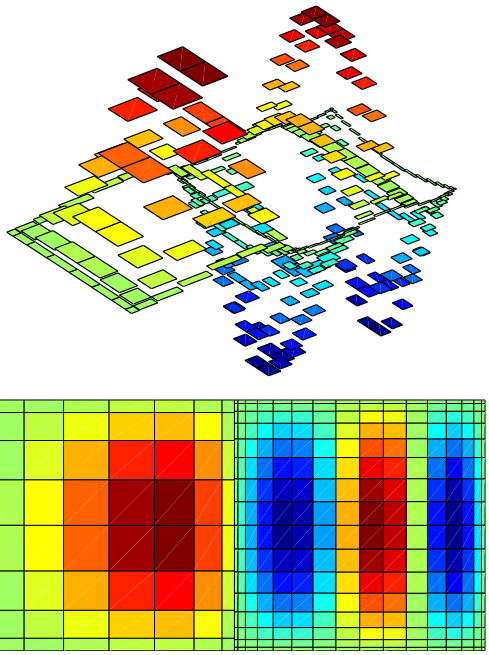
Delft University of Technology
Master of Science Thesis

Mimetic Mesh Refinement

A mortar element approach



Peter Kuystermans
1265830

Faculty of Aerospace Engineering
Delft University of Technology

Mimetic Mesh Refinement
A mortar element approach

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering at Delft University of Technology

Peter Kuystermans

August 14, 2012

Faculty of Aerospace Engineering · Delft University of Technology

Delft University of Technology

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF AERODYNAMICS

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance the thesis entitled **Mimetic Mesh Refinement** by **Peter Kuystermans** in fulfilment of the requirements for the degree of **Master of Science**.

Dated: August 14, 2012

Supervisors:

Dr.ir. M.I. Gerritsma

Dr. R.P. Dwight

Ir. J.J. Kreeft

Ir. P.J. Pinto Rebelo

PREFACE

I would like to express my thanks to my supervisor Marc Gerritsma for his time and patience. His guidance and support proved to be invaluable in the completion of my thesis. I would also like to thank Jasper Kreeft for the many times he helped me. Whenever I was stuck with a derivation, he always took the time to explain and make sure I could continue. Pedro and Deepesh also deserve my appreciation. Even though on numerous occasions I bothered them with a lot of questions, they were always willing to answer them. We also had a lot of fun, something for which I am grateful too. I enjoyed my time with the students in the basement, even though I am happy it is now finally over. It has taken long enough. In the end, none of this would have been possible without the unending support from my parents. Something for which I am extremely thankful. I simply could not have done this without them.

Peter Kuystermans

ABSTRACT

This thesis aims to introduce mesh refinement into the Mimetic Spectral Element Method (MSEM). The concept of mimetic discretizations is to mimic the properties of continuous Partial Differential Equations (PDEs) discretely. In many discretization methods information is lost in the actual discretization step which is detrimental to the physical fidelity of the approximated solution. Mimetic methods try to prevent this, a feat achieved by combining the fields of differential geometry and algebraic topology. Where differential geometry describes the continuous problem, algebraic topology functions as its discrete equivalent. By accounting for the spatial and temporal geometric objects each physical quantity is associated with, mimetic methods preserve as much as possible of the continuous structure.

Upon mesh refinement (whether h- or p-refinement; h-refinement splits elements up into smaller ones while p-refinement increases element order) a discrepancy arises between coarser and finer parts of the mesh. To resolve this problem an approach is introduced based on the Mortar Element Method (MEM). This introduces so-called mortar elements with which to connect these parts. As the scalar Laplacian in differential geometry gives rise to a 0-form and 2-form Poisson problem, the new approach employs the 0-forms (nodes) and 1-forms (edges) present at the shared boundary. In both cases a mortar element is used to approximate a mortar solution which is based on the solutions at the boundaries of the coarse and fine parts it connects. Solving the system returns the mortar solution from which, with the help of a projection matrix, the solution at the shared boundary can be retrieved.

The results achieved with the mimetic approach to the MEM are promising. In the test cases considered, in particular for p-refinement, accuracy was improved given less degrees of freedom. Conditioning and sparsity of the system matrix were only slightly affected. One test case in particular for p-refinement showed a significant improvement. This involved approximating the maximum of a concentrated peak located somewhere in the computational domain. For h-refinement the method has been shown to work as well, although results were less convincing compared to p-refinement. This is most likely caused by the choice of the test case. In conclusion it can be said that the developed approach to mimetic mesh refinement works. It provides for increased flexibility, accuracy as well as performance. There are however plenty of opportunities for future work to improve upon this method.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

MSEM Mimetic Spectral Element Method vii
PDE Partial Differential Equation vii
MEM Mortar Element Method vii
GLL Gauss-Lobatto Legendre xi
SEM Spectral Element Method 1
CFD Computational Fluid Dynamics 1
FEM Finite Element Method 37

CHAPTER 1

INTRODUCTION

People working in the field of Computational Fluid Dynamics (CFD) are in the business of solving PDEs. How they do this varies significantly, as a plethora of discretization techniques exist. Besides the classical finite volume, finite element and finite difference methods and its many variations, altogether different methods are being used as well. In some cases methods overlap or are combined to create a new one. In fact, the method used in this work finds its origin in the Spectral Element Method (SEM), a combination of finite elements and spectral methods. However, one other ingredient is added to the mix to make it even better. This ingredient is the mimetic approach resulting in what is called the MSEM. However, before diving into mimetic methods and explaining the nuts and bolts that make it work, the focus of this thesis will be on mesh refinement and its implementation into the MSEM. Nevertheless, as both aspects are still equally important to the success of this work, a short introduction to both is presented here.

1.1 Mesh refinement

Starting with mesh refinement, the principle behind it is very straightforward. Simply put, it allows the computational mesh to be refined in areas of interest while in other parts coarsened as not to waste computation power. An example in which mesh refinement has been applied is the reactor coolant flow depicted in Figure 1.1. Close to the boundary of the surface the magnitude

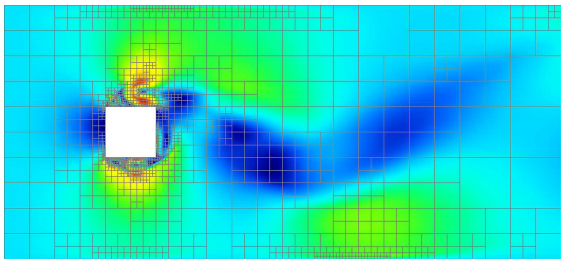


Figure 1.1 – Reactor coolant flow showing the velocity magnitude (hpfem.org).

of the velocity fluctuates considerably, hence more refinement is found there than further away

from the reactor. To give an example closer to home, the boundary layer of an airfoil often requires refinement as well.

Mesh refinement introduces a number of challenges. One of these challenges is the connection between finer and coarser parts of the mesh. To see more closely what happens, consider figure 1.2. This mesh can be seen to consist of a coarse (left) half and a fine (right) half, resulting in

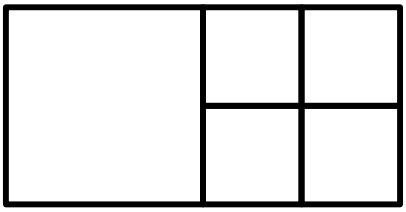


Figure 1.2 – A non-conforming mesh.

what is called a non-conforming mesh. That is just a formal way of saying that at the connection between these two parts not every node or vertex is shared resulting in a 'hanging node'. Indeed, in this example a discrepancy exists at the mid-section in the number of nodes. To see why this might pose a problem, consider the heat equation as given by the following expression:

$$\frac{\partial u}{\partial t} - \alpha \Delta u = 0. \tag{1.1}$$

Here u represents the temperature and α the thermal diffusivity. In this case the unknowns (the temperature) are located at the vertices of the mesh as depicted in Figure 1.3. In order to

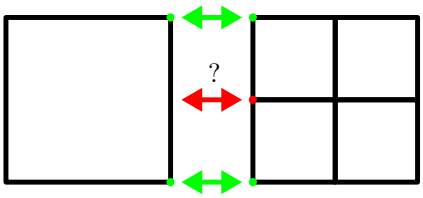


Figure 1.3 – Solving the heat equation on a non-conforming mesh.

make the connection, it lacks one additional node on the left. In this configuration there is no straightforward way to communicate the temperature from right to left. This disagreement must be resolved.

As similar situation arises in problems where fluxes are involved. Take for example the energy equation:

$$\nabla \cdot \mathbf{q} + \frac{\partial u}{\partial t} = 0 \tag{1.2}$$

Here u represents the local energy density and $\mathbf{q}$ the energy flux vector. The energy flux represents the transfer of energy per unit cross-sectional area per unit of time. Once again, the question

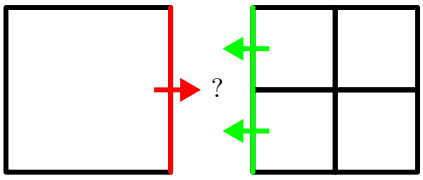


Figure 1.4 – Solving the energy equation on a non-conforming mesh.

arises how to connect these fluxes. It seems imperative that a balance must exist on the shared boundary: the sum of fluxes on the right must equal the single flux on the left. If that is not the case, energy would not be conserved across the boundary and non-physical solutions would result containing anomalies such as reflections. It does not always have to be energy which is not conserved, it could just as well be mass. In either case, the connection is of prime importance.

A possible solution to this problem is provided by the so-called MEM, the method of choice in this thesis. As the name implies, this method *mortars* the mesh elements together by introducing the mortar element. Figure 1.5 contains a depiction of this approach. Information from either

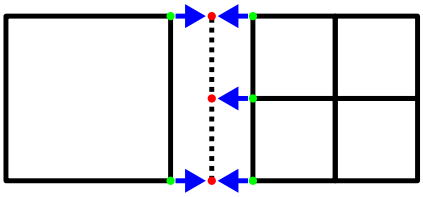


Figure 1.5 – The mortar element method.

side is now projected onto the mortar element where a mortar solution is calculated. This mortar solution is expressed in terms of the unknowns it connects and can be solved for. In the end, the complete solution covers the whole domain but above all, it is continuous at the boundary.

Up till now it has not been clarified how the right half is refined. In principle there are two options in a SEM: 1) it consists of four separate elements or 2) it is a second order element. This corresponds to what is known as h- and p-refinement respectively, or simply hp-refinement when both approaches are combined.

Starting with h-refinement, it reduces element size by splitting them up in smaller elements. The reverse is also possible, combining elements to coarsen the mesh. The need for h-refinement arises whenever the solution exhibits non-smooth behavior. Evidently, introducing more and smaller elements will allow the solution to be captured more accurately.

In the case of p-refinement not the element size but the element order is varied. Hence, it allows the order of elements to be locally increased for solutions which are smooth but exhibit large variations for example.

Not surprisingly, combining both techniques yields an hp-refinement method. It is a flexible method that increases accuracy where needed while at the same time is able to speed up computations significantly. Although not part of this thesis, the next logical step would be an hp-

adaptive method. Adaptive methods combine a refinement strategy with an adaptive algorithm. An adaptive algorithm consists of 1) a stopping criterion or tolerance level and 2) a modification strategy (Eriksson et. al. [9]). Using error estimators, it keeps track of the (local) solution error. At each iteration it checks whether or not the stopping criterion has been met. As long as this is not the case, a refinement approach is suggested (which in this case consists of either h- or p-refinement) to reduce the error. Elements in which the solution error is large are then flagged for refinement and, based on gradient or curvature information of the solution, the next course of action is decided: split the element or increase the polynomial order. This process is repeated at every iteration until the (local) error everywhere has been reduced below a specified value. In fact, an hp-adaptive method was used to solve the problem depicted in Figure 1.1.

Although this thesis focuses on mesh refinement, two more refinement strategies can be distinguished. One of these strategies is to simply adjust the initial mesh using a spring analogy. The accuracy of this approach is limited as the number of degrees of freedom is fixed from the start. The other approach is complete regeneration of the mesh. In the end, it requires fewer degrees of freedom than a straightforward refinement method and convergence is faster. However, regenerating the complete mesh at every iteration is an elaborate and complex task (Nithiarasu and Zienkiewicz [26]).

1.2 Mimetic methods

As mentioned, in CFD discretization techniques come in many flavors, the main ones being finite element, finite volume and finite difference. Although these techniques work in practice, they leave something to be desired. In general, properties such as accuracy, stability and consistency are important. However, these do not necessarily say anything about how well it captures the underlying physics. For example, losing either mass or energy almost always is the result of not fully adhering to the physics.

In many cases this can be attributed to a loss of information in the translation from the continuous PDE to its discrete equivalent. In order to yield physically sound results, new approaches are needed that retain that information. Mimetic methods aim to achieve this by accounting for the spatial and temporal geometric object each physical quantity is associated with. The goal of mimetic methods is in fact to mimic the continuous behavior of differential equations at the discrete level. In a sense it tries to preserve as much as possible of the continuous structure.

To give an example, consider the quantities density, temperature and velocity. These are often evaluated in points. This is not however always the most logical approach. Of these three quantities only temperature represents a physical variable that is naturally associated with a point. Density on the other hand is actually related to a volume while velocity can be more suitably represented along a line. To promote physical fidelity, it becomes necessary to take into account the relation between the physical variable and its associated geometry. This requires two new tools, differential geometry and algebraic topology.

Within differential geometry physical quantities are described using so-called differential forms. As will become clear in this thesis, these differential forms incorporate the structure or geometry a certain quantity is associated with. Unlike vector calculus, where density, temperature and velocity would for example be denoted as ρ, T and v, their notation in differential geometry is different, reflecting their geometrical association. For example, the temperature, associated with a point, would be denoted as $T^{(0)}$. The velocity, which resides on a line, would be denoted as $v^{(1)}$. And similarly, the density which is associated with a volume would be denoted as $\rho^{(3)}$. A graphical representation of these quantities can be found in Figure 1.6. While differential geometry acts on the continuous level, the representation at the discrete level is done

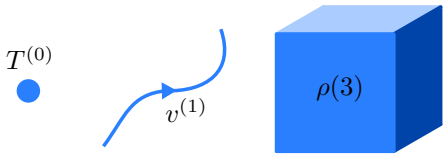


Figure 1.6 – Physical quantities and their associated geometrical objects.

with the help of algebraic topology. Like its name implies, it is solely concerned with topological structures. Its main building blocks are simply called p-cells, which are nothing more than p-dimensional geometrical objects, not much different from the objects depicted in Figure 1.6. In turn, a collection of these p-cells yields what are called chains and cochains. These and the fact that many of the properties found in differential geometry have an equivalent in algebraic topology makes it an ideal combination. The resulting method is a mimetic one in which new operators, such as the reduction and reconstruction, provide a way to translate PDE's in differential geometry to their discrete equivalent in algebraic topology and back again.

1.3 Motivation

Although the short introduction here may suggest otherwise, mimetic methods are gaining more attention. Steadily advanced fluid flow problems come within reach, requiring the type of refinement presented here. In the first place to more accurately resolve particular flow features but at the same time keeping an eye on the computational requirements as well. As hp-refinement provides this flexibility, it is considered a useful addition to the MSEM.

The goal of this thesis is therefore to combine the MSEM with hp-refinement while keeping the desirable properties of a mimetic method as well as the efficiency and flexibility of mesh refinement. The resulting method will of course be tested and verified using a number of different test cases.

1.4 Literature

So far a lot has been mentioned about mesh refinement and mimetic methods. These concepts were not just invented here, some of them have been around for a while. This thesis is therefore based on the works by people working in the field of CFD. Consequently, this section is devoted to existing literature in both fields.

In the context of spectral elements, the book by Canuto et. al. [7] must first be mentioned. Looking at mesh refinement in particular, hp-refinement goes back a long way. For properties on h-, p- and hp versions of the finite element method the interested reader is referred to Gui and Babuška [13, 14, 15]. An extensive overview of spectral elements as well as hp-refinement is provided in the book by Karniadakis and Sherwin [18]. This includes a chapter on non-conforming discretizations, describing the MEM. The origin of the MEM can be found in the papers by Maday et. al. and Anagnostou et. al. [23, 1] (derived from the works by Anagnostou [29] and Mavriplis [25]).

The relation between differential geometry and algebraic topology has already been established early on by Tonti [32]. Many of these concepts also return in the more recent works by

Mattiussi [24], Bossavit [4, 5] and Desbrun et. al. [8]. The extensive theory of differential geometry is explained in the books by Frankel [11] and Flanders [10]. The principles of mimetic discretizations are explained in various works such as the ones by Bochev and Hyman [3] and Kreeft et. al. [22]. In particular, applications of the mimetic spectral element methods are found in Kreeft et. al. [21], Palha and Gerritsma [28], Kreeft and Gerritsma [19, 20].

Finally, related work by MSc. students at the Delft University of Technology focused on mimetic methods include Bouman [6], Oud [27], Hiemstra [16] and Rebelo [30].

1.5 Outline

The goal of this thesis is to implement, test en verify *hp*-refinement within the mimetic framework. Based on the theory of mortar elements, an attempt is made at formulating its mimetic equivalent. Analysis will be limited to the scalar Laplacian. Therefore the Poisson problem is the differential equation of choice here. It will be solved for both 0-forms and 2-forms, which both require a different approach.

This thesis will be structured as follows. First of all, in order to familiarize the reader with the mimetic framework the next chapter (Chapter 2) starts of with a proper introduction to differential geometry. It will cover all the necessary ingredients that are needed to read this thesis. Chapter 3 will then introduce the discrete discrete world of algebraic topology. Having introduced these new fields, Chapter 4 combines the two into a mimetic approach.

Once all the theory has been covered, Chapter 5 starts of by solving a simple 2D Poisson problem for 0- and 2-forms in order to familiarize the reader with mechanics of a MSEM. Having done that, the report then moves on to Chapter 6 to discuss the theory of the mortar element approach within a mimetic framework, the focus of this thesis. Chapter 7 then follows by demonstrating the new implementation using a number of different test cases. Finally, conclusions and recommendations are given in Chapter 8.

This report also contains a short appendix, Appendix A and Appendix B. The first appendix contains some of the long derivations encountered when solving the 2D Laplace problem in Chapter 5. The second one solely contains tables. These tables belong to the test cases presented in Chapter 7.

CHAPTER 2

DIFFERENTIAL GEOMETRY

As previously mentioned, the mimetic framework relies heavily on the fields of differential geometry and algebraic topology. This chapter serves as an introduction to the former. As will be shown, differential geometry will prove to be instrumental in obtaining a continuous formulation for any PDE. Although differential geometry is a vast and interesting field in itself, this introduction will be limited to the concepts applicable to this thesis only. To that end, this chapter starts by introducing the main building blocks in differential geometry, the differential forms (Section 2.1). This will be followed by an explanation of the various operators that can act on them. The first two are the wedge product (Section 2.2) and the exterior derivative (Section 2.3), followed by the theorem of the the generalized Stokes problem (Section 2.4). Next are the definition of the Hodge star operator as well as the inner product (Section 2.5), ending the chapter with a short explanation of the pull-back (Section 2.6). More details on differential geometry can be found in the works by Flanders [10] and Frankel [11], as well as Kreeft et al. [22] whose description is already tailored towards a mimetic framework.

2.1 Differential forms

Differential forms or k-forms represent (physical) quantities associated with k-dimensional geometric objects. The space of k-forms is denoted by $\Lambda^k(\Omega)$ with Ω representing an n-dimensional domain in $\mathbb{R}^n$. Starting with the formal definition, each k-form $\in \Lambda^k(\Omega)$ can be written as:

$$\alpha^{(k)} = \sum_I f_I(\boldsymbol{x}) \mathrm{d}x^I. \tag{2.1}$$

In this definition f is a smooth function which is infinitely differentiable (as many times as needed). The term $\mathrm{d}x^I$ denotes the so-called basis vectors:

$$\mathrm{d}x^I = \mathrm{d}x^{i_1} \wedge \mathrm{d}x^{i_2} \wedge \cdots \wedge \mathrm{d}x^{i_k}. \tag{2.2}$$

Hence, the index I takes on the values $1 \leq i_1 < \cdots < i_k \leq n$. What follows next are a number of examples of the various k-forms in both $\mathbb{R}^2$ and $\mathbb{R}^3$ (with coordinates x, y and z) which will also clarify the presence of the basis vectors.

0-forms: A 0-form is simply a scalar valued function which is naturally integrated over a 0-dimensional geometrical object or point (as opposed to a line, surface or volume). This is in fact not much different from scalar values or scalar valued functions seen in classical vector calculus. In $\mathbb{R}^2$ and $\mathbb{R}^3$ the 0-form $\alpha^{(0)}$ looks as follows:

$$\alpha^{(0)} = a(x,y) \quad \in \Lambda^0(\mathbb{R}^2), \tag{2.3}$$
$$\alpha^{(0)} = a(x,y,z) \quad \in \Lambda^0(\mathbb{R}^3). \tag{2.4}$$

Due to the *point*-wise nature of its measurement, an example of a physical quantity that can be represented as a 0-form is temperature.

1-forms: Unlike the 0-form, the 1-form is associated with lines instead of points. In classical vector calculus its proxy is therefore considered to be a vector. With 1-forms the basis vectors are also starting to play a role, which in this case are $\mathrm{d}x$, $\mathrm{d}y$ and $\mathrm{d}z$. Examples of a 1-form $\alpha^{(1)}$ in $\mathbb{R}^2$ and $\mathbb{R}^3$ are given below:

$$\alpha^{(1)} = a(x,y)\mathrm{d}x + b(x,y)\mathrm{d}y \quad \in \Lambda^1(\mathbb{R}^2), \tag{2.5}$$
$$\alpha^{(1)} = a(x,y,z)\mathrm{d}x + b(x,y,z)\mathrm{d}y + c(x,y,z)\mathrm{d}z \quad \in \Lambda^1(\mathbb{R}^3). \tag{2.6}$$

An example of a physical quantity that can be represented as a 1-form is the velocity. By definition is it associated with the distance traveled along a *line* in a given time interval.

