://doi.org/10.1016/j.jcp.2008.01.054>.

Mönkölä, Sanna, and Joona Rätty. 2022. “Discrete exterior calculus for photonic crystal waveguides.” *International Journal for Numerical Methods in Engineering* (November 21, 2022): nme.7144. ISSN: 0029-5981, 1097-0207, accessed December 4, 2022. <https://doi.org/10.1002/nme.7144>.

Mullen, Patrick, Pooran Memari, Fernando de Goes, and Mathieu Desbrun. 2011. “HOT: Hodge-optimized triangulations.” In *ACM SIGGRAPH 2011 papers*, 1–12. SIGGRAPH ’11. New York, NY, USA: Association for Computing Machinery, July 25, 2011. ISBN: 978-1-4503-0943-1, accessed June 22, 2023. <https://doi.org/10.1145/1964921.1964998>.

Mur, G. 1993. “The finite-element modeling of three-dimensional electromagnetic fields using edge and nodal elements.” Conference Name: IEEE Transactions on Antennas and Propagation, *IEEE Transactions on Antennas and Propagation* 41, no. 7 (July): 948–953. ISSN: 1558-2221. <https://doi.org/10.1109/8.237627>.

Myyrä, Mikael. 2023. “wavepool,” May 26, 2023. GitHub repository. Accessed June 21, 2023. <https://github.com/m0lentum/wavepool>.

Nitschke, Ingo, Sebastian Reuther, and Axel Voigt. 2017. “Discrete exterior calculus (DEC) for the surface Navier-Stokes equation,” 177–197. Accessed January 14, 2023. https://doi.org/10.1007/978-3-319-56602-3_7. arXiv: 1611.04392[physics]. <http://arxiv.org/abs/1611.04392>.

Peng, Zhichao, and Daniel Appelö. 2022. “EM-WaveHoltz: A Flexible Frequency-Domain Method Built From Time-Domain Solvers.” Conference Name: IEEE Transactions on Antennas and Propagation, *IEEE Transactions on Antennas and Propagation* 70, no. 7 (July): 5659–5671. ISSN: 1558-2221. <https://doi.org/10.1109/TAP.2022.3161448>.

Räbinä, Jukka. 2014. “On a numerical solution of the Maxwell equations by discrete exterior calculus,” 144.

Räbinä, Jukka, Lauri Kettunen, Sanna Mönkölä, and Tuomo Rossi. 2018. “Generalized wave propagation problems and discrete exterior calculus.” Number: 3 Publisher: EDP Sciences, *ESAIM: Mathematical Modelling and Numerical Analysis* 52, no. 3 (May 1, 2018): 1195–1218. ISSN: 0764-583X, 1290-3841, accessed February 23, 2023. <https://doi.org/10.1051/m2an/2018017>.

Räbinä, Jukka, Pekko Kuopanportti, Markus I. Kivioja, Mikko Möttönen, and Tuomo Rossi. 2018. “Three-dimensional splitting dynamics of giant vortices in Bose-Einstein condensates.” Publisher: American Physical Society, *Physical Review A* 98, no. 2 (August 22, 2018): 023624. Accessed July 6, 2023. <https://doi.org/10.1103/PhysRevA.98.023624>.

Räbinä, Jukka, Sanna Mönkölä, and Tuomo Rossi. 2015. “Efficient Time Integration of Maxwell’s Equations with Generalized Finite Differences.” Publisher: Society for Industrial and Applied Mathematics, *SIAM Journal on Scientific Computing* 37, no. 6 (January): B834–B854. ISSN: 1064-8275, accessed February 23, 2023. <https://doi.org/10.1137/140988759>.

Rossi, Tuomo, Jukka Räbinä, Sanna Mönkölä, Sampsa Kiiskinen, Jonni Lohi, and Lauri Kettunen. 2021. “Systematisation of Systems Solving Physics Boundary Value Problems.” In *Numerical Mathematics and Advanced Applications ENUMATH 2019*, edited by Fred J. Vermolen and Cornelis Vuik, 35–51. Lecture Notes in Computational Science and Engineering. Cham: Springer International Publishing. ISBN: 978-3-030-55874-1. https://doi.org/10.1007/978-3-030-55874-1_3.

Saad, Yousef. 2003. *Iterative Methods for Sparse Linear Systems*. Other Titles in Applied Mathematics. Society for Industrial / Applied Mathematics, January. ISBN: 978-0-89871-534-7, accessed June 18, 2023. <https://doi.org/10.1137/1.9780898718003>. <https://epubs.siam.org/doi/book/10.1137/1.9780898718003>.

Whitney, Hassler. 1957. *Geometric Integration Theory*.

Yee, Kane. 1966. “Numerical solution of initial boundary value problems involving maxwell’s equations in isotropic media.” Conference Name: IEEE Transactions on Antennas and Propagation, *IEEE Transactions on Antennas and Propagation* 14, no. 3 (May): 302–307. ISSN: 1558-2221. <https://doi.org/10.1109/TAP.1966.1138693>.

Young, Peter. 2014. “The leapfrog method and other “symplectic” algorithms for integrating Newton’s laws of motion” (April 21, 2014). <https://young.physics.ucsc.edu/115/leapfrog.pdf>.