2-forms: Next is the 2-form, which is associated with surfaces. In classical vector calculus this is also considered a vector. This reveals an important difference with differential geometry where a 2-form is distinctly different from a 1-form. This is mainly caused by the basis vectors. The example below of a 2-form $\alpha^{(2)}$ in both $\mathbb{R}^2$ and $\mathbb{R}^3$ should clarify this:

$$\alpha^{(2)} = a(x,y)\mathrm{d}x \wedge \mathrm{d}y \quad \in \Lambda^2(\mathbb{R}^2), \tag{2.7}$$
$$\alpha^{(2)} = a(x,y,z)\mathrm{d}y \wedge \mathrm{d}z + b(x,y,z)\mathrm{d}z \wedge \mathrm{d}x + c(x,y,z)\mathrm{d}x \wedge \mathrm{d}y \quad \in \Lambda^2(\mathbb{R}^3). \tag{2.8}$$

An example of a physical quantity that can be represented as a 2-form is vorticity which is defined as the circulation per unit *area*.

3-forms: Finally, a 3-form is associated with volumes. In classical vector calculus its proxy is again simply a scalar value or scalar valued function. In $\mathbb{R}^3$ the 3-form $\alpha^{(3)}$ looks as follows:

$$\alpha^{(3)} = a(x,y,z)\mathrm{d}x \wedge \mathrm{d}y \wedge \mathrm{d}z \quad \in \Lambda^3(\mathbb{R}^3). \tag{2.9}$$

An example of a physical quantity that can be represented as a 3-form is the density which is defined as the mass per unit *volume*.

With the presence of the basis vectors comes the concept of duality pairing. As k-forms are associated with a particular geometrical object, an integral can be defined along with it. For example, a 1-form $\in \Lambda^1(\mathbb{R}^3)$ can be integrated along curve $\mathcal{M}^1$ as follows:

$$\int_{\mathcal{M}^1} \alpha^{(1)} = \int_{\mathcal{M}^1} A(x,y,z)\mathrm{d}x + B(x,y,z)\mathrm{d}y + C(x,y,z)\mathrm{d}z.$$

Which implies the following duality pairing:

$$\langle \alpha^{(1)}, \mathcal{M}^1 \rangle = \int_{\mathcal{M}^1} \alpha^{(1)} \quad \in \mathrm{R}. \tag{2.10}$$

Table 2.1 – Duality pairing of k-forms.

k-form	duality paring		
0-form	$\langle \alpha^{(0)}, \mathcal{M}^0 \rangle$	$=$	$\alpha^{(0)}(\mathcal{M}^1)$
1-form	$\langle \alpha^{(1)}, \mathcal{M}^1 \rangle$	$=$	$\int_{\mathcal{M}^1} \alpha^{(1)}$
2-form	$\langle \alpha^{(2)}, \mathcal{M}^2 \rangle$	$=$	$\int_{\mathcal{M}^2} \alpha^{(2)}$
3-form	$\langle \alpha^{(3)}, \mathcal{M}^3 \rangle$	$=$	$\int_{\mathcal{M}^3} \alpha^{(3)}$

Similarly, for all k-forms up to $k = 3$ the duality pairing is summarized in Table 2.1. Differential forms obey the conventional rules of addition and subtraction for forms of equal degree as well as multiplication by a function.

Up till now the appearance of the wedge symbol $\wedge$ in the various k-forms has not been explained. The next section will clarify this new operator.

2.2 Wedge product

The first operator to be introduced is the wedge product which allows multiplication of differential forms:

$$\wedge : \Lambda^k(\Omega) \times \Lambda^l(\Omega) \mapsto \Lambda^{k+l}(\Omega). \tag{2.11}$$

Hence, the wedge product is the product between a k-form and an l-form (with their respective spaces being denoted by $\Lambda^k(\Omega)$ and $\Lambda^l(\Omega)$) and results in a $(k+l)$-form. Note that $k + l \leq n$, with n the dimension of the vector space. The basic properties of the wedge product can be summarized as follows. Given the k-forms $\alpha, \beta \in \Lambda^k(\Omega)$, $\gamma, \delta \in \Lambda^l(\Omega)$ as well as the constants a, b, c, $d \in \mathbb{R}^3$, then the wedge product is:

1. distributive: $(a\alpha + b\beta) \wedge (c\gamma + d\delta) = ac(\alpha \wedge \gamma) + ad(\alpha \wedge \delta) + bc(\beta \wedge \gamma) + bd(\beta \wedge \delta)$

2. associative: $(\alpha \wedge \beta) \wedge \gamma = \alpha \wedge (\beta \wedge \gamma) = \alpha \wedge \beta \wedge \gamma$

3. anticommutative: $\alpha \wedge \beta = (-1)^{kl} \beta \wedge \alpha$

Additionally, for *odd* degree forms the following holds:

$$\alpha^{(2k+1)} \wedge \alpha^{(2k+1)} = 0. \tag{2.12}$$

Finally, the wedge product between two 1-forms yields an expression similar to the cross product in traditional vector calculus. Consider the following two 1-forms in $\mathbb{R}^3$:

$$\alpha^{(1)} = a_1 \mathrm{d}x + b_1 \mathrm{d}y + c_1 \mathrm{d}z,$$
$$\beta^{(1)} = a_2 \mathrm{d}x + b_2 \mathrm{d}y + c_2 \mathrm{d}z.$$

Application of the wedge product results in:

$$\begin{aligned}\alpha^{(1)} \wedge \beta^{(1)} =&(a_1\mathrm{d}x + b_1\mathrm{d}y + c_1\mathrm{d}z) \wedge (a_2\mathrm{d}x + b_2\mathrm{d}y + c_2\mathrm{d}z)\\ =&(a_1a_2)\underbrace{\mathrm{d}x \wedge \mathrm{d}x}_{=0} + (a_1b_2)\mathrm{d}x \wedge \mathrm{d}y + (a_1c_2)\mathrm{d}x \wedge \mathrm{d}z +\\ &(b_1a_2)\underbrace{\mathrm{d}y \wedge \mathrm{d}x}_{=-\mathrm{d}x\wedge\mathrm{d}y} + (b_1b_2)\underbrace{\mathrm{d}y \wedge \mathrm{d}y}_{=0} + (b_1c_2)\mathrm{d}y \wedge \mathrm{d}z +\\ &(c_1a_2)\underbrace{\mathrm{d}z \wedge \mathrm{d}x}_{=-\mathrm{d}x\wedge\mathrm{d}z} + (c_1b_2)\underbrace{\mathrm{d}z \wedge \mathrm{d}y}_{=-\mathrm{d}y\wedge\mathrm{d}z} + (c_1c_2)\underbrace{\mathrm{d}z \wedge \mathrm{d}z}_{=0}.\end{aligned}$$

Hence, equation (2.12) and the anticommutative property cause this expression to collapse to:

$$\alpha^{(1)} \wedge \beta^{(1)} = (b_1c_2 - c_1b_2)\mathrm{d}y \wedge \mathrm{d}z + (c_1a_2 - a_1c_2)\mathrm{d}z \wedge \mathrm{d}x + (a_1b_2 - b_1a_2)\mathrm{d}x \wedge \mathrm{d}y.$$

This result is similar to the cross product between vectors in classical vector calculus.

As a final note, the wedge symbol is often not written down explicitly to simplify notation. For example, the 2-form of equation (2.8) could also be written as:

$$\alpha^{(2)} = a(x, y, z)\mathrm{d}y\mathrm{d}z + b(x, y, z)\mathrm{d}z\mathrm{d}x + c(x, y, z)\mathrm{d}x\mathrm{d}y. \tag{2.13}$$

This is still the exact same 2-form as given by equation (2.8).

2.3 Exterior derivative

The second operator is the exterior derivative d which is responsible for differentiation. When acting on a k-form it transforms it to a $(k+1)$-form:

$$\mathrm{d} : \Lambda^k(\Omega) \mapsto \Lambda^{k+1}(\Omega). \tag{2.14}$$

Here $k = 0, 1, \ldots, n-1$, with n again equal to the dimension of the vector space. Important properties of the exterior derivative are listed below. Given the k-forms $\alpha \in \Lambda^k(\Omega)$ and $\beta \in \Lambda^l(\Omega)$ then:

1. d is additive, $\mathrm{d}(\alpha + \beta) = \mathrm{d}\alpha + \mathrm{d}\beta$ (provided $k = l$)
2. $\mathrm{d}\alpha^{(0)}$ is the usual differential of the function $\alpha^{(0)}$
3. $\mathrm{d}(\alpha \wedge \beta) = \mathrm{d}\alpha \wedge \mathrm{d}\beta + (-1)^k\alpha \wedge \mathrm{d}\beta$
4. $\mathrm{d}^2\alpha := \mathrm{d}(\mathrm{d}\alpha) = 0$

Examples of applying the exterior derivative to k-forms in both $\mathbb{R}^3$ and $\mathbb{R}^2$ (with coordinates x, y and z) are given below.

0-forms: The exterior derivative applied to the 0-forms of equations (2.4) and (2.3) yields the following 1-forms:

$$\mathrm{d}\alpha^{(0)} = \frac{\partial a}{\partial x}\mathrm{d}x + \frac{\partial a}{\partial y}\mathrm{d}y \qquad \in \Lambda^1(\mathbb{R}^2), \tag{2.15}$$

$$\mathrm{d}\alpha^{(0)} = \frac{\partial a}{\partial x}\mathrm{d}x + \frac{\partial a}{\partial y}\mathrm{d}y + \frac{\partial a}{\partial z}\mathrm{d}z \quad \in \Lambda^1(\mathbb{R}^3). \tag{2.16}$$

The vector proxy of the exterior derivative applied to a 0-form is the gradient or **grad** operator (∇).

1-forms. Similarly, the exterior derivative applied to the 1-form of equation (2.5) yields the following 2-form:

$$\begin{aligned}\mathrm{d}\alpha^{(1)} =& \frac{\partial a}{\partial x}\underbrace{\mathrm{d}x\wedge\mathrm{d}x}_{=0}+\frac{\partial a}{\partial y}\underbrace{\mathrm{d}y\wedge\mathrm{d}x}_{-\mathrm{d}x\wedge\mathrm{d}y}+\frac{\partial b}{\partial x}\mathrm{d}x\wedge\mathrm{d}y+\frac{\partial b}{\partial y}\underbrace{\mathrm{d}y\wedge\mathrm{d}y}_{=0}\\ =& \left(\frac{\partial b}{\partial x}-\frac{\partial a}{\partial y}\right)\mathrm{d}x\wedge\mathrm{d}y \quad \in\Lambda^2(\mathbb{R}^2).\end{aligned} \tag{2.17}$$

Likewise for the 1-form of equation (2.6):

$$\begin{aligned}\mathrm{d}\alpha^{(1)} =& \frac{\partial a}{\partial x}\underbrace{\mathrm{d}x\wedge\mathrm{d}x}_{=0}+\frac{\partial a}{\partial y}\underbrace{\mathrm{d}y\wedge\mathrm{d}x}_{-\mathrm{d}x\wedge\mathrm{d}y}+\frac{\partial a}{\partial z}\underbrace{\mathrm{d}z\wedge\mathrm{d}x}_{-\mathrm{d}x\wedge\mathrm{d}z}+\\ & \frac{\partial b}{\partial x}\mathrm{d}x\wedge\mathrm{d}y+\frac{\partial b}{\partial y}\underbrace{\mathrm{d}y\wedge\mathrm{d}y}_{=0}+\frac{\partial b}{\partial z}\underbrace{\mathrm{d}z\wedge\mathrm{d}y}_{-\mathrm{d}y\wedge\mathrm{d}z}+\\ & \frac{\partial c}{\partial x}\mathrm{d}x\wedge\mathrm{d}z+\frac{\partial c}{\partial y}\mathrm{d}y\wedge\mathrm{d}z+\frac{\partial c}{\partial z}\underbrace{\mathrm{d}z\wedge\mathrm{d}z}_{=0}.\end{aligned}$$

Which eventually reduces to:

$$\begin{aligned}\mathrm{d}\alpha^{(1)} =& \left(\frac{\partial c}{\partial y}-\frac{\partial b}{\partial z}\right)\mathrm{d}y\wedge\mathrm{d}z+\\ & \left(\frac{\partial c}{\partial x}-\frac{\partial a}{\partial z}\right)\mathrm{d}z\wedge\mathrm{d}x+\\ & \left(\frac{\partial b}{\partial x}-\frac{\partial a}{\partial y}\right)\mathrm{d}x\wedge\mathrm{d}y \quad \in\Lambda^2(\mathbb{R}^3).\end{aligned} \tag{2.18}$$

In vector calculus this corresponds to the **curl** operator ($\nabla\times$).

2-forms. The exterior derivative applied to the 2-form of equation (2.7) yields:

$$\mathrm{d}\alpha^{(2)} = \frac{\partial a}{\partial x}\underbrace{\mathrm{d}x\wedge(\mathrm{d}x\wedge\mathrm{d}y)}_{=0}+\frac{\partial a}{\partial y}\underbrace{\mathrm{d}y\wedge(\mathrm{d}x\wedge\mathrm{d}y)}_{=0}=0. \tag{2.19}$$

Since the maximum degree of a differential form in $\mathbb{R}^2$ is 2, this returns 0. Next the exterior derivative is applied to the 2-form of equation (2.8) which yields:

$$\begin{aligned}\mathrm{d}\alpha^{(2)} =& \frac{\partial a}{\partial x}\mathrm{d}x\wedge(\mathrm{d}y\wedge\mathrm{d}z)+\frac{\partial a}{\partial y}\underbrace{\mathrm{d}y\wedge(\mathrm{d}y\wedge\mathrm{d}z)}_{=0}+\frac{\partial a}{\partial z}\underbrace{\mathrm{d}z\wedge(\mathrm{d}y\wedge\mathrm{d}z)}_{=0}\\ & \frac{\partial b}{\partial x}\underbrace{\mathrm{d}x\wedge(\mathrm{d}z\wedge\mathrm{d}x)}_{=0}+\frac{\partial b}{\partial y}\mathrm{d}y\wedge(\mathrm{d}z\wedge\mathrm{d}x)+\frac{\partial b}{\partial z}\underbrace{\mathrm{d}z\wedge(\mathrm{d}z\wedge\mathrm{d}x)}_{=0}+\\ & \frac{\partial c}{\partial x}\underbrace{\mathrm{d}x\wedge(\mathrm{d}x\wedge\mathrm{d}y)}_{=0}+\frac{\partial c}{\partial y}\underbrace{\mathrm{d}y\wedge(\mathrm{d}x\wedge\mathrm{d}y)}_{=0}+\frac{\partial c}{\partial z}\mathrm{d}z\wedge(\mathrm{d}x\wedge\mathrm{d}y.\end{aligned}$$

Which in turn simplifies to:

$$\mathrm{d}\alpha^{(2)} = \left(\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y} + \frac{\partial c}{\partial z}\right) \mathrm{d}x \wedge \mathrm{d}y \wedge \mathrm{d}z \quad \in \Lambda^3(\mathbb{R}^3). \tag{2.20}$$

In vector calculus this corresponds to the divergence or **div** operator $(\nabla\cdot)$.

3-forms. Finally, the exterior derivative applied to the 3-form of equation (2.9) reduces to 0 as well:

$$\begin{aligned}\mathrm{d}\alpha^{(3)} =& \frac{\partial a}{\partial x} \underbrace{\mathrm{d}x \wedge (\mathrm{d}x \wedge \mathrm{d}y \wedge \mathrm{d}z)}_{=0} + \\ & \frac{\partial a}{\partial y} \underbrace{\mathrm{d}y \wedge (\mathrm{d}x \wedge \mathrm{d}y \wedge \mathrm{d}z)}_{=0} + \\ & \frac{\partial a}{\partial z} \underbrace{\mathrm{d}z \wedge (\mathrm{d}x \wedge \mathrm{d}y \wedge \mathrm{d}z)}_{=0} = 0.\end{aligned} \tag{2.21}$$

Property 4 of the exterior derivative can easily be shown by applying the exterior derivative to any of the results above. For example, applying the d once more to equation (2.16) (the result of the d acting on a 0-form in $\mathbb{R}^3$) yields:

$$\begin{aligned}\mathrm{d}\mathrm{d}\alpha^{(0)} =& \underbrace{\left(\frac{\partial^2 a}{\partial y \partial z} - \frac{\partial^2 a}{\partial z \partial y}\right)}_{=0} \mathrm{d}y \wedge \mathrm{d}z + \\ & \underbrace{\left(\frac{\partial^2 a}{\partial x \partial z} - \frac{\partial^2 a}{\partial z \partial x}\right)}_{=0} \mathrm{d}z \wedge \mathrm{d}x + \\ & \underbrace{\left(\frac{\partial^2 a}{\partial x \partial y} - \frac{\partial^2 a}{\partial y \partial x}\right)}_{=0} \mathrm{d}x \wedge \mathrm{d}y = 0.\end{aligned}$$

This particular result corresponds to the vector calculus identity:

$$\nabla \times (\nabla f) \equiv 0 \quad \text{or} \quad \mathbf{curl} \cdot \mathbf{grad} \equiv 0. \tag{2.22}$$

Similarly, applying the d once more to the 2-form of equation (2.18) (the result of the d acting on a 1-form $\mathbb{R}^3$) yields:

$$\begin{aligned}\mathrm{d}\mathrm{d}\alpha^{(1)} =& \underbrace{\left(\frac{\partial^2 a}{\partial y \partial z} - \frac{\partial^2 a}{\partial z \partial y}\right)}_{=0} \mathrm{d}x\mathrm{d}y\mathrm{d}z + \\ & \underbrace{\left(\frac{\partial^2 b}{\partial z \partial x} - \frac{\partial^2 b}{\partial x \partial z}\right)}_{=0} \mathrm{d}x\mathrm{d}y\mathrm{d}z + \\ & \underbrace{\left(\frac{\partial^2 c}{\partial x \partial y} - \frac{\partial^2 c}{\partial y \partial x}\right)}_{=0} .\mathrm{d}x\mathrm{d}y\mathrm{d}z = 0\end{aligned}$$

This corresponds to the vector calculus identity:

$$\nabla \cdot (\nabla \times \mathbf{u}) \equiv 0 \quad \text{or} \quad \mathbf{div} \cdot \mathbf{curl} \equiv 0. \tag{2.23}$$

In both cases, the fact that the individual terms evaluate to 0 is reflected in another property from vector calculus:

$$\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right). \tag{2.24}$$

This is the rule for mixed partial derivatives.

Previous results can be combined into a so-called De Rham complex which in two dimensions is written as:

$$\mathbb{R} \longrightarrow \Lambda^0(\Omega) \xrightarrow[\mathbf{grad}]{\mathrm{d}} \Lambda^1(\Omega) \xrightarrow[\mathbf{curl}]{\mathrm{d}} \Lambda^2(\Omega) \xrightarrow{\mathrm{d}} 0 \ . \tag{2.25}$$

Or:

$$\mathbb{R} \longrightarrow \Lambda^0(\Omega) \xrightarrow[\mathbf{rot}]{\mathrm{d}} \Lambda^1(\Omega) \xrightarrow[\mathbf{div}]{\mathrm{d}} \Lambda^2(\Omega) \xrightarrow{\mathrm{d}} 0 \ . \tag{2.26}$$

And in three dimensions:

$$\mathbb{R} \longrightarrow \Lambda^0(\Omega) \xrightarrow[\mathbf{grad}]{\mathrm{d}} \Lambda^1(\Omega) \xrightarrow[\mathbf{curl}]{\mathrm{d}} \Lambda^2(\Omega) \xrightarrow[\mathbf{div}]{\mathrm{d}} \Lambda^3(\Omega) \xrightarrow{\mathrm{d}} 0 \ . \tag{2.27}$$

The fact that in two dimensions not just one, but two De Rham complexes are constructed, anticipates the introduction of the concept of orientation. Details will follow in Section 2.5 which introduces the Hodge star operator.

As a final note, the De Rham complex can be generalized to higher dimensional spaces Λ^k which has implications for the theorem to be introduced next.

2.4 The generalized Stokes' theorem

In the previous section correspondence has been shown between the exterior derivative and the **grad**, **curl** and **div** operators from classical vector calculus. In differential geometry these integral theorems can be combined into the generalized Stokes' theorem.

Consider a k-form $\alpha^{(k)}$ and a $(k+1)$-dimensional geometrical object or domain Ω^{k+1} as well as its boundary $\partial\Omega^{k+1}$. Then the generalized Stokes' theorem states:

$$\int_{\partial\Omega^{k+1}} \alpha^{(k)} = \int_{\Omega^{k+1}} \mathrm{d}\alpha^{(k)}. \tag{2.28}$$

Which summarizes integration in differential geometry. In fact, it is the duality pairing from Table 2.1 in Section 2.1. Hence, this theorem can therefore also be written as:

$$\langle \alpha^k, \partial\Omega^{k+1} \rangle = \langle \mathrm{d}\alpha^k, \Omega^{k+1} \rangle. \tag{2.29}$$

The fact that the exterior derivative d is the formal adjoint of the boundary operator ∂ will prove to have ramifications later on. Furthermore, this theorem generalizes the Gradient ($k = 0$),

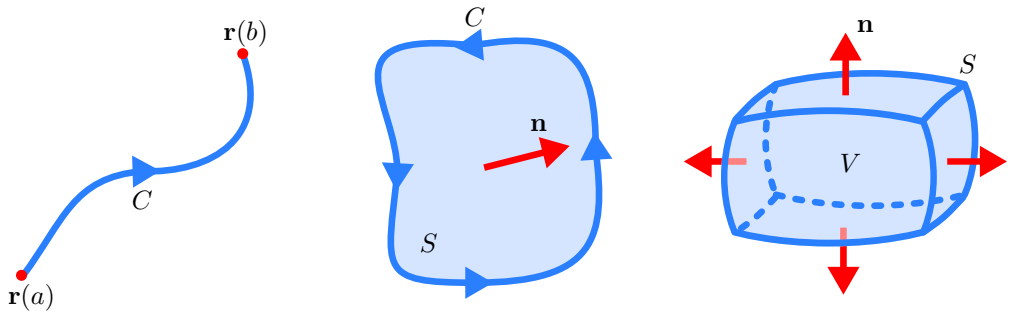
(Kelvin-)Stokes' ($k = 1$) and Divergence ($k = 2$) theorems known from vector calculus:

$$k = 0: \quad \int_a^b (\nabla u)\mathrm{d}s \quad = f(b) - f(a), \tag{2.30}$$

$$k = 1: \quad \int_S (\nabla \times \mathbf{u})\mathrm{d}S = \int_{\partial S} \mathbf{u} \cdot \mathbf{n}\,\mathrm{d}s, \tag{2.31}$$

$$k = 2: \quad \int_V (\nabla \cdot \mathbf{u})\mathrm{d}V = \int_{\partial V} \mathbf{v} \cdot \mathbf{n}\,\mathrm{d}S. \tag{2.32}$$

Hence, the **grad** maps points to lines, the **curl** maps lines to surfaces and the **div** maps surfaces to volumes. A geometrical representation of these individual theorems is visualized in Figure 2.1.



(a) $\int_C \nabla f \cdot \mathrm{d}\mathbf{r} = f(\mathbf{r}(b) - f(\mathbf{r}(a))$ **(b)** $\iint_S \nabla \times \mathbf{F} \cdot \mathrm{d}\mathbf{S} = \int_C \mathbf{F} \cdot \mathrm{d}\mathbf{r}$ **(c)** $\iiint_V \nabla \mathbf{F}\ \mathrm{d}V = \iint_S \mathbf{F} \cdot \mathrm{d}\mathbf{S}$

Figure 2.1 – A geometric representation of the generalized Stokes' theorem.

2.5 Hodge star

Before discussing the next operator the concept of orientation must be introduced. For an extensive background on orientation, see the works by Tonti [32], Mattiussi [24] and Bossavit [4, 5]. Within these works it is explained that every geometrical object, whether a point, line, surface or volume, can be endowed with two types of orientation: an inner and outer one.

Starting with the inner orientation, a visual representation is given in Figure 2.2. In this

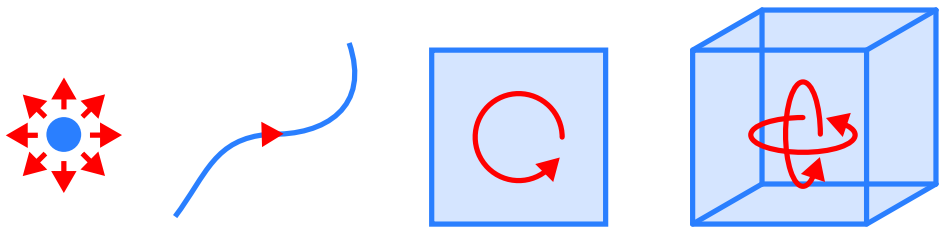


Figure 2.2 – Inner oriented geometrical objects in $\mathbb{R}^3$.

particular example the orientation of a point is defined as a source, hence the arrows are drawn in an outward pointing direction. Evidently, it could also be defined as a sink which would require the arrows pointing inwards. For a line the orientation is defined as a direction along the line itself. This can either be in the direction shown here or opposite to it. For the surface the orientation is defined as a clockwise or counterclockwise rotation. Finally, the orientation of a volume is simply an inward or outward pointing symbol.

In addition to the inner orientation, each geometrical object can also be endowed with an outer orientation as depicted in Figure 2.3. The orientation of an outer oriented point is similar

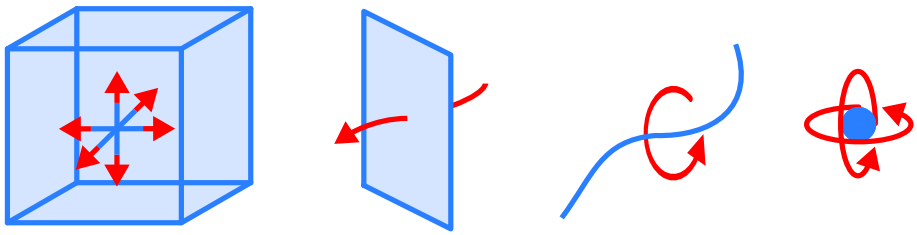


Figure 2.3 – Outer oriented geometrical objects in $\mathbb{R}^3$.

to that of an inner oriented volume, i.e. an inward or outward symbol. The outer orientation of a line is the rotation around it, which can either be defined clockwise or counterclockwise. Furthermore, the outer orientation of a surface is simply a line through it. Finally, the outer oriented volume is defined similar to the inner oriented point, i.e. it can either be a source or a sink.

In two dimensions, the orientation of inner oriented geometric objects is exactly the same as in three dimensions. However, this is not true for the outer orientation. The different outer orientations in two dimensions are given in Figure 2.4. The inverted ordering of the outer

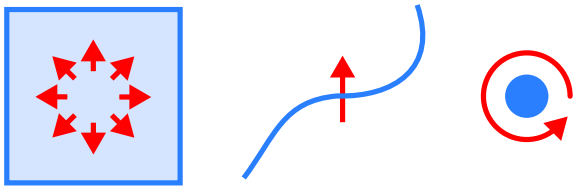


Figure 2.4 – Outer oriented geometrical objects in $\mathbb{R}^2$.

oriented geometrical objects in Figures 2.3 and 2.4 is done to emphasize the correspondence in the type of orientation with the inner oriented geometrical objects. In fact, the notion of orientation is intimately connected with the operator to be introduced next which establishes a relation between inner and outer oriented geometrical objects. To that end, consider the Poisson problem $\Delta u = \nabla \cdot (\nabla u) = f$ in $\mathbb{R}^3$ represented as two first-order equations:

$$\nabla u = \mathbf{q} \quad (\text{or} \quad \mathbf{grad}\ u = \mathbf{q}), \tag{2.33}$$

$$\nabla \cdot \mathbf{q} = f \quad (\text{or} \quad \mathbf{div}\ q = f). \tag{2.34}$$

Previously it has been shown that the **grad** operator acts on 0-forms (which yields a 1-form) while the **div** operator acts on 2-forms (which yields a 3-form). Hence, in differential geometry these first order equations become:

$$\mathrm{d}u^{(0)} = q^{(1)}, \tag{2.35}$$

$$\mathrm{d}q^{(2)} = f^{(3)}. \tag{2.36}$$

Although in the vector calculus formulation of the Poisson problem there is no discernible difference between $\mathbf{q}$ appearing in equation (2.33) and $\mathbf{q}$ appearing in equation (2.34), this is not true for its equivalent in differential geometry in which it appears as both a 1-form and a 2-form. This inconsistency prevents a solution to the Poisson problem.

Now enter orientation. Since k-forms are associated with geometrical objects they inherit the orientation these objects are endowed with, whether that is inner or outer oriented. This choice depends on the physical problem being modeled, specifically how its main variables are most suitably represented. For example, velocity is most naturally associated with an inner oriented line (quantities that move along a line) while a flux has more in common with an outer oriented surface (quantities moving through a surface).

In this particular example, the quantity $u^{(0)}$ is chosen to be represented by an inner oriented point. Application of the exterior derivative then yields an inner oriented 1-form $q^{(1)}$. The orientations of $q^{(2)}$ and $f^{(3)}$ are unclear at this point as a step appears to be missing from $q^{(1)}$ to $q^{(2)}$.

To solve this problem requires the introduction of a new operator, more specifically the Hodge star operator which is denoted as $\star$. By definition it maps k-forms to $(n-k)$-forms:

$$\star : \Lambda^k(\Omega) \mapsto \Lambda^{n-k}(\Omega). \tag{2.37}$$

It is defined as:

$$\alpha^{(k)} \wedge \star\beta^k = (\alpha^{(k)}, \beta^{(k)})\omega^n. \tag{2.38}$$

Here ω^n is the unit volume form, i.e. $\star\omega^{(n)} = 1$. It also provides an inner product for k-forms:

$$(\alpha^{(k)}, \beta^{(k)})_\Omega := \int_\Omega \alpha^{(k)} \wedge \star\beta^{(k)}. \tag{2.39}$$

Additionally, it satisfies the property:

$$\star(\star\alpha^{(k)}) = (-1)^{k(n-k)}\alpha^{(k)}. \tag{2.40}$$

The Hodge star operator connects the De Rham complex from equation (2.27) with one oriented in opposite direction which in $\mathbb{R}^3$ looks as follows:

$$\begin{array}{ccccccccccc} \mathbb{R} & \longrightarrow & \Lambda^0(\Omega) & \xrightarrow[\textbf{grad}]{\mathrm{d}} & \Lambda^1(\Omega) & \xrightarrow[\textbf{curl}]{\mathrm{d}} & \Lambda^2(\Omega) & \xrightarrow[\textbf{div}]{\mathrm{d}} & \Lambda^3(\Omega) & \xrightarrow{\mathrm{d}} & 0 \\ & & \star\updownarrow & & \star\updownarrow & & \star\updownarrow & & \star\updownarrow & & \\ 0 & \xleftarrow{\mathrm{d}} & \Lambda^3(\Omega) & \xleftarrow[\textbf{div}]{\mathrm{d}} & \Lambda^2(\Omega) & \xleftarrow[\textbf{curl}]{\mathrm{d}} & \Lambda^1(\Omega) & \xleftarrow[\textbf{grad}]{\mathrm{d}} & \Lambda^0(\Omega) & \longleftarrow & \mathbb{R} \end{array}. \tag{2.41}$$

A graphical representation is given in Figure 2.5. The Hodge star operator changes not only the geometric object, but also the orientation. Some examples of its application in $\mathbb{R}^2$ (with the basis vectors $\mathrm{d}x$ and $\mathrm{d}y$):

$$\star\, 1 = \mathrm{d}x \wedge \mathrm{d}y, \tag{2.42}$$

$$\star\, \mathrm{d}x = \mathrm{d}y,\ \star\mathrm{d}y = -\mathrm{d}x, \tag{2.43}$$

$$\star\, (\mathrm{d}x \wedge \mathrm{d}y) = 1. \tag{2.44}$$

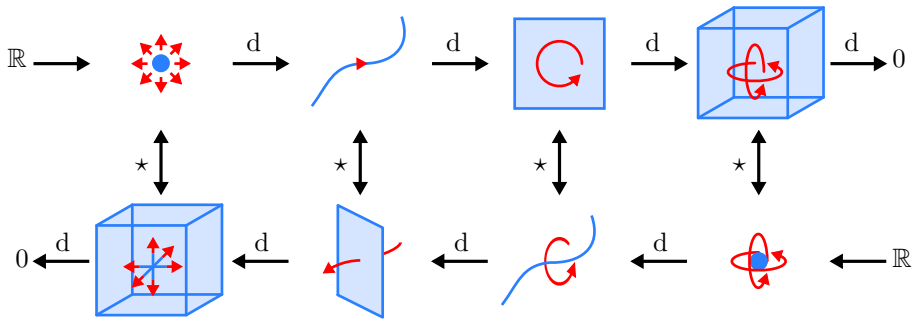


Figure 2.5 – Visual representation of the De Rham complex in $\mathbb{R}^3$.

Similarly application of the Hodge star operator in $\mathbb{R}^3$ (with the basis vectors $\mathrm{d}x$, $\mathrm{d}y$ and $\mathrm{d}z$) yields:

$$\star\, 1 = \mathrm{d}x \wedge \mathrm{d}y \wedge \mathrm{d}z, \tag{2.45}$$
$$\star\, (\mathrm{d}x \wedge \mathrm{d}y \wedge \mathrm{d}z) = 1, \tag{2.46}$$
$$\star\, \mathrm{d}x = \mathrm{d}y \wedge \mathrm{d}z,\ \star \mathrm{d}y = \mathrm{d}z \wedge \mathrm{d}x,\ \star \mathrm{d}z = \mathrm{d}x \wedge \mathrm{d}y, \tag{2.47}$$
$$\star\, (\mathrm{d}y \wedge \mathrm{d}z) = \mathrm{d}x,\ \star(\mathrm{d}z \wedge \mathrm{d}x) = \mathrm{d}y,\ \star(\mathrm{d}x \wedge \mathrm{d}y) = \mathrm{d}z. \tag{2.48}$$

These equations correspond to the vertical connections in Figure 2.5. In $\mathbb{R}^2$ the dual De Rham complex reduces to:

$$\begin{array}{ccccccc} \mathbb{R} & \longrightarrow & \Lambda^0(\Omega) & \xrightarrow[\textbf{grad}]{\mathrm{d}} & \Lambda^1(\Omega) & \xrightarrow[\textbf{curl}]{\mathrm{d}} & \Lambda^2(\Omega) \longrightarrow 0 \\ & & \star\updownarrow & & \star\updownarrow & & \star\updownarrow \\ 0 & \longleftarrow & \Lambda^2(\Omega) & \xleftarrow[\textbf{div}]{\mathrm{d}} & \Lambda^1(\Omega) & \xleftarrow[\textbf{rot}]{\mathrm{d}} & \Lambda^0(\Omega) \longleftarrow \mathbb{R} \end{array} . \tag{2.49}$$

In this particular De Rham complex the top row corresponds to inner oriented geometrical objects (equation (2.25)) and the bottom row to outer oriented geometrical objects (equation (2.26)), which gives rise to different operators appearing in the upper and lower complexes (**rot** instead of **grad** and **div** instead of **curl**). In $\mathbb{R}^3$ this does not occur. A graphical representation of the dual De Rham complex in $\mathbb{R}^2$ is given in Figure 2.6.

The Hodge operator gives rise to another operator called the codifferential which does exactly the opposite of the exterior derivative as it maps k-forms to $(k-1)$-forms:

$$\mathrm{d}^\star : \Lambda^k(\Omega) \mapsto \Lambda^{k-1}(\Omega). \tag{2.50}$$

The codifferential is the adjoint of the exterior derivative: $(\mathrm{d}^\star\alpha, \beta) = (\alpha, \mathrm{d}\beta)$. It is defined as:

$$\mathrm{d}^\star\alpha^{(k)} := (-1)^{(n(k+1)+1)} \star \mathrm{d} \star \alpha^{(k)}. \tag{2.51}$$

Having defined the codifferential a formal definition of the Hodge Laplacian can be given:

$$-\Delta := \mathrm{d}\mathrm{d}^\star + \mathrm{d}^\star\mathrm{d}. \tag{2.52}$$

This represent a mapping $\Lambda^k(\Omega) \to \Lambda^k(\Omega)$.

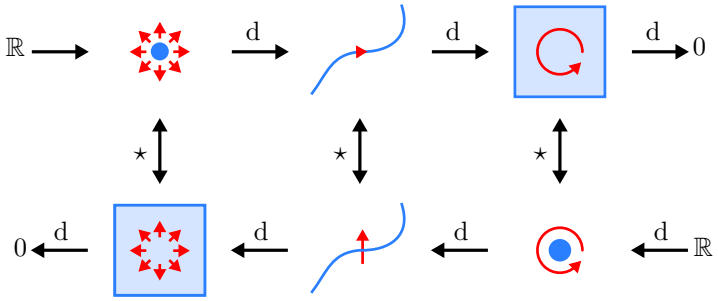


Figure 2.6 – Visual representation of the De Rham complex in $\mathbb{R}^2$.

Now returning to the Poisson problem it can be reformulated as:

$$\mathrm{d}u^{(0)} = q^{(1)}, \tag{2.53}$$
$$\star q^{(1)} = \tilde{q}^{(2)}, \tag{2.54}$$
$$\mathrm{d}\tilde{q}^{(2)} = \tilde{f}^{(3)}, \tag{2.55}$$
$$\star \tilde{f}^{(3)} = f^{(0)}. \tag{2.56}$$

Applying the Hodge to the inner oriented 1-form $q^{(1)}$ conveniently results in the required 2-form $\tilde{q}^{(2)}$ with outer orientation which closes the problem (note the tilde to indicate the distinction). The action of the Hodge in the last step returns it to an inner oriented 0-form again. These steps are visualized below:

$$\begin{array}{ccc} \Lambda^0(\Omega) & \xrightarrow[\textbf{grad}]{\mathrm{d}} & \Lambda^1(\Omega) \\ \star\uparrow & & \star\downarrow \\ \Lambda^3(\Omega) & \xleftarrow[\textbf{div}]{\mathrm{d}} & \Lambda^2(\Omega) \end{array} . \tag{2.57}$$

Markings in red indicate the action of the codifferential.

Finally, with the operators introduced so far the integration by parts formula can be derived here. It will be used frequently throughout this thesis. The starting point is property 3 of the exterior derivative. Substitution of the two k-forms $\alpha^{(k-1)} \in \Lambda^{k-1}(\Omega)$ and $\beta^{(k)} \in \Lambda^k(\Omega)$ yields:

$$\begin{aligned} \mathrm{d}(\alpha \wedge \star\beta) &= \mathrm{d}\alpha \wedge \star\beta + (-1)^{(k-1)}\alpha \wedge (\mathrm{d} \star \beta) \\ &= \mathrm{d}\alpha \wedge \star\beta + (-1)^{(k-1)}\alpha \wedge ((-1)^{(k-1)(n-k+1)} \star\star \mathrm{d} \star \beta) \\ &= \mathrm{d}\alpha \wedge \star\beta + (-1)^{n(k+1)}\alpha \wedge (\star\star \mathrm{d} \star \beta). \end{aligned}$$

The first step is just simply copying property 3. At the second step equation (2.40) is used to introduce additional Hodge operators. At the third step only the exponent on the sign is simplified. Employing the definition of the codifferential as given by equation (2.51), this result is then rewritten as:

$$\begin{aligned} \mathrm{d}\alpha \wedge \star\beta &= (-1)^{(n(k+1)+1)}\alpha \wedge \star(\star \mathrm{d} \star \beta) + \mathrm{d}(\alpha \wedge \star\beta) \\ &= \alpha \wedge \star(\mathrm{d}^\star \beta) + \mathrm{d}(\alpha \wedge \star\beta). \end{aligned}$$

Next this expression is integrated. With the help of the inner product definition of equation (2.38) and the duality implied by Stokes' theorem this yields after some rearrangement:

$$(\mathrm{d}\alpha^{(k-1)}, \beta^{(k)}) - (\alpha^{(k-1)}, \mathrm{d}^\star \beta^{(k)}) = \int_{\partial\Omega} \mathrm{tr}\ \alpha^{(k-1)} \wedge \mathrm{tr}\ \star\beta^{(k)}. \tag{2.58}$$

This concludes this brief discussion of the Hodge star operator. For more information on the Hodge star operator, see Hiptmair [17].

2.6 Pullback

The final operator to be introduced is the pullback, which is simply a transformation. Consider the reference domain Ω_0 with coordinates (ξ, η, ζ) and some arbitrary domain Ω with coordinates (x, y, z). Then a mapping ϕ exists:

$$\phi : \Lambda^k(\Omega_0) \mapsto \Lambda^k(\Omega). \tag{2.59}$$

From this mapping ϕ a new (induced) mapping can be derived. This mapping is called the pullback $\phi^\star$ and is given by:

$$\phi^\star : \Lambda^k(\Omega) \mapsto \Lambda^k(\Omega_0). \tag{2.60}$$

An arbitrary mapping ϕ in $\mathbb{R}^2$ and its associated pullback is visualized in Figure 2.7. The

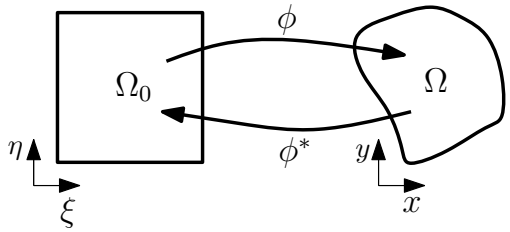


Figure 2.7 – Visualization of an arbitrary mapping in $\mathbb{R}^2$.

pullback has a number of important properties which are listed below. Once more $\alpha \in \Lambda^k(\Omega)$ and $\beta \in \Lambda^l(\Omega)$ (unless otherwise indicated).

1. ϕ^* is additive, $\phi^\star(\alpha + \beta) = \phi^\star(\alpha) + \phi^\star(\beta)$ (given that $k = l$)
2. ϕ^* commutes with $\wedge$, $\phi^\star(\alpha \wedge \beta) = (\phi^\star(\alpha)) \wedge (\phi^\star(\beta))$
3. ϕ^* commutes with d, $\mathrm{d}(\phi^\star(\alpha)) = \phi^\star(\mathrm{d}(\alpha))$
4. if there exist two (different) mappings ϕ and ψ, then $(\psi \circ \phi)^\star = \phi^\star \circ \psi^\star$

Additionally, given a mapping ϕ and its associated pullback $\phi^\star$ then the following relation holds:

$$\int_{\phi(\Omega)} \alpha^{(k)} = \int_\Omega \phi^\star(\alpha^{(k)}). \tag{2.61}$$

In other words, the pullback is the adjoint of the mapping ϕ:

$$\langle \alpha^{(k)}, \phi(\Omega) \rangle = \langle \phi^\star \alpha^{(k)}, \Omega \rangle. \tag{2.62}$$

It is important to note that the pull-back operator acts differently on the different k-forms. For $\mathbb{R}^2$ and $\mathbb{R}^3$ examples are given below, starting with 0-forms (for the original k-forms, see Section 2.1):

$$\begin{aligned}
\phi^\star\alpha^{(0)} =& a(x(\xi,\eta),y(\xi,\eta)) = a(\boldsymbol{\xi}) && \in \Lambda^0(\mathbb{R}^2),\\
\phi^\star\alpha^{(0)} =& a(x(\xi,\eta,\zeta),y(\xi,\eta,\zeta),z(\xi,\eta,\zeta)) = a(\boldsymbol{\xi}) && \in \Lambda^0(\mathbb{R}^3).
\end{aligned}$$

Note that $\boldsymbol{\xi} = (\xi,\eta)$ and $\boldsymbol{\xi} = (\xi,\eta,\zeta)$ in $\mathbb{R}^2$ and $\mathbb{R}^3$ respectively. For 1-forms it becomes slightly more involved:

$$\begin{aligned}
\phi^\star\alpha^{(1)} =& \left(a(\boldsymbol{\xi})\frac{\partial x}{\partial \xi} + \ b(\boldsymbol{\xi})\frac{\partial y}{\partial \xi}\right)\mathrm{d}\xi + \left(a(\boldsymbol{\xi})\frac{\partial x}{\partial \eta} + b(\boldsymbol{\xi})\frac{\partial y}{\partial \eta}\right)\mathrm{d}\eta \quad \in \Lambda^1(\mathbb{R}^2),\\
\phi^\star\alpha^{(1)} =& \left(a(\boldsymbol{\xi})\frac{\partial x}{\partial \xi} + b(\boldsymbol{\xi})\frac{\partial y}{\partial \xi} + c(\boldsymbol{\xi})\frac{\partial z}{\partial \xi}\right)\mathrm{d}\xi +\\
& \left(a(\boldsymbol{\xi})\frac{\partial x}{\partial \eta} + b(\boldsymbol{\xi})\frac{\partial y}{\partial \eta} + c(\boldsymbol{\xi})\frac{\partial z}{\partial \eta}\right)\mathrm{d}\eta +\\
& \left(a(\boldsymbol{\xi})\frac{\partial x}{\partial \zeta} + b(\boldsymbol{\xi})\frac{\partial y}{\partial \zeta} + c(\boldsymbol{\xi})\frac{\partial z}{\partial \zeta}\right)\mathrm{d}\zeta \qquad \in \Lambda^1(\mathbb{R}^3).
\end{aligned}$$

Next are the 2-forms, which requires determinants:

$$\begin{aligned}
\phi^\star\alpha^{(2)} =& a(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta}\\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta}\end{pmatrix}\mathrm{d}\xi\wedge\mathrm{d}\eta \quad \in \Lambda^2(\mathbb{R}^2),\\
\phi^\star\alpha^{(2)} =& \left(a(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta}\\ \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta}\end{pmatrix} + b(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta}\\ \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta}\end{pmatrix} + c(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta}\\ \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta}\end{pmatrix}\right)\mathrm{d}\eta\wedge\mathrm{d}\zeta +\\
& \left(a(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial y}{\partial \zeta} & \frac{\partial y}{\partial \xi}\\ \frac{\partial z}{\partial \zeta} & \frac{\partial z}{\partial \xi}\end{pmatrix} + b(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial z}{\partial \zeta} & \frac{\partial z}{\partial \xi}\\ \frac{\partial x}{\partial \zeta} & \frac{\partial x}{\partial \xi}\end{pmatrix} + c(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial x}{\partial \zeta} & \frac{\partial x}{\partial \xi}\\ \frac{\partial y}{\partial \zeta} & \frac{\partial y}{\partial \xi}\end{pmatrix}\right)\mathrm{d}\zeta\wedge\mathrm{d}\xi +\\
& \left(a(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \eta}\\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta}\end{pmatrix} + b(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta}\\ \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta}\end{pmatrix} + c(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta}\\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta}\end{pmatrix}\right)\mathrm{d}\xi\wedge\mathrm{d}\eta\\
& \in \Lambda^2(\mathbb{R}^3).
\end{aligned}$$

And finally the 3-forms:

$$\phi^\star\alpha^{(3)} = a(\boldsymbol{\xi})\det\begin{pmatrix}\frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta}\\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta}\\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta}\end{pmatrix}\mathrm{d}\xi\wedge\mathrm{d}\eta\wedge\mathrm{d}\zeta \quad \in \Lambda^3(\mathbb{R}^3).$$

This concludes a brief overview of differential geometry. In the next chapter its discrete counterpart, algebraic topology, is discussed.

CHAPTER 3
ALGEBRAIC TOPOLOGY

Previously the basic elements of differential geometry were introduced as a means to represent continuous PDE's. With differential forms being associated with geometrical objects, accompanying relations and operators reflected much of this as well. The next step is to find their discrete analogues which allows for similar representations in a discrete setting. These are provided by the field of algebraic topology, the topic of this chapter. It supplies the tools necessary to define discrete representations of PDE's. This chapter is built up out of two large sections. The first (Section 3.1) introduces the main building blocks, the chains. The second (Section 3.2) builds on the previous one by introducing the co-chains. Associated operators and terminology are presented along the way. Many of the important concepts of algebraic topology have been extensively covered by Tonti [32] and Mattiussi [24].

3.1 Chains

The field of algebraic topology is, as its name suggests, solely concerned with topological structures. Therefore notions such as distances and angles do not bear any relevance, only the connectivity is of importance. Its main building blocks are the so-called p-cells: p-dimensional (geometrical) objects of which only their connectivity with other p-cells is known. Analogous to the four p-dimensional objects in $\mathbb{R}^3$ there are 0-cells (points), 1-cells (lines), 2-cells (surfaces) and 3-cells (volumes), as visualized in Figure 3.1. A mesh resulting from a discretization could

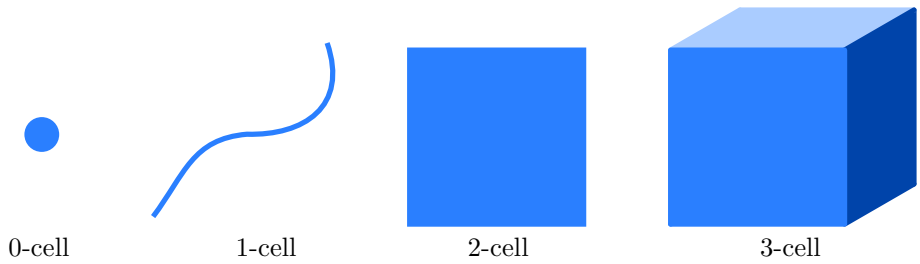


Figure 3.1 – Examples of p-cells in $\mathbb{R}^3$.

be considered to be built up of p-cells. Such an assembly of p-cells is called a cell complex K. An example of a cell complex in $\mathbb{R}^2$ is depicted in Figure 3.2. A finite collection of n_p oriented

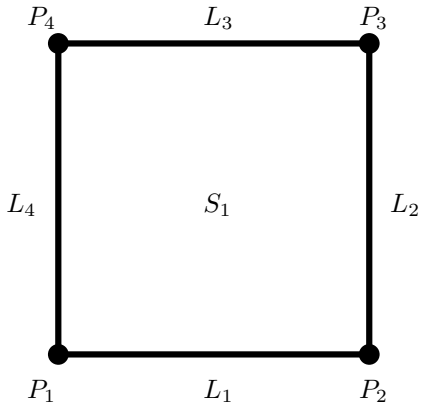


Figure 3.2 – Example of a cell complex in $\mathbb{R}^2$ with numbered p-cells. Points (0-cells) are denoted with a P, lines (1-cells) with an L and surfaces (2-cells) with an S.

p-cells is called a p-chain and is formally defined as:

$$c_{(p)} = \sum_{i=1}^{n_p} m^i_{(p)} \sigma^i_{(p)}. \tag{3.1}$$

Here $\sigma^i_{(p)}$ represents the p-cells and the $m^i_{(p)}$ are scalar coefficients denoting orientation which take on the values 0, +1 or -1. When the weight of a particular p-cell is 0 it is not included in the chain. When it has a value of +1 or -1 it is included in the chain with either positive or negative orientation respectively. This notion of orientation is similar to the one introduced in the previous chapter. An example of an (inner) oriented cell complex is depicted in Figure 3.3.

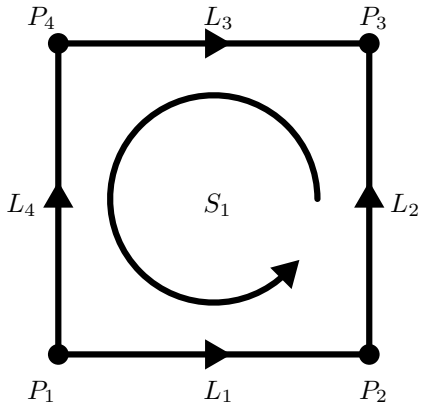


Figure 3.3 – Example of an (inner) oriented cell complex in $\mathbb{R}^2$. Points (or 0-cells) are considered sources as opposed to sinks.

In algebraic topology there exists a boundary operator ∂ which acts on chains as:

$$\partial c_{(p)} = \partial \left(\sum_{i=1}^{n_p} m^i_{(p)} \sigma^i_{(p)} \right) = \sum_{i=1}^{n_p} m^i_{(p)} \partial \sigma^i_{(p)}. \tag{3.2}$$

The boundary operator ∂ represents a homomorphism from the space of p-chains to the space of $(p-1)$-chains:

$$\partial : C_p(K) \mapsto C_{p-1}(K). \tag{3.3}$$

It has a property similar to the exterior derivative from differential geometry, i.e. the boundary of the boundary is zero:

$$\partial^2 c_{(p)} := \partial(\partial c_{(p)}) = 0. \tag{3.4}$$

Additionally, an exact sequence can be constructed:

$$0 \longleftarrow C_0(K) \xleftarrow{\partial} C_1(K) \xleftarrow{\partial} C_2(K) \xleftarrow{\partial} C_3(K) \xleftarrow{\partial} 0 \ . \tag{3.5}$$

In order to demonstrate some of these concepts, the boundary operator is applied to the example cell complex from Figure 3.3. Given the indicated labeling and orientation, applying the boundary operator to the single 2-cell yields:

$$\partial S_1 = \underbrace{\begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix}}_{\mathbb{E}^{(2,1)}} \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{pmatrix}. \tag{3.6}$$

The incidence matrix $\mathbb{E}^{(2,1)}$ relates the 2-cell to its boundary of 1-cells, as indicated by the superscript. Similarly, applying the boundary operator to the 1-cells yields:

$$\partial \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{pmatrix} = \begin{pmatrix} \partial L_1 \\ \partial L_2 \\ \partial L_3 \\ \partial L_4 \end{pmatrix} = \underbrace{\begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{pmatrix}}_{\mathbb{E}^{(1,0)}} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix}. \tag{3.7}$$

With both incidence matrices known the identity $\partial^2 := 0$ can easily be shown to hold for this particular example:

$$\mathbb{E}^{(2,1)} \cdot \mathbb{E}^{(1,0)} = \begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.8}$$

These steps are also visualized in Figure 3.4. With a consistent labeling (or numbering) as well as orientation the construction of the incidence matrices for larger cell complexes should remain a relatively straightforward procedure.

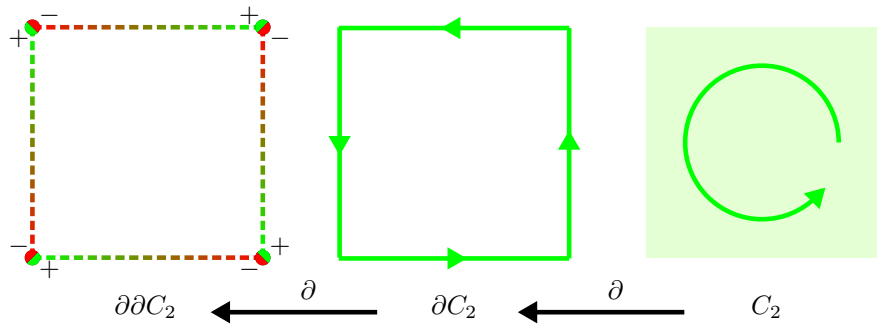


Figure 3.4 – The boundary operator in $\mathbb{R}^2$.

3.2 Cochains

In this section the concept of the cochains is introduced. Simply put, a cochain associates with each p-cell a value:

$$c^{(p)} = \sum_{i=1}^{n_p} a_i^{(p)} \tau_i^{(p)}. \tag{3.9}$$

Here $\tau_i^{(p)}$ represents a chain which only assigns 1's or 0's to p-cells to include or exclude them from this particular cochain while the list of numbers $a_i^{(p)}$ assigns values to these p-cells. Cochains are the result of integrating a k-form over its associated k-dimensional geometrical object or p-chain (when $p = k$):

$$\int_{c_{(p)}} \alpha^{(k)} = \sum_{i=1}^{n_p} m_{(p)}^i \int_{\sigma_{(p)}^i} \alpha^{(k)}. \tag{3.10}$$

Note that:

$$\langle \tau_i^{(p)}, \sigma_{(p)}^j \rangle = \delta_{ij}. \tag{3.11}$$

The duality pairing of a p-chain and a p-co-chain then yields:

$$\langle c^{(p)}, c_{(p)} \rangle = \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} a_i^{(p)} m_j^{(p)} \langle \tau_i^{(p)}, \sigma_{(p)}^j \rangle = \sum_{i=1}^{n_p} a_i^{(p)} m_i^{(p)}. \tag{3.12}$$

This duality pairing is important as it allows the definition of a topological equivalent of the generalized Stokes Theorem. Within the framework of algebraic topology this is represented as follows:

$$\langle \delta c^{(p)}, c_{(p+1)} \rangle = \langle c^{(p)}, \partial c_{(p+1)} \rangle. \tag{3.13}$$

Which is indeed the discrete analogue of equation (2.28). To continue with the discrete analogues, the coboundary operator δ behaves like the exterior derivative as it maps p-cochains into $(p+1)$-cochains:

$$\delta : C^p(K) \mapsto C^{p+1}(K). \tag{3.14}$$

And obeys the familiar identity:

$$\delta^2 c^{(p)} := \delta(\delta c^{(p)}) = 0. \tag{3.15}$$

Evidently, this corresponds to property 4 of the exterior derivative d (which the boundary operator ∂ also obeys, as seen in equation (3.4)).

Additionally, since δ is the adjoint of ∂ an equivalent De Rham sequence of p-cochains can be constructed which in $\mathbb{R}^3$ looks as follows:

$$0 \longrightarrow C^0(K) \xrightarrow{\delta} C^1(K) \xrightarrow{\delta} C^2(K) \xrightarrow{\delta} C^3(K) \xrightarrow{\delta} 0 \ . \tag{3.16}$$

Similar to Figure 3.4 a visual representation of the coboundary operator is given in Figure 3.5.

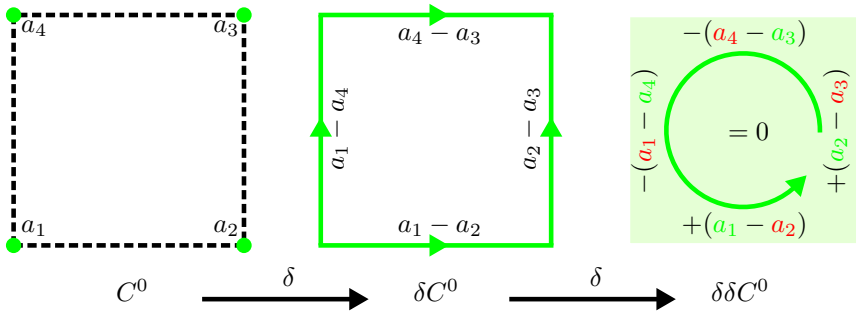


Figure 3.5 – The coboundary operator in $\mathbb{R}^2$.

Clearly, the De Rham complex of the coboundary operator behaves in much the same way as the exterior derivative. In order to emphasize the equivalence, the De Rham complex is repeated here once more:

$$\mathbb{R} \longrightarrow \Lambda^0(\Omega) \xrightarrow[\textbf{grad}]{\mathrm{d}} \Lambda^1(\Omega) \xrightarrow[\textbf{curl}]{\mathrm{d}} \Lambda^2(\Omega) \xrightarrow[\textbf{div}]{\mathrm{d}} \Lambda^3(\Omega) \xrightarrow{\mathrm{d}} 0 \ .$$

In fact, the coboundary operator fulfills discretely the same functions as the gradient, curl and divergence operators from classical vector calculus (and thus the exterior derivative).

Finally, the coboundary operator is discretely represented as an incidence matrix as well. Although it is denoted as $\mathbb{D}$, it is equal to $\mathbb{E}$ from the previous section.

CHAPTER 4

THE MIMETIC FRAMEWORK

The previous two chapters were concerned with an introduction to the fields of differential geometry and algebraic topology. The reason for their introduction might still be somewhat obscured, something which this particular chapter aims to resolve. The main goal is to combine the continuous world of differential forms with the discrete world of (co)chains, completing what is called the mimetic framework. To that end, three new operators are introduced in Section 4.1. The first two are the reduction and reconstruction which are covered in subsections 4.1.1 and 4.1.2 respectively. Combining these operators yields a third one as will be explained in subsection 4.1.3. Finally, this chapter ends with some practical examples of the implementation encountered in this work (Section 4.2). Much of the theory discussed here has already been extensively covered by for example Bochev in [2] and more recently by Kreeft et al. [22].

4.1 Mimetic operators

As mentioned in the introduction, the goal of this chapter is to connect the continuous and discrete worlds of differential geometry and algebraic topology respectively. This requires a number of new operators, the first being the reduction operator.

4.1.1 Reduction

The first operator to be introduced is the reduction operator $\mathcal{R}$ which maps differential forms to cochains:

$$\mathcal{R} : \Lambda^k(\Omega) \mapsto C^k(K). \tag{4.1}$$

And is defined by integration as:

$$\langle \mathcal{R}\alpha^{(k)}, c_{(p)} \rangle = \int_{c_{(p)}} \alpha^{(k)}. \tag{4.2}$$

It has the following important property:

$$\mathcal{R}\mathrm{d} = \delta\mathcal{R}. \tag{4.3}$$

Denoted as the commuting diagram property, it can be visualized as follows:

$$\begin{array}{ccc} \Lambda^k & \xrightarrow{\mathrm{d}} & \Lambda^{k+1} \\ \mathcal{R}\downarrow & & \mathcal{R}\downarrow \\ C^k & \xrightarrow{\delta} & C^{k+1} \end{array} . \tag{4.4}$$

Hence there is no difference in first applying the exterior derivative and then mapping the result to the discrete space or first applying the mapping and then the coboundary operator. The generalized Stokes' theorem, in combination with the duality of the boundary and coboundary operators as given by equation (3.13), provides the proof for this identity:

$$\begin{aligned} \langle \mathcal{R}\mathrm{d}\alpha^{(k)}, c_{(p)} \rangle = \int_{c_{(p)}} \mathrm{d}\alpha^{(k)} = \int_{\partial c_{(p)}} \alpha^{(k)} = \\ \langle \mathcal{R}\alpha^{(k)}, \partial c_{(p)} \rangle = \langle \delta \mathcal{R}\alpha^{(k)}, c_{(p)} \rangle. \end{aligned} \tag{4.5}$$

Finally, the reduction commutes with the continuous and discrete pullback:

$$\mathcal{R}\phi^\star = \phi^\sharp \mathcal{R}. \tag{4.6}$$

Where $\phi^\sharp$ represents the discrete pullback which acts on cochains instead of differential forms.

4.1.2 Reconstruction

Having mapped k-forms to cochains as shown in the previous section, another operator is required to obtain a continuous representation again. This mapping is performed using the reconstruction operator $\mathcal{I}$:

$$\mathcal{I} : C^k(K) \mapsto \Lambda^k(\Omega). \tag{4.7}$$

In practical terms, the reconstruction defines the interpolation of a (discrete) solution to a continuous one. The spectral element approach taken in this work determines the type of interpolants used, as will be shown in Section 4.2. Hence, this operator is not fixed. It should nevertheless satisfy a number of requirements, the first being the commuting diagram property:

$$\mathrm{d}\mathcal{I} = \mathcal{I}\delta. \tag{4.8}$$

This is visualized as follows:

$$\begin{array}{ccc} \Lambda_h^k & \xrightarrow{\mathrm{d}} & \Lambda_h^{k+1} \\ \mathcal{I}\uparrow & & \mathcal{I}\uparrow \\ C^k & \xrightarrow{\delta} & C^{k+1} \end{array} . \tag{4.9}$$

Additionally, it must be consistent:

$$\mathcal{R}\mathcal{I} = I. \tag{4.10}$$

In other words: $\mathcal{I}$ must be the right-inverse of $\mathcal{R}$ (where I represents the identity). The second one is the approximation property:

$$\mathcal{I}\mathcal{R} = I + \mathcal{O}(h^p). \tag{4.11}$$

The term $\mathcal{O}(h^p)$ represents the truncation error in terms of the grid size h and polynomial order p. Additionally, like the reduction, it must commute with the continuous and discrete pullback.

$$\mathcal{I}\phi^\sharp = \phi^\star \mathcal{I} \quad \text{on } C^k K. \tag{4.12}$$

4.1.3 Projection

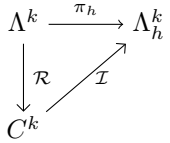
Having defined the reduction and reconstruction operators, their combined action is bundled in a single projection operator π_h:

$$\pi_h = \mathcal{I} \circ \mathcal{R}. \tag{4.13}$$

Which constitutes a mapping:

$$\pi_h : \Lambda^k(\Omega) \mapsto \Lambda_h^k(\Omega). \tag{4.14}$$

The action of this new projection operator can be visualized as follows:



. (4.15)

This combined action of the reduction and reconstruction is an homomorphism. Given two k-forms $\alpha^{(k)}, \beta^{(k)} \in \Lambda^k(\Omega)$ then:

$$\pi_h(\alpha^{(k)} + \beta^{(k)}) = \pi_h(\alpha^{(k)}) + \pi_h(\beta^{(k)}). \tag{4.16}$$

More importantly, the projection commutes with differentiation:

$$\mathrm{d}\pi_h = \pi_h \mathrm{d}. \tag{4.17}$$

And finally, as the reduction and reconstruction operators both commute with the pullback, so does the projection:

$$\phi^\star \pi = \pi \phi^\star. \tag{4.18}$$

In the next section a number of discretization examples will clarify these operators further.

4.2 Implementation

As already hinted at in Section 4.1.2, the reconstruction operator is determined by the type of interpolant used. In the case of 0-forms, a straightforward nodal interpolation using Lagrange polynomials suffices. However k-forms of higher degree can not be adequately reconstructed using nodal interpolation. Therefore the so-called edge functions must be introduced. This section will discuss both, accompanied with some examples. Higher dimensional considerations are given at the end.

4.2.1 Reconstructing 0-forms

As mentioned, 0-forms are reconstructed using nodal interpolants. In this work the Lagrange polynomials are used for that purpose. Only a definition will be given here, but more details can be found in (for example) the work by Canuto [7].

The definition of the Lagrange polynomials starts with the introduction of a set of $(N+1)$ points x_i $(0 \leq i \leq N)$. Through these points a Lagrange polynomial $h_i(x)$ of order N can be defined which has unit value at the nodal points x_i and is zero at x_j $(i \neq j)$:

$$h_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}. \tag{4.19}$$

The product form reveals how a Lagrange polynomial can be constructed:

$$h_i(x) = \frac{\prod_{j=0, j\neq i}^{N} (x - x_j)}{\prod_{j=0, j\neq i}^{N} (x_i - x_j)}. \tag{4.20}$$

Exactly because of the property given by equation (4.19) Lagrange polynomials are often used as an interpolation basis. The set of nodal points x_i are chosen to be the so-called GLL nodes which are based on the roots of the Gauss-Lobatto polynomials. Note that a clever choice of nodal points positively affects both the stability and the conditioning of the system (see Karniadakis and Sherwin [18]).

To give an example of a Lagrange polynomial, consider the case $N = 4$. Evaluated on the domain $[-1, 1]$, the second Lagrange polynomial $h_2(x)$ looks as depicted in Figure 4.1. As

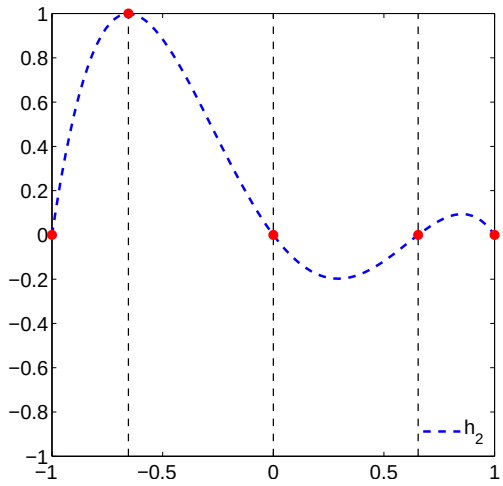


Figure 4.1 – Visualization of the fourth order Lagrange polynomial $h_2(x)$ on the standard domain $[-1, 1]$ with GLL nodes.

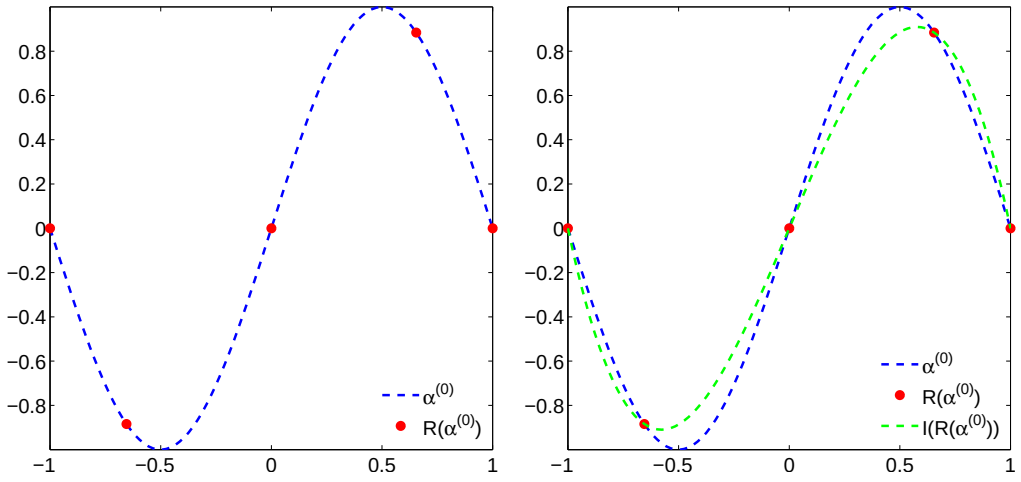
expected, at the second GLL node the Lagrange polynomial reaches a value of 1 while at any of the other nodes it is 0.

Reconstructing a 0-form is now a straightforward procedure. Given the 0-form $\alpha^{(0)} = a(x)$, then reconstruction using Lagrange polynomials yields:

$$\alpha_h^{(0)}(x) = \sum_{i=0}^{N} a(x_i) h_i(x) = \sum_{i=1}^{N} a_i h_i(x). \tag{4.21}$$

Note that by definition of the Lagrange polynomial this reconstruction is always exact at the nodal points x_i. Whenever $a(x)$ itself is a polynomial of order N then $\alpha_h^{(0)}(x)$ becomes exact, i.e. $\alpha_h^{(0)} = \alpha^{(0)}$.

To give an example of a reconstruction, consider the 0-form $\alpha^{(0)} = \sin(\pi x)$. Figure 4.2 contains two plots: the one on the left represents the reduction while the one on the right is the reconstruction using Lagrange polynomials. In this particular example the reduction (Figure



(a) Reduction $\mathcal{R}$. **(b)** Reconstruction $\mathcal{I}$.

Figure 4.2 – Visualization of the reduction and reconstruction operators acting on the 0-form $\alpha^{(0)} = \sin(\pi x)$ in $\mathbb{R}$.

4.2a) represents a mapping to a cochain consisting of five 0-cells (the GLL nodes). Looking at the reconstruction (Figure 4.2b), there appears to be a significant discrepancy as a result of the low order approximation (predicted by the property given by equation (4.11)). Evidently, increasing the order (or using more than one element) would reduce $\mathcal{O}(h^p)$.

4.2.2 Reconstructing 1-forms

While the reconstruction of 0-forms requires nodal interpolation, higher degree k-forms require the use of edge functions as introduced by Gerritsma [12] and Robidoux [31]. Their definition will be given here. For more details, see the provided references.

Edge functions are simply defined as:

$$e_j(x) = -\sum_{i=0}^{N} \mathrm{d}h_i(x), \quad j = 1, ..., N. \tag{4.22}$$

They have a similar property as the Lagrange polynomials:

$$\int_{x_{j-1}}^{x_j} e_i(x) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} . \tag{4.23}$$

In order to clarify this, consider Figure 4.3 which depicts an edge function. This graph shows the fourth order edge polynomial $e_2(x)$ based on a mesh with $N = 4$ (similar to the Lagrange example of Figure 4.1). It is clear that the integral evaluated over each line segment is 0, except at the second one where it is 1.

Reconstructing a 1-form using edge functions occurs in much the same way as reconstructing a 0-form. The only difference is the interpolant. Hence, given the 1-form $\alpha^{(1)} = a(x)\mathrm{d}x$, then its

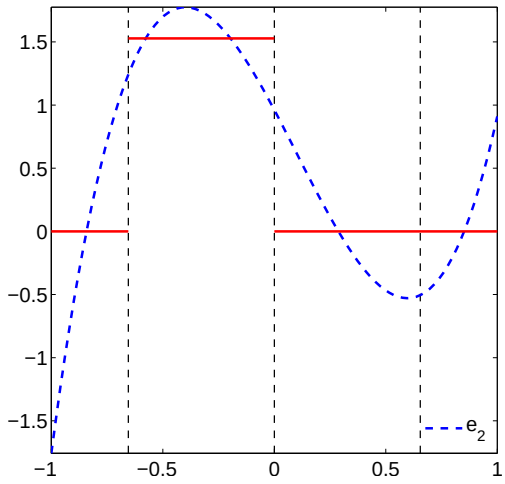


Figure 4.3 – Visualization of the fourth order edge function $e_2(x)$ on the standard domain $[-1, 1]$ with GLL nodes.

reconstruction is given by:

$$\alpha_h^{(1)}(x) = \sum_{i=1}^{N} a(x_i) e_i(x) = \sum_{i=1}^{N} a_i e_i(x). \tag{4.24}$$

To give an example, consider the 1-form $\alpha^{(a)} = \sin(\pi x)\mathrm{d}x$. Figure 4.4 contains its reduction on the left and reconstruction (using edge functions) on the right. The 1-form is reduced to a cochain consisting of four 1-cells (the line segments in between the GLL nodes). Looking at the reconstruction (Figure 4.4b), again there appears to be a significant discrepancy as a result of the low order.

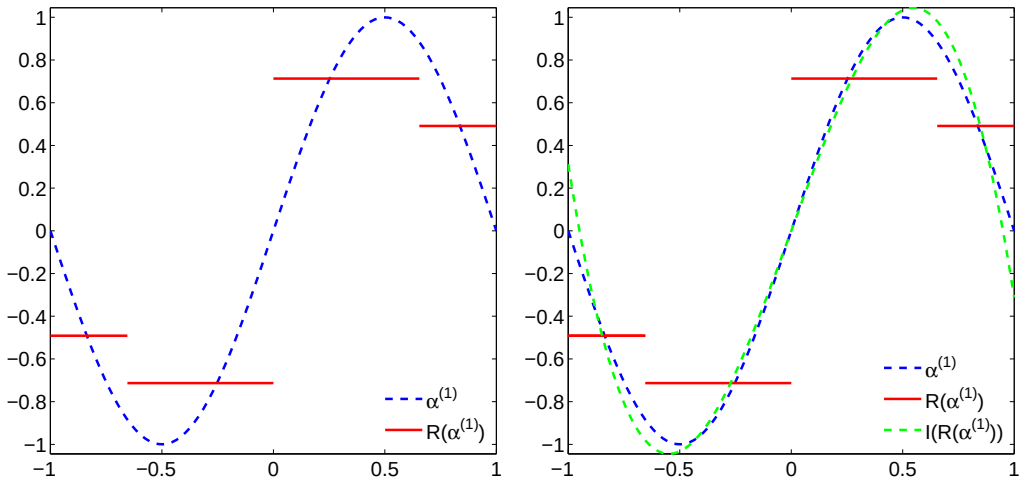
4.2.3 Higher dimensional considerations

So far reduction and reconstruction have been demonstrated in $\mathbb{R}$ only. As this work focuses on problems in $\mathbb{R}^2$, an extension to higher dimensions is needed. This does not only affects the reconstruction of 0-forms and 1-forms, it introduces 2-forms as well. Fortunately, many of the concepts in $\mathbb{R}$ are easily extended using the tensor product. For simplicity, the final result is summarized in Table 4.1. To finish this chapter, an example of the reduction and reconstruction

Table 4.1 – Reconstruction of k-forms in $\mathbb{R}$.

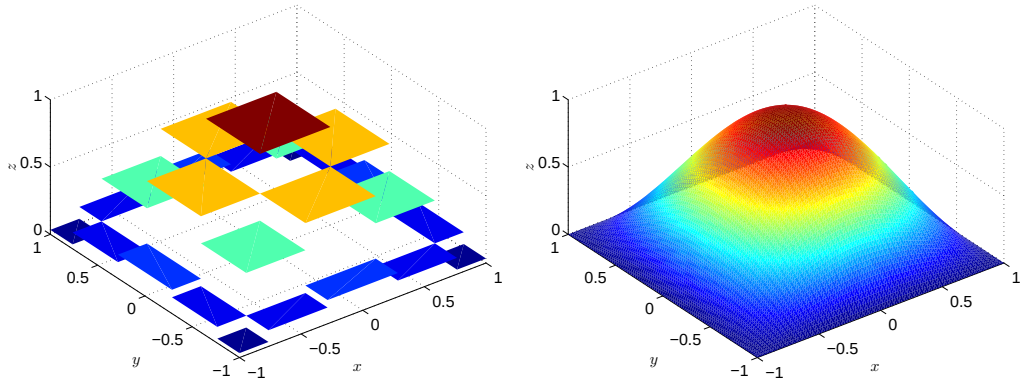
k-form	Reconstruction
$\alpha^{(0)} = a(x,y)$	$\sum_{i=0}^{N}\sum_{i=0}^{N} a_{ij} h_i(x) h_j(y)$
$\alpha^{(1)} = a(x,y)\mathrm{d}x + b(x,y)\mathrm{d}y$	$\sum_{i=1}^{N}\sum_{i=0}^{N} a_{ij} e_i(x) h_j(y) + \sum_{i=0}^{N}\sum_{i=1}^{N} b_{ij} h_i(x) e_j(y)$
$\alpha^{(2)} = a(x,y)\mathrm{d}x\mathrm{d}y$	$\sum_{i=1}^{N}\sum_{i=1}^{N} a_{ij} e_i(x) e_j(y)$

of 2-forms is given in Figure 4.5. A similar reconstruction of 0-forms and 1-forms in $\mathbb{R}^2$ is omitted as the added dimension does not affect the visual representation of lines and points (they remain 0- and 1-dimensional objects).



(a) Reduction $\mathcal{R}$. **(b)** Reconstruction $\mathcal{I}$.

Figure 4.4 – Visualization of the reduction and reconstruction operators acting on the 1-form $\alpha^{(1)} = \sin(\pi x)\mathrm{d}x$ in $\mathbb{R}$.



(a) Reduction $\mathcal{R}$. **(b)** Reconstruction $\mathcal{I}$.

Figure 4.5 – Visualization of the reduction and reconstruction operators acting on the 2-form $\alpha^{(2)} = \cos(\frac{1}{2}\pi x)cos(\frac{1}{2}\pi y)\mathrm{d}x\mathrm{d}y$ in $\mathbb{R}^2$.

CHAPTER 5

SCALAR LAPLACE IN TWO DIMENSIONS

In this chapter the MSEM is demonstrated by solving the two-dimensional Poisson problem for 0-forms and 2-forms. A good understanding of this particular problem will be of great benefit when the application of the MEM within the mimetic framework is explained in Chapter 6. This chapter is therefore structured as follows. The two-dimensional problem description is outlined first in Section 5.1. In subsequent sections this will be solved for both 0-forms (Section 5.2) and 2-forms (Section 5.3). For both of these analyses the starting point is the continuous formulation from which a step by step approach is taken which ends at the construction of the system matrices itself.

5.1 Two-dimensional problem description

The classical formulation of the Poisson problem, subject to a homogeneous Dirichlet boundary condition, is given by the following equation:

$$\begin{aligned} -\Delta u &= f \text{ on } \Omega, \\ u &= 0 \text{ on } \partial\Omega. \end{aligned} \tag{5.1}$$

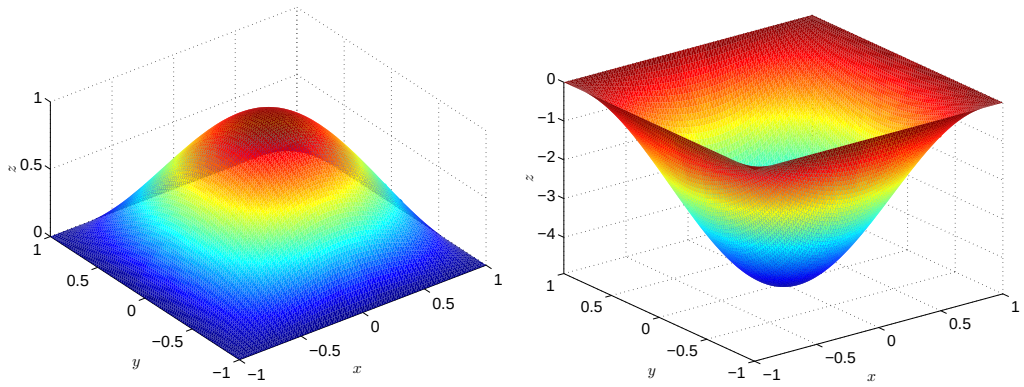
The domain $\Omega : x, y \subset \mathbb{R}$ is of dimensions $(-1, 1) \times (-1, 1)$. For the purpose of demonstrating the MSEM an exact solution $u(x, y)$ is chosen to be:

$$u(x, y) = \cos(\frac{1}{2}\pi x)\cos(\frac{1}{2}\pi y). \tag{5.2}$$

The Laplacian of $u(x, y)$ results in the right-hand side source term $f(x, y)$ which is equal to:

$$f(x, y) = \frac{1}{2}\pi^2 \cos(\frac{1}{2}\pi x)\cos(\frac{1}{2}\pi y). \tag{5.3}$$

Graphically $u(x, y)$ and $f(x, y)$ are depicted in Figures 5.1a and 5.1b. In the next section equation this two-dimensional Poisson problem will be solved for 0-forms.



(a) $u(x,y) = \cos(\frac{1}{2}\pi x)\cos(\frac{1}{2}\pi y)$. **(b)** $f(x,y) = \frac{1}{2}\pi^2\cos(\frac{1}{2}\pi x)\cos(\frac{1}{2}\pi y)$.

Figure 5.1 – Graphical representation of the exact solution of the test problem.

5.2 0-forms

Recall the general expression for the Hodge Laplacian:

$$-\Delta = \mathrm{d}^\star\mathrm{d} + \mathrm{d}\mathrm{d}^\star.$$

With the help of the (dual) De Rham complex in 2D

$$\begin{array}{ccccccccc} \mathbb{R} & \longrightarrow & \Lambda^0(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^1(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^2(\Omega) & \xrightarrow{\mathrm{d}} & 0 \\ & & \star\updownarrow & & \star\updownarrow & & \star\updownarrow & & \\ 0 & \xleftarrow{\mathrm{d}} & \Lambda^2(\Omega) & \xleftarrow{\mathrm{d}} & \Lambda^1(\Omega) & \xleftarrow{\mathrm{d}} & \Lambda^0(\Omega) & \longleftarrow & \mathbb{R} \end{array},$$

the Hodge Laplacian for 0-forms reduces to:

$$\mathrm{d}^\star\mathrm{d}u^{(0)} = f^{(0)}. \tag{5.4}$$

Although equation (5.4) can be written as a set of first-order equations, in the end it can always be simplified to just a single one. Equation (5.4) is cast into a variational formulation:

$$(v^{(0)}, \mathrm{d}^\star\mathrm{d}u^{(0)}) = (v^{(0)}, f^{(0)}). \tag{5.5}$$

With the help of the integration by parts formula as given by equation (2.58) in Chapter 2 the right-hand side reduces to:

$$(v^{(0)}, \mathrm{d}^\star\mathrm{d}u^{(0)}) = (\mathrm{d}v^{(0)}, \mathrm{d}u^{(0)}) - \int_{\partial\Omega} \mathrm{tr}\ v^{(0)} \wedge \mathrm{tr}\ \star\,\mathrm{d}u^{(0)}. \tag{5.6}$$

The boundary integral vanishes for Dirichlet boundary conditions, yielding a simplified equation:

$$(\mathrm{d}v^{(0)}, \mathrm{d}u^{(0)}) = (v^{(0)}, f^{(0)}). \tag{5.7}$$

Hence, the goal is to find $u^{(0)} \in \Lambda^0(\Omega)$ given $f^{(0)} \in \Lambda^0(\Omega)$ for all test functions $v^{(0)} \in \Lambda^0(\Omega)$ subject to the boundary condition $u = 0$ on $\partial\Omega$. Using the definition of the inner product, equation (2.39) in Chapter 2, this is expanded as:

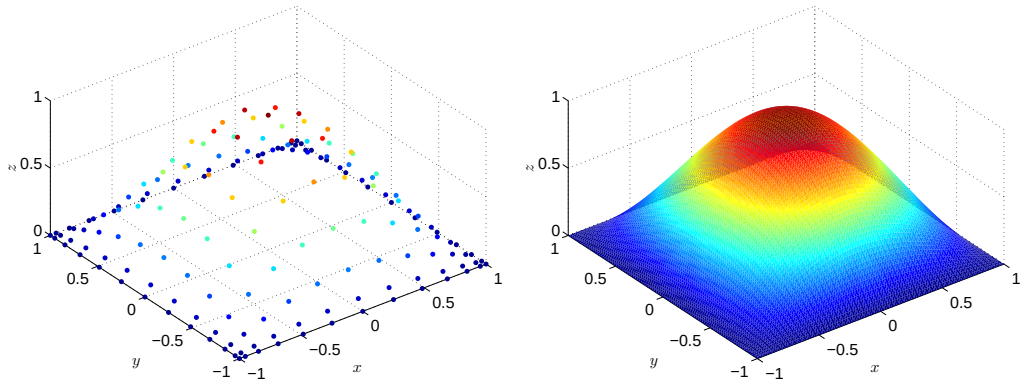
$$\left(\mathrm{d}v^{(0)}, \mathrm{d}u^{(0)}\right)_\Omega = \int_\Omega \mathrm{d}v^{(0)} \wedge \star\mathrm{d}u^{(0)}, \tag{5.8}$$

$$\left(v^{(0)}, f^{(0)}\right)_\Omega = \int_\Omega v^{(0)} \wedge \star f^{(0)}. \tag{5.9}$$

Discretizing (as shown in Appendix A.1) eventually yields the following matrix system:

$$\mathbb{D}^T M_1 \mathbb{D}\mathbf{u} = M_0\mathbf{f}. \tag{5.10}$$

With $\mathbb{D}$ the incidence matrix. This can be solved for the 0-form $u^{(0)}$. A cochain representation and its reconstruction on the standard element with order 12 is depicted in Figure 5.2.



(a) Cochain representation of the solution. **(b)** Reconstruction of the cochains.

Figure 5.2 – Solution of the 0-form Poisson problem on the standard element with order 12.

This particular example considers only a single element. It can however be extended to multiple elements. This requires, like every other Finite Element Method (FEM)/SEM code, both a global numbering and local numbering of the nodes (i.e. the 0-forms $u^{(0)}$). Whenever there are multiple elements, the system matrix $\mathbb{D}^T M_1 \mathbb{D}$ and accompanying right-hand side vector $M_0\mathbf{f}$ (as given by equation (5.10)) are first constructed for each element. Using then the global numbering, each of these matrices is *added* to the global system matrix. In practical terms, whenever nodes are shared between neighboring elements, the contributions from both element matrices are added up. As an example, consider the two linear elements depicted in Figure 5.3. The local numbering is indicated. These two elements must be connected at two nodes. Using

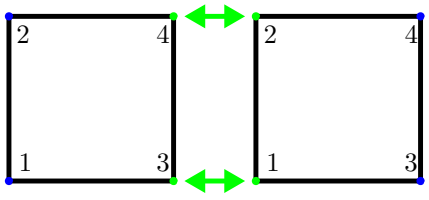


Figure 5.3 – Local element numbering of 0-forms.

global numbering, this is depicted in Figure 5.4. For each of the two elements a mass matrix can be constructed:

$$K^{(n)} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix}. \tag{5.11}$$

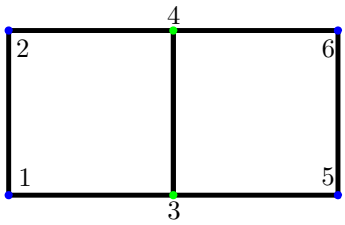


Figure 5.4 – Global element numbering of 0-forms.

Here n indicates the element number. Assembly of the global mass matrix then yields:

$$K^{(n)} = \begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} & k_{13}^{(1)} & k_{14}^{(1)} & & \\ k_{21}^{(1)} & k_{22}^{(1)} & k_{23}^{(1)} & k_{24}^{(1)} & & \\ k_{31}^{(1)} & k_{32}^{(1)} & (k_{33}^{(1)} + k_{11}^{(2)}) & (k_{34}^{(1)} + k_{12}^{(2)}) & k_{13}^{(2)} & k_{14}^{(2)} \\ k_{41}^{(1)} & k_{42}^{(1)} & (k_{43}^{(1)} + k_{21}^{(2)}) & (k_{44}^{(1)} + k_{22}^{(2)}) & k_{23}^{(2)} & k_{24}^{(2)} \\ & & k_{31}^{(2)} & k_{32}^{(2)} & k_{33}^{(2)} & k_{34}^{(2)} \\ & & k_{41}^{(2)} & k_{42}^{(2)} & k_{43}^{(2)} & k_{44}^{(2)} \end{bmatrix}. \tag{5.12}$$

In the next section the Poisson problem for 2-forms is solved. There it will be shown that the connection between elements is not made at the nodes but at the edges they share.

5.3 2-forms

In this section the Poisson problem for 2-forms is discussed. Repeating once more the expression for the Hodge Laplacian:

$$-\Delta = \mathrm{d}^\star \mathrm{d} + \mathrm{d}\mathrm{d}^\star.$$

And the (dual) De Rham complex in 2D:

$$\begin{array}{ccccccccc} \mathbb{R} & \longrightarrow & \Lambda^0(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^1(\Omega) & \xrightarrow{\mathrm{d}} & \Lambda^2(\Omega) & \xrightarrow{\mathrm{d}} & 0 \\ & & \star\updownarrow & & \star\updownarrow & & \star\updownarrow & & \\ 0 & \xleftarrow{\mathrm{d}} & \Lambda^2(\Omega) & \xleftarrow{\mathrm{d}} & \Lambda^1(\Omega) & \xleftarrow{\mathrm{d}} & \Lambda^0(\Omega) & \longleftarrow & \mathbb{R} \end{array}.$$

Hence, the differential form of the Poisson problem for 2-forms reduces to:

$$\mathrm{d}\mathrm{d}^\star u^{(2)} = f^{(2)} \tag{5.13}$$

This needs to be written as a set of two first-order equations:

$$\mathrm{d}q^{(1)} = f^{(2)} \tag{5.14}$$

$$\mathrm{d}^\star u^{(2)} = q^{(1)} \tag{5.15}$$

The variational formulation requires finding $(q^{(1)}, u^{(2)}) \in \Lambda^1(\Omega) \times \Lambda^2(\Omega)$ given $f^{(2)} \in \Lambda^{(2)}(\Omega)$ for all possible test functions $(v^{(0)}, \tau^{(1)}) \in \Lambda^0(\Omega) \times \Lambda^1(\Omega)$ subject to the boundary condition $u = 0$ on $\partial\Omega$. In mathematical notation:

$$\left(v^{(2)}, \mathrm{d}q^{(1)}\right)_\Omega = \left(v^{(2)}, f^{(2)}\right)_\Omega, \tag{5.16}$$

$$\left(\tau^{(1)}, \mathrm{d}^\star u^{(2)}\right)_\Omega = \left(\tau^{(1)}, q^{(1)}\right)_\Omega. \tag{5.17}$$

Rewriting the codifferential requires once more the application of equation (2.58) (integration by parts):

$$\left(\tau^{(1)}, \mathrm{d}^* u^{(2)}\right)_\Omega = \left(\mathrm{d}\tau^{(1)}, u^{(2)}\right)_\Omega - \int_{\partial\Omega} \operatorname{tr} \tau^{(1)} \wedge \star u^{(2)}. \tag{5.18}$$

Since a homogeneous Dirichlet boundary condition is enforced the boundary integral also cancels here.

Unlike the Poisson problem for 0-forms, equations (5.16) and (5.17) can not be condensed to a single equation. Therefore, the system to be solved is simply given by:

$$\left(v^{(2)}, \mathrm{d}q^{(1)}\right)_\Omega = \left(v^{(2)}, f^{(2)}\right)_\Omega, \tag{5.19}$$

$$\left(\mathrm{d}\tau^{(1)}, u^{(2)}\right)_\Omega = \left(\tau^{(1)}, q^{(1)}\right)_\Omega. \tag{5.20}$$

The left-hand and right-hand side inner-products of equation (5.19) can be written as (using the inner product definition of equation (2.39)):

$$\left(v^{(2)}, \mathrm{d}q^{(1)}\right)_\Omega = \int_\Omega v^{(2)} \wedge \star q^{(1)}, \tag{5.21}$$

$$\left(v^{(2)}, f^{(2)}\right)_\Omega = \int_\Omega v^{(2)} \wedge \star f^{(2)}. \tag{5.22}$$

Similarly, the inner-products of equation (5.20) are expanded as:

$$\left(\mathrm{d}\tau^{(1)}, u^{(2)}\right)_\Omega = \int_\Omega \mathrm{d}\tau^{(1)} \wedge \star u^{(2)}, \tag{5.23}$$

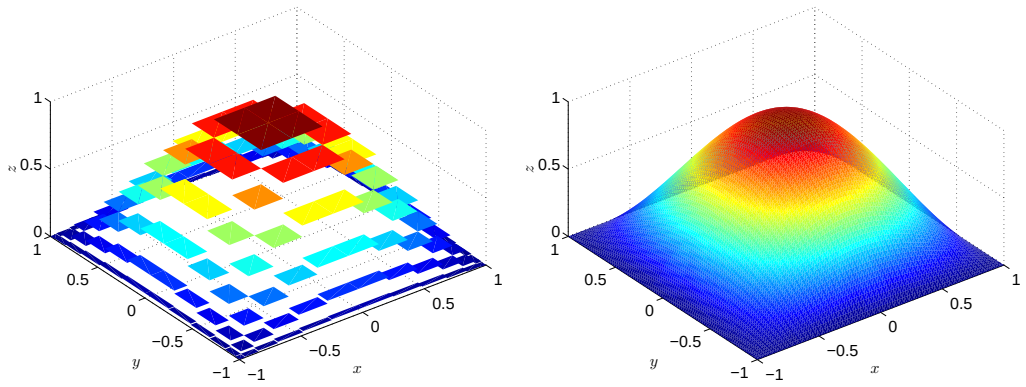
$$\left(\tau^{(1)}, q^{(1)}\right)_\Omega = \int_\Omega \tau^{(1)} \wedge \star q^{(1)}. \tag{5.24}$$

After a rigorous derivation (as shown in Appendix A.2). The end result is the matrix system given by equation (5.25):

$$\begin{bmatrix} -\bar{M}_1 & \mathbb{D}^T M_2 \\ \mathbb{D} M_2 & \varnothing \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \varnothing \\ M_2 \mathbf{f} \end{bmatrix}. \tag{5.25}$$

This system can be solved the 2-form $u^{(2)}$ (as well as the 1-form $q^{(1)}$). A cochain representation and its reconstruction of a 2-form Poisson problem on the standard element with order 12 is depicted in Figure 5.5.

The extension to multiple elements is slightly different in the case of the 2-form Poisson problem. Again this requires both a global numbering and local numbering. This time however the surfaces and edges are numbered. For example, consider the two elements depicted in Figure 5.6. Once again, for each element the system given by equation (5.25) can be constructed. Using the global numbering, each of these matrices is then *added* to the global system matrix. Instead of the nodes (or 0-forms), the elements now share edges (or 1-forms). Therefore, whenever edges are shared between neighboring elements, the contributions from both element matrices are added. The resulting global structure is depicted in Figure 5.7 In the end, the matrix operations are very much similar to what has been shown for the 0-form Poisson problem. However, the mixed nature of the 2-form Poisson problem with its two variables $u^{(2)}$ and $q^{(1)}$ causes the matrix structure to be less so.



(a) Cochain representation of the solution. **(b)** Reconstruction of the cochains.

Figure 5.5 – Solution of the 2-form Poisson problem on the standard element with order 12.

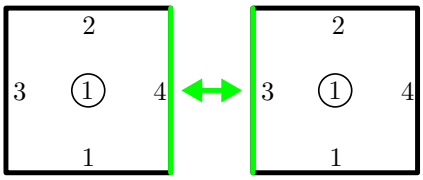


Figure 5.6 – Local element numbering of 1- and 2-forms. The encircled numbers refer to the 2-forms.

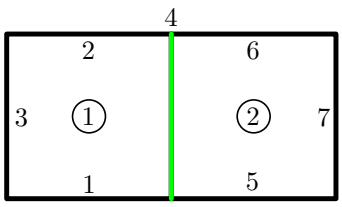


Figure 5.7 – Global element numbering 1- and 2-forms.

CHAPTER 6

MIMETIC MORTAR THEORY

In this chapter a mortar implementation for the MSEM is presented. Note that much of this theory is based on previous work by Maday et. al. and Anagnostou et. al. [23, 1]. Consequently, there will be similarities. The theory of mortar elements within the mimetic framework is explained in Section 6.1 including associated nomenclature. Having introduced the theory, it is put into practice. Therefore, Sections 6.2 and 6.3 present the discretizations required for p- and h-refinement respectively. This will include qualitative and quantitative examples of a simple test case for both 0-forms and 2-forms. Detailed results are deferred to Chapter 7.

6.1 Mimetic mortar element theory

Before introducing the mimetic mortar element theory, consider once more Figure 6.1 (taken from the introduction chapter). This non-conforming mesh is the result of a refinement step. It

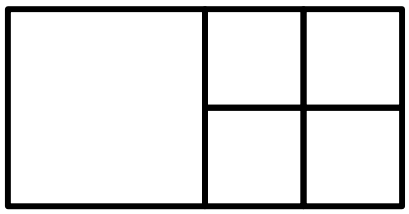
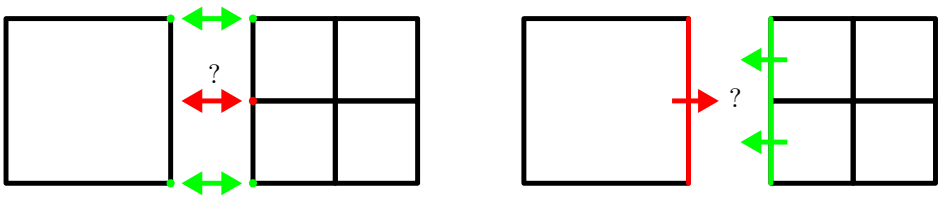


Figure 6.1 – A non-conforming mesh.

is not yet specified what type of refinement and it can therefore either be h- or p-refinement. That distinction is not yet relevant, although it will be when the actual discretization is concerned. However, of greater importance to the discretization process is the degree of the differential forms involved in the problem at hand. As mentioned in the introduction, this thesis restricts itself to the scalar Laplacian in $\mathbb{R}^2$. Hence, this implies solving the Poisson problem for either 0-forms or 2-forms (the solution to which has been shown in the previous chapter, on a conforming mesh).

Inevitably, the mixed formulation of the 2-form Poisson problem introduces 1-forms as well. As it turns out, these 1-forms are essential in the connection between elements. However, non-conforming meshes result in a discrepancy as depicted in Figure 6.2. Although in the introduction



(a) Connecting elements using 0-forms. **(b)** Connecting elements using 1-forms.

Figure 6.2 – Problems with non-conforming meshes.

this problem was presented as matching nodal temperatures or energy and mass fluxes, the analogy to 0-forms and 1-forms seems appropriate. To solve this problem, a mortar element approach is presented for the MSEM.

Some nomenclature must first be introduced. Consider a domain Ω consisting of N elements Ω^i. In algebraic topology this is denoted as a cell complex K, where each element is considered as a smaller (sub-)complex. Elements can therefore be described in terms of p-chains, with the cell complex K to be considered the union of element p-chains:

$$c_{(p)} = \bigcup_{i=1}^{N} c^i_{(p)}. \tag{6.1}$$

Here each $c^i_{(p)}$ denotes a p-chain corresponding to an element i. Note that the topological structure of each element in $\mathbb{R}^2$ requires three chains to be completely defined: a 0-chain for the nodes, a 1-chain for the edges and a 2-chain for the surfaces. The boundary of each element is the following 1-chain:

$$\partial c^i_{(2)} = \bar{\Gamma}^i_{(1)}. \tag{6.2}$$

Given that in this work rectangular elements are used, the boundary of each element can be seen as a summation of four smaller 1-chains:

$$\bar{\Gamma}^i_{(1)} = \sum_{j=1}^{4} \Gamma^{i,j}_{(1)}. \tag{6.3}$$

The shorthand notation for $\Gamma^{i,j}_{(1)}$ is simply Γ (possibly with the superscript i to indicate the element this boundary belongs to). Furthermore, the skeleton S of the mesh is defined as:

$$S = \bigcup_{i=1}^{N} \partial c^i_{(2)} = \bigcup_{i=1}^{N} \Gamma^i_{(1)}. \tag{6.4}$$

Using a simple mesh consisting of two linear elements the new nomenclature is depicted in Figure 6.3. Unlike the mesh depicted in Figure 6.3, most of them are not nicely conformal. Therefore

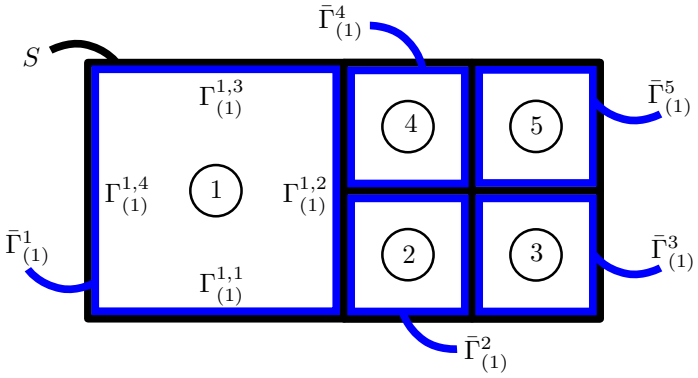


Figure 6.3 – Nomenclature of the mortar element method. Encircled numbers denote elements.

the mortar elements are introduced. Mortar elements are considered one-dimensional geometric objects, represented as the 1-chain $\gamma_{(1)}$ (or shorthand notation γ). It is assumed that each mortar γ corresponds to an entire edge of an element. An example is depicted in Figure 6.4. Associated with each mortar element is the mortar solution $\phi^{(n)}$. It is determined on the basis

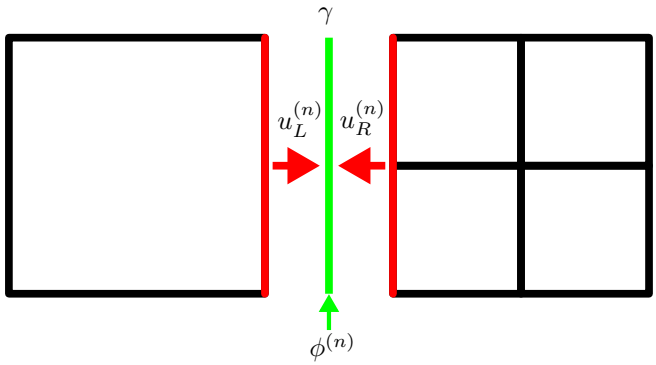


Figure 6.4 – The mortar element connection.

of the element solutions left and right from the mortar element. The degree n of this differential form depends on the problem under consideration. In the previous chapter it has been shown that in the mimetic framework the scalar Laplace can be evaluated for both 0-forms and 2-forms. Connecting element occurs differently in both cases. For the 0-form Poisson problem the connection is made at the nodes or 0-cells while for the 2-form Poisson problem the connection is made at the edges or 1-cells. Therefore, in the current context the mortar solution can either be a 0-form $\phi^{(0)}$ or a 1-form $\phi^{(1)}$.

In conventional MEM theory the mortar solution is defined through an integral matching condition. With the help of the inner product definition given by equation (2.39) a similar condition can be formulated within the framework of the MSEM:

$$((u^{(n)}|_\Gamma - \phi^{(n)}), \psi^{(n)}) = \int_\gamma (u^{(n)}|_\Gamma - \phi^{(n)}) \wedge \star\psi^{(n)} = 0. \tag{6.5}$$

Here $u^{(n)}|_\Gamma$ represents the solution restricted to some element boundary Γ (shorthand notation

used), $\phi^{(n)}$ is the mortar solution and $\psi^{(n)}$ is the test function (which corresponds to the element to which $u^{(n)}|_\Gamma$ belongs). In words, the integral matching condition requires the difference between the element solution and mortar solution to be zero.

The order of the reconstructions associated with these differential forms is not arbitrary. First of all, the solution $u^{(n)}|_\Gamma$ naturally has the same order as the element it belongs to. Since the test function $\psi^{(n)}$ corresponds to that element as well, it is of the same order. Finally, the order of the mortar solution $\phi^{(n)}$ is always equal to the lowest of the adjoining elements. Later sections will show why this has to be the case. Note that from here on the order of a differential form refers to the order of its reconstruction.

This covers the basic theory as well as nomenclature. The next step is the discretization itself. Section 6.2 starts with p-refinement, split up in a separate treatment for the 0-form and 2-form Poisson problem. Section 6.3 follows after that with h-refinement, with the same divide between the two types of problems.

6.2 p-refinement

With p-refinement the element order is increased or decreased locally. The need for doing so could for example arise when a certain part of the flow must be resolved more accurately (provided the underlying exact solution is smooth). In this section the theory of a mortar approach for 0-forms and 2-forms will be detailed, starting with the former.

6.2.1 0-forms

Consider the Poisson problem for 0-forms:

$$\begin{aligned} \mathrm{d}^\star \mathrm{d} u^{(0)} &= f^{(0)} \text{ on } \Omega, \\ \text{tr } u^{(0)} &= 0 \quad \text{ on } \partial\Omega. \end{aligned} \tag{6.6}$$

This must be solved on the mesh depicted in Figure 6.5. This mesh consists of two elements (of arbitrarily chosen order) which are connected by the mortar γ. The arrows indicate the solutions associated with these line segments. With the individual elements dimensioned as $\Omega_1 : x, y \in (-2, 0) \times (-1, 1)$ and $\Omega_2 : x, y \in (0, 2) \times (-1, 1)$, the combined domain Ω is simply the union of the two: $\Omega = \Omega_1 \cup \Omega_2$. A mortar γ connects elements Ω_1 and Ω_2. These are of order N_1 and N_2 respectively, with $N_1 \leq N_2$. Their corresponding solutions are of the same order. Hence, for both:

$$u^{(0)}|_{\Gamma^1} \in \mathbb{P}_{N_1} \text{ and } u^{(0)}|_{\Gamma^2} \in \mathbb{P}_{N_2}. \tag{6.7}$$

For each element the mass matrix can be determined as detailed in Chapter 5. This will not be repeated here. The important step is now to glue these two elements together using the integral matching condition as given by equation (6.5). For the problem at hand, in which 0-forms are the only differential forms, the integral matching condition is written as:

$$((u^{(0)}|_\Gamma - \phi^{(0)}), \psi^{(0)}) = \int_\gamma (u^{(0)}|_\Gamma - \phi^{(0)}) \wedge \star\psi^{(0)} = 0. \tag{6.8}$$

This condition must be imposed on both elements. The order of $\phi^{(0)}$ is defined to be equal to the lowest of the adjoining elements (i.e. $\phi^{(0)} \in \mathbb{P}_{N_1}$ in this case). The order of $\psi^{(0)}$ is however dependent on the element it is associated with. As will become clear further on in the

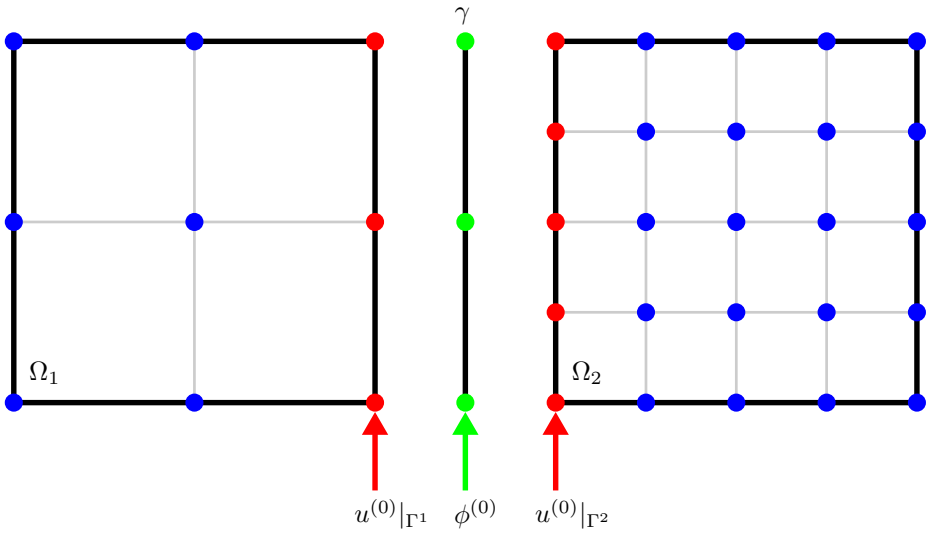


Figure 6.5 – p-Refined mesh for the 0-form Poisson problem.

discretization process, this is required to obtain a square and therefore invertible matrix. The aim is now to find an expression which relates $u^{(0)}$ on Γ to $\phi^{(0)}$:

$$\mathbf{u}|_\Gamma = Z\boldsymbol{\phi}. \tag{6.9}$$

Here Z represents a projection matrix which has yet to be determined and $\mathbf{u}|_\Gamma$ and $\boldsymbol{\phi}$ are vectors. Fortunately, for the left element (specifically because it is the *low* order element in this example) this matrix will be rather trivial, provided all functions are expanded in the same basis. To show this, first the integral matching condition is written down explicitly for element Ω_1. From this moment on, the degree of the differential forms (as indicated by the superscript (n)) is removed and replaced with the order of the function. The integral matching condition for element Ω_1 then becomes:

$$\int_{-1}^{1} \left(u^{N_1}|_\Gamma - \phi^{N_1}\right) \psi^{N_1}|_\Gamma \mathrm{d}s = 0. \tag{6.10}$$

The superscripts reveals that $u^{N_1}|_\Gamma$ and ϕ^{N_1} are both in $\mathbb{P}_{N_1}$ which eliminates the need for a variational formulation. As a result Z will turn out to simply be the identity matrix. This can easily be shown as follows. Start by rewriting the integral matching condition:

$$\int_{-1}^{1} u^{N_1}|_\Gamma \psi^{N_1}|_\Gamma \mathrm{d}s = \int_{-1}^{1} \phi^{N_1} \psi^{N_1}|_\Gamma \mathrm{d}s. \tag{6.11}$$

Next expand using Lagrange polynomials:

$$u^{N_1}|_\Gamma = \sum_{i=0}^{N_1} u^{N_1}(\xi_{N_1}, \eta_i) h_i^{N_1}(s) = \sum_{i=0}^{N_1} \hat{u}_i h_i^{N_1}(s), \tag{6.12}$$

$$\psi^{N_1}|_\Gamma = \sum_{i=0}^{N_1} \psi_i h_i^{N_1}(s), \tag{6.13}$$

$$\phi^{N_1} = \sum_{i=0}^{N_1} \phi_i h_i^{N_1}(s). \tag{6.14}$$

The Lagrangian basis is defined as:

$$h_i^{N_1} \in \mathbb{P}_{N_1}([-1,1]),\ h_i(\xi_j) = \delta_{ij},\ \forall i,j \in \{0,\dots,N_1\}^2. \tag{6.15}$$

Substitution of these expressions into the integral matching condition and performing quadrature in the (N_1+1) GLL nodes yields (for details on quadrature see Appendix A):

$$\begin{aligned}\int_{-1}^{1} u^{N_1}|_\Gamma \psi^{N_1}|_\Gamma \mathrm{d}s &= \sum_{i=0}^{N_1} \psi_i \sum_{j=0}^{N_1} \hat{u}_j \sum_{q=0}^{N_1} w_q h_i^{N_1}(s_q) h_j^{N_1}(s_q) \\ &= \sum_{q=0}^{N_1} w_q \psi_q \hat{u}_q, \end{aligned} \tag{6.16}$$

$$\begin{aligned}\int_{-1}^{1} \phi^{N_1} \psi^{N_1}|_\Gamma \mathrm{d}s &= \sum_{i=0}^{N_1} \psi_i \sum_{j=0}^{N_1} \phi_j \sum_{q=0}^{N_1} w_q h_i^{N_1}(s_q) h_j^{N_1}(s_q) \\ &= \sum_{q=0}^{N_1} w_q \psi_q \phi_q. \end{aligned} \tag{6.17}$$

Since the vertical dimension of the domain has been conveniently chosen to correspond to the standard domain of $(-1,1)$ no mapping terms arise (evidently this does not always have to be the case but is done here to simplify the arithmetic).

The discretized equation can now be cast in matrix-vector form:

$$[\psi^{N_1}]^T [W^{N_1 \times N_1}][\hat{u}^{N_1}] = [\psi^{N_1}]^T [W^{N_1 \times N_1}][\phi^{N_1}]. \tag{6.18}$$

Or similarly:

$$\boldsymbol{\psi}^T W \hat{\mathbf{u}} = \boldsymbol{\psi}^T W \boldsymbol{\phi}. \tag{6.19}$$

Which eventually reduces to:

$$\hat{\mathbf{u}} = \underbrace{I}_{\hat{Z}} \boldsymbol{\phi}. \tag{6.20}$$

With I representing the $N_1 \times N_1$ identity matrix.

For the right element the projection matrix will not be that simple. Starting again with the integral formulation:

$$\int_{-1}^{1} \left(u^{N_2}|_\Gamma - \phi^{N_1}\right) \psi^{N_2}|_\Gamma \mathrm{d}s = 0. \tag{6.21}$$

Or equivalently:

$$\int_{-1}^{1} u^{N_2}|_\Gamma \psi^{N_2}|_\Gamma \mathrm{d}s = \int_{-1}^{1} \phi^{N_1} \psi^{N_2}|_\Gamma \mathrm{d}s. \tag{6.22}$$

The next step is again the discretization:

$$u^{N_2}|_\Gamma = \sum_{i=0}^{N_2} u^{N_2}(\xi_0, \eta_i) h_i^{N_2}(s) = \sum_{i=0}^{N_2} \check{u}_i h_i^{N_2}(s), \tag{6.23}$$

$$\psi^{N_2}|_\Gamma = \sum_{i=0}^{N_2} \psi_i h_i^{N_2}(s), \tag{6.24}$$

$$\phi^{N_1} = \sum_{i=0}^{N_1} \phi_i h_i^{N_1}(s). \tag{6.25}$$

With the second Lagrangian basis given as:

$$h_i^{N_2} \in \mathbb{P}_{N_1}([-1,1]),\ h_i\left(\xi_j\right) = \delta_{ij},\ \forall i,j \in \{0,\ldots,N_2\}^2. \tag{6.26}$$

Substitution and integration in the (N_2+1) GLL nodes yields:

$$\begin{aligned}\int_{-1}^{1} u^{N_2}|_\Gamma \psi^{N_2}|_\Gamma \mathrm{d}s &= \sum_{i=0}^{N_2} \psi_i \sum_{j=0}^{N_2} \check{u}_j \sum_{q=0}^{N_2} w_q h_i^{N_2}(s_q) h_j^{N_2}(s_q) \\ &= \sum_{q=0}^{N_2} w_q \psi_q \check{u}_q, \end{aligned} \tag{6.27}$$

$$\begin{aligned}\int_{-1}^{1} \phi^{N_1} \psi^{N_2}|_\Gamma \mathrm{d}s &= \sum_{i=0}^{N_2} \psi_i \sum_{j=0}^{N_1} \phi_j \sum_{q=0}^{N_2} w_q h_i^{N_2}(s_q) h_j^{N_1}(s_q) \\ &= \sum_{j=0}^{N_1} \phi_j \sum_{q=0}^{N_2} w_q \psi_q \phi_q. \end{aligned} \tag{6.28}$$

In matrix-vector format this becomes:

$$[\psi^{N_2}]^T [W^{N_2 \times N_2}][\check{u}^{N_2}] = [\psi^{N_2}]^T [\tilde{W}^{N_2 \times N_1}][\phi^{N_1}]. \tag{6.29}$$

Or similarly:

$$\boldsymbol{\psi}^T W \check{\mathbf{u}} = \boldsymbol{\psi}^T \tilde{W} \boldsymbol{\phi}. \tag{6.30}$$

Important to note here is that W is of dimensions $N_2 \times N_2$ (and thus square) while $\tilde{W}$ has dimensions $N_2 \times N_1$ (which is clearly not square). Again canceling the test function $\boldsymbol{\psi}$ (it should hold for all possible $\boldsymbol{\psi}$) and left-multiplying both sides by $\tilde{W}^{-1}$ yields:

$$\check{\mathbf{u}} = \underbrace{W^{-1}\tilde{W}}_{\check{Z}} \boldsymbol{\phi}. \tag{6.31}$$

Since taking the inverse of W requires it to be square, the order of the mortar solution $\phi^{(0)}$ had to be chosen equal to the order of the *lowest* order element while the order of test function $\psi^{(0)}$ should always be equal to the element currently under consideration.

Right now a relation has been established for both elements between the solution $u^{(0)}$ on the boundary Γ and the mortar solution $\phi^{(0)}$ in the form of the projection matrix Z. From this a *global* projection matrix can be obtained which also includes the nodes not located on the inner element boundary. These nodes will from here on be referred to as interior nodes and indicated with a subscript i. For element Ω_1 this global projection is just the identity matrix as shown. For element Ω_2 however the following matrix results:

$$\underbrace{\begin{bmatrix}\mathbf{u}_i\\ \check{\mathbf{u}}\end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix}I & 0\\ 0 & \check{Z}\end{bmatrix}}_{Z}\underbrace{\begin{bmatrix}\mathbf{u}_i\\ \boldsymbol{\phi}\end{bmatrix}}_{\tilde{\mathbf{u}}}. \tag{6.32}$$

In simplified form this yields:

$$\mathbf{u} = Z\tilde{\mathbf{u}}. \tag{6.33}$$

The global projection matrix Z can be expanded to include the projection matrices of other elements as well (in this case Ω_1). Now consider the Poisson problem from Chapter 5 of which the resulting matrix system is of the form $A\mathbf{u} = \mathbf{f}$ with A the Laplacian block matrix. Substitute equation (6.33) for $\boldsymbol{u}$ into this expression yields:

$$AZ\tilde{\mathbf{u}} = \mathbf{f}. \tag{6.34}$$

To obtain a symmetric matrix again premultiply by Z^T:

$$Z^T AZ\tilde{\mathbf{u}} = Z^T\mathbf{f}. \tag{6.35}$$

This can be solved for $\tilde{\mathbf{u}}$ and with the help of equation (6.33) converted back to $\mathbf{u}$.

Example test case for 0-forms

At this point it might be worthwhile to qualitatively (and quantitatively) investigate a simple test case as a visual aid to the theory just presented. To that end consider the Poisson problem for 0-forms subject to homogeneous boundary conditions as given by equation (6.6) and repeated here for convenience:

$$\begin{aligned}&\mathrm{d}^\star \mathrm{d}u^{(0)} = f^{(0)} \text{ on } \Omega,\\ &\text{tr } u^{(0)} = 0 \quad \text{ on } \partial\Omega.\end{aligned}$$

The domain $\Omega : x, y \in (0,2)\times(0,1)$ consists of two elements:

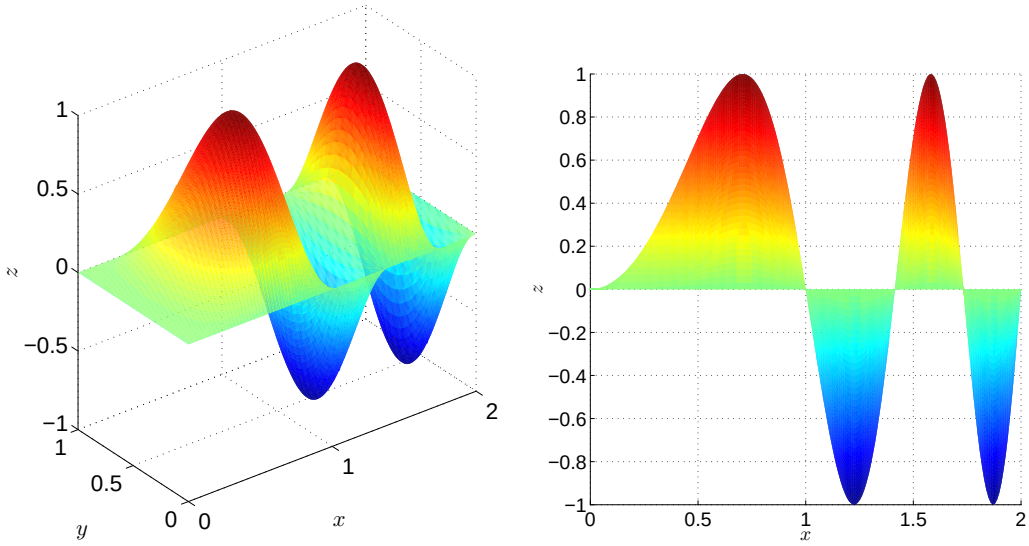
$$\Omega_1 : x, y \in (0,1)\times(0,1), \tag{6.36}$$
$$\Omega_2 : x, y \in (1,2)\times(0,1). \tag{6.37}$$

The exact solution $u^{(0)}$ is chosen to be:

$$u^{(0)} = \sin\left(\pi x^2\right)\sin\left(\pi y\right). \tag{6.38}$$

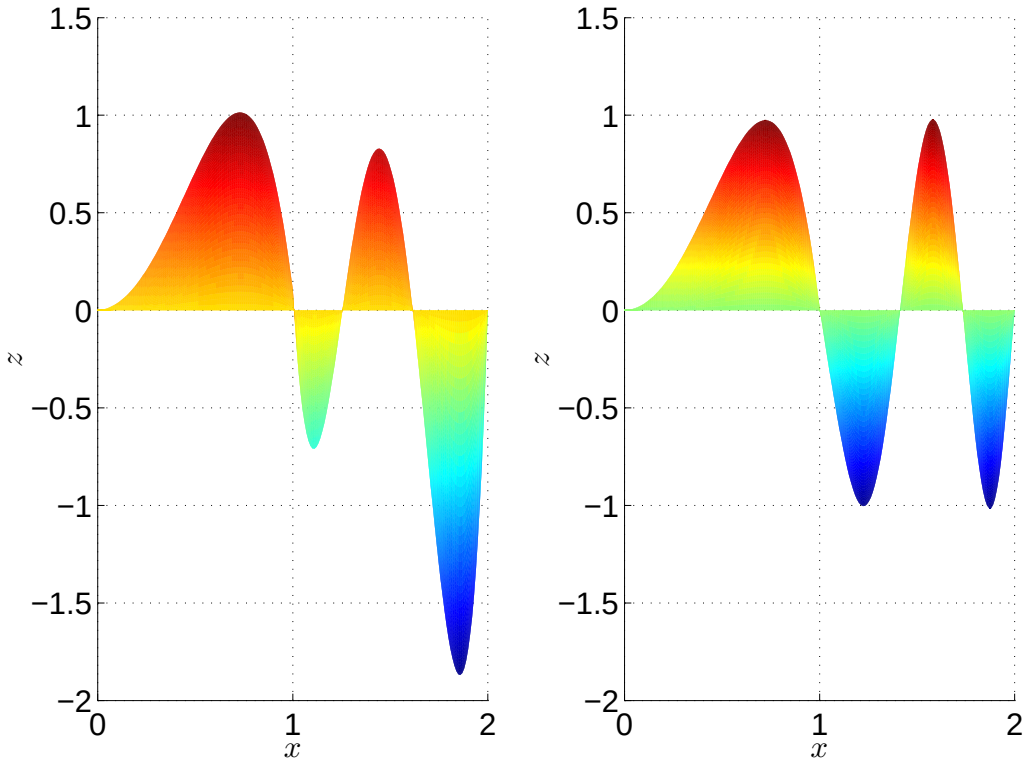
With associated right-hand side function $f^{(0)}$:

$$f^{(0)} = \pi\sin(\pi y)\left(2\cos(\pi x^2) - \pi(4x^2+1)\sin(\pi x^2)\right). \tag{6.39}$$



(a) Isometric view. **(b)** Side view.

Figure 6.6 – Graphical representation of equation (6.38).



(a) Uniform, $p = 4$. **(b)** Refined, $p = 4$ (l) and $p = 8$ (r).

Figure 6.7 – Reconstruction of the solution to the 0-form Poisson problem on a uniform and refined mesh.

The exact solution is visualized in Figure 6.6. From this it can readily be concluded that the right element (containing three extrema as opposed to one) would benefit from an increase in order. To clearly visualize this, first consider the solution when both elements are 4^{th} order as depicted in Figure 6.7a. Quick inspection of the results reveal that the order of element Ω_2 is insufficient to represent the solution accurately. The extrema should all be of magnitude 1 which none of them are. Viewing from left to right, the first two never reach their intended value while the last one exceeds it by a significant margin. This is reflected in the $L^2\Lambda^0(= ||u - u_h||_{L^2\Lambda^0})$ error for each element: $L^2\Lambda^0|_{\Omega_1} = 3.973 \times 10^{-2}$ and $L^2\Lambda^0|_{\Omega_2} = 5.597 \times 10^{-1}$. This problem has 45 unique degrees of freedom. For the purpose of this example it is proposed to double the order of element Ω_2. This result is depicted Figure 6.7b.

These results show that increasing the order of element Ω_2 by 4 results in a far more accurate solution. The magnitudes of the extrema of element Ω_2 are very close to 1 which result in a lower $L^2\Lambda^0$ for both elements: $L^2\Lambda^0|_{\Omega_1} = 1.938 \times 10^{-2}$ and $L^2\Lambda^0|_{\Omega_2} = 9.561 \times 10^{-3}$. This particular example reveals the interesting effect that even the error of element Ω_1 has reduced by almost a factor of 2. Hence, in the uniform case the large error of element Ω_2 has a significant influence on the error of element Ω_1. This effect must be taken into consideration when a certain distribution of element orders is imposed. The need arises to introduce gradual transitions between elements. Simply put, the accuracy of a solution on a mesh consisting of a linear element which is connected to an element of significantly higher order will by severely limited. Note that this particular test case has 97 degrees of freedom.

Finally, when both elements are of order 8, the errors become $L^2\Lambda^0|_{\Omega_1} = 1.075 \times 10^{-4}$ and $L^2\Lambda^0|_{\Omega_2} = 9.211 \times 10^{-3}$ respectively at the cost of 153 degrees of freedom. With respect to the refined case the error in element 1 is decreased by a factor of 100 while the error in element 2 remained almost the same. The number of degrees of freedom did however increase by more than 50% with respect to the refined case. It immediately becomes clear why an adaptive approach might be preferred in a situation like this. A summary of these results can be found in Table 6.1.

Table 6.1 – Results summary for the 0-form Poisson problem test case.

p	$L^2\Lambda^0\|_{\Omega_1}$	$L^2\Lambda^0\|_{\Omega_2}$	N_{DOF}	$\Delta N_{DOF}(\%)$
[4 4]	3.973×10^{-2}	5.597×10^{-1}	45	–
[4 8]	1.938×10^{-2}	9.561×10^{-3}	97	+116%
[8 8]	1.075×10^{-4}	9.211×10^{-3}	153	+240%

This covers p-refinement for 0-forms. In principle there are many similarities between this approach and the original MEM. In the next subsection however, which treats 2-forms, certain differences will arise.

6.2.2 2-forms

Consider the Poisson problem for 2-forms:

$$\begin{aligned} \mathrm{d}\mathrm{d}^\star u^{(2)} &= f^{(2)} \text{ on } \Omega, \\ \mathrm{tr}\ u^{(2)} &= 0 \quad \text{ on } \partial\Omega. \end{aligned} \tag{6.40}$$

Which could be rewritten as a set of first-order equations (see Chapter 5):

$$\mathrm{d}q^{(1)} = f^{(2)}, \tag{6.41}$$

$$\mathrm{d}^\star u^{(2)} = q^{(1)}. \tag{6.42}$$

Note the appearance of the 1-form $q^{(1)}$. This 1-form will be shown to be instrumental in solving the Poisson problem for 2-forms with the mortar element approach. Consider once more the previous mesh as depicted here again in Figure 6.8 with adjusted notation. It is dimensioned

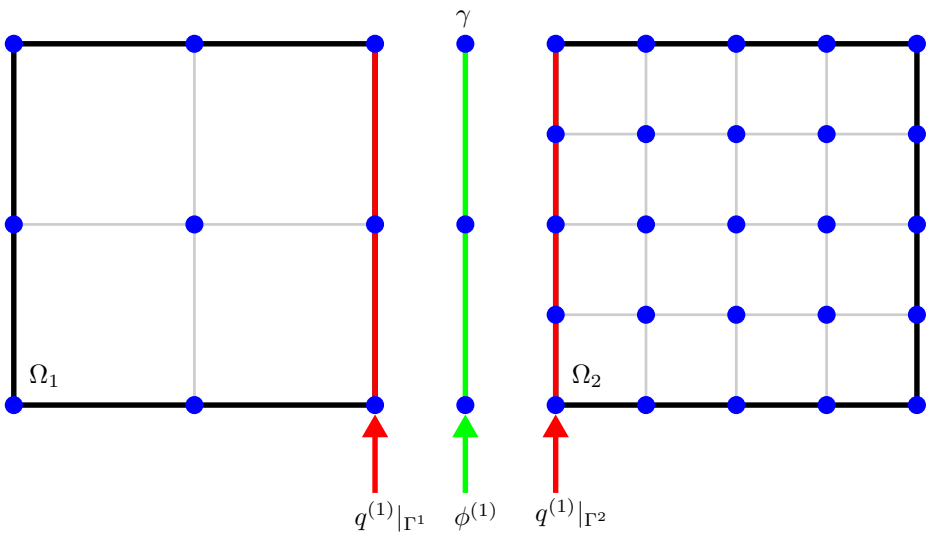


Figure 6.8 – p-Refined mesh for the 2-form Poisson problem.

exactly the same. And as before, the mortar γ connects elements Ω_1 and Ω_2, which are of order N_1 and N_2 respectively (with $N_1 \leq N_2$). This implies:

$$q^{(1)}|_{\Gamma^1} \in \mathbb{P}_{N_1} \text{ and } q^{(1)}|_{\Gamma^2} \in \mathbb{P}_{N_2}. \tag{6.43}$$

Once again, the mass matrices of each element can be determined as explained in Chapter 5. The challenge is now to connect these.

Different from the Poisson problem for 0-forms, at the inter element boundary there are now 1-forms. Adjusting the integral matching condition from equation (6.5) accordingly yields:

$$((q^{(1)}|_\Gamma - \phi^{(1)}), \psi^{(1)}) = \int_\gamma (q^{(1)}|_\Gamma - \phi^{(1)}) \wedge \star\psi^{(1)} = 0. \tag{6.44}$$

The mortar solution $\phi^{(1)}$ is now a 1-form. Its order is defined to be equal to the lowest of the adjoining elements (i.e. $\phi^{(1)} \in \mathbb{P}_{N_1}$) while $\psi^{(1)} \in \mathbb{P}_{N_1}$ for the left element and $\psi^{(1)} \in \mathbb{P}_{N_2}$ for the right element. The goal is now to find an expression which relates $q^{(1)}$ on Γ to $\phi^{(1)}$:

$$\mathbf{q}|_\Gamma = Z\boldsymbol{\phi}. \tag{6.45}$$

Once more a a projection matrix Z has to be determined. As seen previously, for the left element this matrix will be rather trivial. Provided the same basis is used everywhere, this will turn out to be the identity matrix. More interesting to see is how to cope with the element on the right.

Starting with the integral formulation (replacing once more the degree of the differential forms with the order):

$$\int_{-1}^{1} \left(q^{N_2}|_\Gamma - \phi^{N_1}\right) \psi^{N_2}|_\Gamma \mathrm{d}s = 0. \tag{6.46}$$

Or equivalently:

$$\int_{-1}^{1} q^{N_2}|_\Gamma \psi^{N_2}|_\Gamma \mathrm{d}s = \int_{-1}^{1} \phi^{N_1} \psi^{N_2}|_\Gamma \mathrm{d}s. \tag{6.47}$$

Moving on to the discretization:

$$q^{N_2}|_\Gamma = \sum_{i=1}^{N_2} q^{N_2}(\xi_0, \eta_i) e_i^{N_2}(s) = \sum_{i=1}^{N_2} \check{q}_i e_i^{N_2}(s), \tag{6.48}$$

$$\psi^{N_2}|_\Gamma = \sum_{i=1}^{N_2} \psi_i e_i^{N_2}(s), \tag{6.49}$$

$$\phi^{N_1} = \sum_{i=1}^{N_1} \phi_i e_i^{N_1}(s). \tag{6.50}$$

With the basis of edge functions given as:

$$e_i^{N_2} \in \mathbb{P}_{N_1}([-1,1]),\ \int_{\xi_{j-1}}^{\xi_j} e_i\left(\xi_j\right) = \delta_{ij},\ \forall i,j \in \{1,\ldots,N_2\}^2. \tag{6.51}$$

Substitution and integration in the (N_2+1) GLL nodes yields:

$$\int_{-1}^{1} q^{N_2}|_\Gamma \psi^{N_2}|_\Gamma \mathrm{d}s = \sum_{i=1}^{N_2} \psi_i \sum_{j=1}^{N_2} \check{q}_j \sum_{p=0}^{N_2} w_p e_i^{N_2}(s_p) e_j^{N_2}(s_p), \tag{6.52}$$

$$\int_{-1}^{1} \phi^{N_1} \psi^{N_2}|_\Gamma \mathrm{d}s = \sum_{i=1}^{N_2} \psi_i \sum_{j=1}^{N_1} \phi_j \sum_{p=0}^{N_2} w_q e_i^{N_2}(s_p) e_j^{N_1}(s_p). \tag{6.53}$$

In matrix-vector format this becomes:

$$[\psi^{N_2}]^T [W^{N_2 \times N_2}][\check{q}^{N_2}] = [\psi^{N_2}]^T [\tilde{W}^{N_2 \times N_1}][\phi^{N_1}]. \tag{6.54}$$

Or similarly:

$$\boldsymbol{\psi}^T W \check{\boldsymbol{q}} = \boldsymbol{\psi}^T \tilde{W} \boldsymbol{\phi}. \tag{6.55}$$

Once again, note that W is of dimensions $N_2 \times N_2$ (and thus square and invertible) while $\tilde{W}$ has dimensions $N_2 \times N_1$ (which is clearly not square and invertible). Canceling the test function $\boldsymbol{\psi}$ and left-multiplying both sides by $\tilde{W}^{-1}$ yields:

$$\check{\mathbf{q}} = \underbrace{W^{-1}\tilde{W}}_{\tilde{Z}} \boldsymbol{\phi}. \tag{6.56}$$

Finally a global projection matrix can be obtained for the 1-forms $q^{(1)}$ which for element Ω_1 is just the identity matrix. For element Ω_2 however the following matrix results:

$$\underbrace{\begin{bmatrix}\mathbf{q}_i\\ \check{\mathbf{q}}\end{bmatrix}}_{\mathbf{q}} = \underbrace{\begin{bmatrix}I & 0\\ 0 & \check{Z}\end{bmatrix}}_{\mathring{Z}} \underbrace{\begin{bmatrix}\mathbf{q}_i\\ \boldsymbol{\phi}\end{bmatrix}}_{\tilde{\mathbf{q}}}. \tag{6.57}$$

In simplified form this yields:

$$\mathbf{q} = \mathring{Z}\tilde{\mathbf{q}}. \tag{6.58}$$

Consider now the Poisson problem for 2-forms cast into the form:

$$A\begin{bmatrix}\mathbf{q}\\ \mathbf{u}\end{bmatrix} = \mathbf{f}. \tag{6.59}$$

Introduce the projection from equation (6.58) (where $\mathring{Z}$ now also includes the projection matrix of element Ω_1):

$$A\begin{bmatrix}I & 0\\ 0 & \mathring{Z}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{q}}\\ \mathbf{u}\end{bmatrix} = AZ\begin{bmatrix}\tilde{\mathbf{q}}\\ \mathbf{u}\end{bmatrix} = \mathbf{f}. \tag{6.60}$$

To obtain a symmetric matrix again premultiply by Z^T:

$$Z^TAZ\begin{bmatrix}\tilde{\mathbf{q}}\\ \mathbf{u}\end{bmatrix} = Z^T\mathbf{f}. \tag{6.61}$$

This can be solved for both $\mathbf{u}$ and $\tilde{\mathbf{q}}$ (from which $\mathbf{q}$ can be obtained using equation (6.58)).

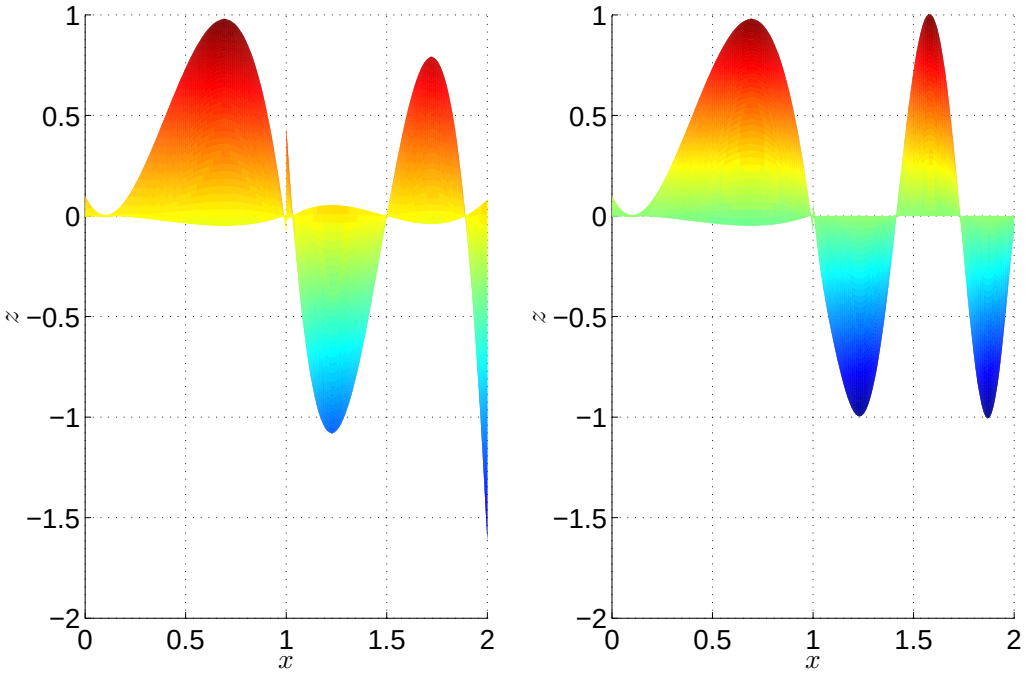
Example test case for 2-forms

For illustrative purposes another qualitative investigation is deemed useful to demonstrate the theory for 2-forms. Consider the Poisson problem for 2-forms subject to homogeneous boundary conditions as given by equation (6.40) and repeated here for convenience:

$$\begin{aligned}&\mathrm{d}\mathrm{d}^\star u^{(2)} = f^{(2)} \text{ on } \Omega,\\ &\text{tr } u^{(2)} = 0 \quad \text{ on } \partial\Omega.\end{aligned}$$

The remaining details are exactly equal to those of the previous example for 0-forms. The two test cases as well: first both elements are of 4^{th} order after which the order of the second element is doubled to 8. The results are depicted in Figure 6.9. Once more the order of element Ω_2 is insufficient to approximate the solution accurately. The extrema are again not reaching their correct values. For the error ($L^2\Lambda^2 = ||u-u_h||_{L^2\Lambda^2}$) it is found that $L^2\Lambda^2|_{\Omega_1} = 2.540\times10^{-2}$ and $L^2\Lambda^2|_{\Omega_2} = 4.573\times10^{-1}$. This particular mesh configuration has 108 degrees of freedom (which includes both the 2-forms $u^{(2)}$ and 1-forms $q^{(1)}$).

Doubling the order of element Ω_2 yields the result depicted in Figure 6.9b. This shows a significant improvement. The magnitudes of the extrema of element Ω_2 are very close to 1 which results in a lower error for both elements: $L^2\Lambda^2|_{\Omega_1} = 2.528\times10^{-2}$ and $L^2\Lambda^2|_{\Omega_2} = 1.010\times10^{-2}$. Compared to the 0-form case, this time the error in element Ω_1 is barely affected. From the results it can easily be observed that the solution in element Ω_1 near the boundary $x = 0$ leaves something to be desired. Nevertheless, the solution in element Ω_2 improved significantly as the



(a) Uniform, $p = 4$ **(b)** Refined, $p = 4$ (l) and $p = 8$ (r)

Figure 6.9 – Reconstruction of the solution to the 2-form Poisson problem on a uniform and refined mesh.

error has decreased by more than a factor of 100. The number of degrees of freedom has however increased to 256.

Finally, when both elements are of order 8, the errors become $L^2\Lambda^2|_{\Omega_1} = 1.485 \times 10^{-4}$ and $L^2\Lambda^2|_{\Omega_2} = 9.960 \times 10^{-3}$ respectively at the cost of 408 degrees of freedom. Evidently this significantly decreased the error in element Ω_1 while the error in element Ω_2 decreased only slightly. The number of degrees of freedom did however almost double with respect to the refined mesh, which is a very steep increase. A summary of these results can be found in Table 6.2.

Table 6.2 – Results summary for the 2-form Poisson problem test case.

p	$L^2\Lambda^2\|_{\Omega_1}$	$L^2\Lambda^2\|_{\Omega_2}$	N_{DOF}	$\Delta N_{DOF}(\%)$
[4 4]	2.540×10^{-2}	4.573×10^{-1}	108	–
[4 8]	2.528×10^{-2}	1.010×10^{-2}	256	+137%
[8 8]	1.485×10^{-4}	9.960×10^{-3}	408	+278%

This covers p-refinement for 2-forms. In the end, the usage of the inherent structure of the Poisson problem for 2-forms allowed it to be solved relatively straightforward. In the next section a similar approach to h-refinement (for both 0-forms and 2-forms) will be discussed.

6.3 h-refinement

In this section the focus of attention will be h-refinement. Even though there are many similarities between this type of refinement and p-refinement, the differences that do exist warrant a separate inspection. This section starts with a discussion of the 0-form case, which is then followed by an elaboration of the approach for 2-forms.

6.3.1 0-forms

This subsection begins by repeating parts of the discussion on p-refinement. Starting with the Poisson problem for 0-forms:

$$\begin{aligned} \mathrm{d}^\star \mathrm{d} u^{(0)} &= f^{(0)} \text{ on } \Omega, \\ u^{(0)} &= 0 \quad \text{ on } \partial\Omega. \end{aligned}$$

The h-refined domain on which this equation is to be solved is depicted in Figure 6.10. As can be seen, it consists of the three elements Ω_1, Ω_2 and Ω_3 (of arbitrarily chosen order) which, for the purpose of this example, are dimensioned as $\Omega_1 : x, y \in (-2, 0) \times (-1, 1), \Omega_2 : x, y \in (0, 1) \times (-1, 0)$ and $\Omega_3 : x, y \in (0, 1) \times (0, 1)$. A mortar γ is introduced which connects Ω_1 to Ω_2 and Ω_3. Hence, the shared node between elements Ω_2 and Ω_3 is covered by a single mortar. Indicated in the figure are also the inter element boundaries Γ^i. To simplify further arithmetic it is assumed that all elements are of equal order N:

$$u|_{\Gamma^1},\ u|_{\Gamma^2},\ u|_{\Gamma^3} \in \mathbb{P}_N. \tag{6.62}$$

Once again the integral matching condition must be imposed, in this case the 0-form variant. Since the mortar element connects three elements it also requires three mortar projections. These will be derived here. Repeating once more the integral matching condition:

$$((u^{(0)}|_\Gamma - \phi^{(0)}), \psi^{(0)}) = \int_\gamma (u^{(0)}|_\Gamma - \phi^{(0)}) \wedge \star\psi^{(0)} = 0.$$

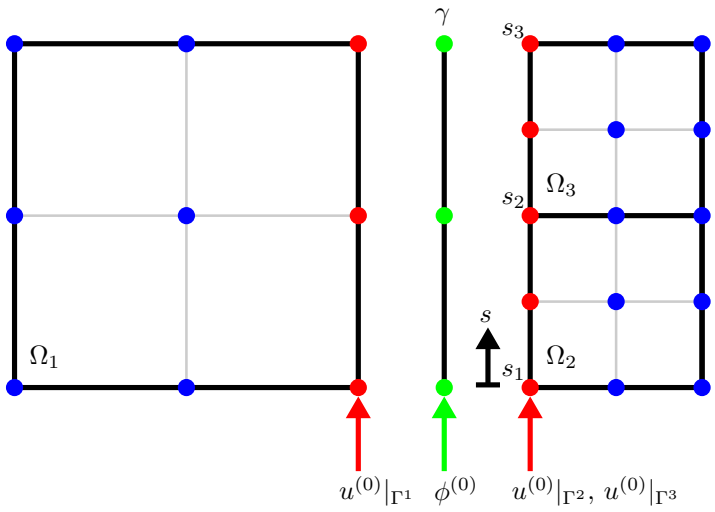


Figure 6.10 – h-Refined mesh for the 0-form Poisson problem.

Many familiar elements from the discussion on p-refinement return. These will not be elaborated upon any further. It suffices to say a mortar function $\phi^{(0)}$ is introduced (of order N) as well as a test function $\psi^{(0)}$ (also of order N). Again, for each element an expression is sought of the form:

$$\mathbf{u}|_\Gamma = Z\boldsymbol{\phi}$$

Starting with element Ω_1, its projection will be very straightforward. It has already been shown twice (both for 0- and 2-forms) that Z, given $\psi^{(0)}$, $\phi^{(0)}$ and $u^{(0)}$ being of equal order and expanded into the same basis, will turn out to be simply the identity matrix. These conditions are met for element Ω_1 and therefore no further derivations are given.

Next are the projection matrices of elements Ω_2 and Ω_3 which will be derived concurrently as they are largely similar. Starting with the integral matching condition (in slightly rewritten form; the integrals are split):

$$\int_{\Gamma^2} u_2|_{\Gamma^2}\psi|_{\Gamma^2}\mathrm{d}s = \int_{\Gamma^2} \phi\psi|_{\Gamma^2}\mathrm{d}s = 0, \tag{6.63}$$

$$\int_{\Gamma^3} u_3|_{\Gamma^3}\psi|_{\Gamma^3}\mathrm{d}s = \int_{\Gamma^3} \phi\psi|_{\Gamma^3}\mathrm{d}s = 0. \tag{6.64}$$

first of all, note the absence of the superscripts denoting either order of the reconstruction or degree of the differential form. This is done to enhance readability. Secondly, when combined the integrals over Γ^i constitute integration over the mortar γ. In order to evaluate these integrals a mapping to the standard element must be introduced. This is readily achieved with the use of the following equations (tailored to linear scaling):

$$s = \Phi(\xi) \quad = c_1\xi + c_2, \tag{6.65}$$

$$\xi = \Phi^{-1}(s) = \frac{s - c_2}{c_1}. \tag{6.66}$$

Here $\xi \in (-1, 1)$ while c_1 (the scaling factor) and c_2 (the coordinate shift) represent constants

specific to the mapping, both of which can be calculated as follows:

$$c_1 = \frac{s_e - s_s}{2}, \tag{6.67}$$

$$c_2 = s_s + c_1. \tag{6.68}$$

The variables s_s and s_e represent the *start* and *end* coordinates of the line segment (or boundary) under consideration. Applying the change of coordinates to the integral matching condition yields:

$$\frac{|\Gamma^2|}{2}\int_{-1}^{1} u_2|_{\Gamma^2}(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi = \frac{|\Gamma^2|}{2}\int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi, \tag{6.69}$$

$$\frac{|\Gamma^3|}{2}\int_{-1}^{1} u_3|_{\Gamma^3}(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi = \frac{|\Gamma^3|}{2}\int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi. \tag{6.70}$$

In this particular example, simply due to the fact that the vertical dimension of elements Ω_2 and Ω_3 is exactly half of that of element 1, the mapping terms $\frac{|\Gamma^2|}{2}$ and $\frac{|\Gamma^3|}{2}$ are equal to $\frac{|\Gamma^1|}{4}$. However, since this constant appears on both sides of the equation it can be canceled yielding:

$$\int_{-1}^{1} u_2|_{\Gamma^2}(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi = \int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi, \tag{6.71}$$

$$\int_{-1}^{1} u_3|_{\Gamma^3}(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi = \int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi. \tag{6.72}$$

Defining the integrals as:

$$I_{11} = \int_{-1}^{1} u_2|_{\Gamma^2}(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi, \tag{6.73}$$

$$I_{12} = \int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi, \tag{6.74}$$

$$I_{21} = \int_{-1}^{1} u_3|_{\Gamma^3}(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi, \tag{6.75}$$

$$I_{22} = \int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi. \tag{6.76}$$

Which results in:

$$I_{11} = I_{12}, \tag{6.77}$$

$$I_{21} = I_{22}. \tag{6.78}$$

The integrals on the left can easily be evaluated by introducing the appropriate Lagrangian basis:

$$h_i^N \in \mathbb{P}_N([-1,1]),\ h_i\left(\xi_j\right) = \delta_{ij},\ \forall i,j \in \{0,\ldots,N\}^2.$$

Next discretize the functions:

$$u_2|_{\Gamma^2} = \sum_{i=0}^{N} u_2(\xi_0, \eta_i) h_i(\xi) = \sum_{i=0}^{N} \hat{u}_i h_i(\xi) \tag{6.79}$$

$$u_3|_{\Gamma^3} = \sum_{i=0}^{N} u_3(\xi_0, \eta_i) h_i(\xi) = \sum_{i=0}^{N} \check{u}_i h_i(\xi) \tag{6.80}$$

$$\psi|_{\Gamma^2} = \psi|_{\Gamma^3} = \sum_{i=0}^{N} \psi_i h_i(\xi) \tag{6.81}$$

Substitution and numerical integration in the $(N+1)$ GLL nodes yields:

$$\begin{aligned} I_{11} &= \sum_{i=0}^{N} \psi_i \sum_{j=0}^{N} \hat{u}_j \sum_{q=0}^{N} w_q h_i(\xi_q) h_j(\xi_q) \\ &= \sum_{q=0}^{N} w_q \psi_q \hat{u}_q \end{aligned} \tag{6.82}$$

$$\begin{aligned} I_{21} &= \sum_{i=0}^{N} \psi_i \sum_{j=0}^{N} \check{u}_j \sum_{q=0}^{N} w_q h_i(\xi_q) h_j(\xi_q) \\ &= \sum_{q=0}^{N} w_q \psi_q \check{u}_q \end{aligned} \tag{6.83}$$

As mentioned, the right-hand side integrals I_{12} and I_{22} require some more work. This results from the fact that the mortar function itself is defined on γ while the integration is performed on either Γ^2 and Γ^3 which $\neq \gamma$. Therefore two additional mappings are introduced for elements Ω^2 and Ω^3 respectively:

$$\mu_2(\xi) = \frac{1}{2}(\xi - 1) \ \forall \Phi(\xi) \in (s_1, s_2), \tag{6.84}$$

$$\mu_3(\xi) = \frac{1}{2}(1 + \xi) \ \forall \Phi(\xi) \in (s_2, s_3). \tag{6.85}$$

Using the same Lagrangian basis, this yields two discretizations for the mortar solution:

$$\phi_{\mu_2} = \sum_{i=0}^{N} \phi_i h_i(\mu_2(\xi)), \tag{6.86}$$

$$\phi_{\mu_3} = \sum_{i=0}^{N} \phi_i h_i(\mu_3(\xi)). \tag{6.87}$$

With the subscript denoting the applied mapping. The test function $\psi^{(0)}$ remains unchanged.

Substitution and integration yields:

$$\begin{aligned} I_{12} &= \sum_{i=0}^{N} \psi_i \sum_{j=0}^{N} \phi_j \sum_{q=0}^{N} w_q h_i(\xi_q) h_j(\mu_1(\xi_q)) \\ &= \sum_{j=0}^{N} \phi_j \sum_{q=0}^{N} w_q \psi_q h_j(\mu_1(\xi_q)), \end{aligned} \tag{6.88}$$

$$\begin{aligned} I_{22} &= \sum_{i=0}^{N} \psi_i \sum_{j=0}^{N} \phi_j \sum_{q=0}^{N} w_q h_i(\xi_q) h_j(\mu_2(\xi_q)) \\ &= \sum_{j=0}^{N} \phi_j \sum_{q=0}^{N} w_q \psi_q h_j(\mu_2(\xi_q)). \end{aligned} \tag{6.89}$$

Assembling the left and right hand side integrals yields:

$$\sum_{q=0}^{N} w_q \psi_q \hat{u}_q = \sum_{j=0}^{N} \phi_j \sum_{q=0}^{N} w_q \psi_q h_j(\mu_1(\xi_q)), \tag{6.90}$$

$$\sum_{q=0}^{N} w_q \psi_q \check{u}_q = \sum_{j=0}^{N} \phi_j \sum_{q=0}^{N} w_q \psi_q h_j(\mu_2(\xi_q)). \tag{6.91}$$

In ways similar to that of p-refinement the assembled system can be cast in to matrix-vector form:

$$[\psi^N]^T [W^{N\times N}][\hat{u}^N] = [\psi^N]^T [\tilde{W}^{N\times N}][\phi^N], \tag{6.92}$$

$$[\psi^N]^T [W^{N\times N}][\check{u}^N] = [\psi^N]^T [\tilde{W}^{N\times N}][\phi^N]. \tag{6.93}$$

Further simplified:

$$\boldsymbol{\psi}^T W \hat{\mathbf{u}} = \boldsymbol{\psi}^T \tilde{W} \boldsymbol{\phi}, \tag{6.94}$$

$$\boldsymbol{\psi}^T W \check{\mathbf{u}} = \boldsymbol{\psi}^T \tilde{W} \boldsymbol{\phi}. \tag{6.95}$$

Canceling the test function $\boldsymbol{\psi}$ and left-multiplying both sides by $\tilde{W}^{-1}$ yields:

$$\hat{\mathbf{u}} = \underbrace{W^{-1}\tilde{W}}_{\hat{Z}} \boldsymbol{\phi}, \tag{6.96}$$

$$\check{\mathbf{u}} = \underbrace{W^{-1}\tilde{W}}_{\check{Z}} \boldsymbol{\phi}. \tag{6.97}$$

Once again a relation has been established between $u^{(0)}$ and $\phi^{(0)}$. The global projection matrix is similar to equation (6.32) which will eventually lead to a system of discrete equations of the form:

$$Z^T A Z \tilde{\mathbf{u}} = Z^T \mathbf{f}. \tag{6.98}$$

From which, with the help of the projection matrix, $\mathbf{u}$ can be obtained (containing the solution for all three elements).

This covers h-refinement for 0-forms. In the next subsection a Poisson problem for 2-forms will be treated under similar conditions.

6.3.2 2-forms

This final section is devoted to the Poisson problem for 2-forms:

$$\begin{aligned} \mathrm{d}\mathrm{d}^\star u^{(2)} &= f^{(2)} \text{ on } \Omega, \\ \mathrm{tr}\ u^{(2)} &= 0 \quad \text{ on } \partial\Omega. \end{aligned}$$

The Poisson equation is solved on the same mesh as before. It is depicted here once more with adjusted notation in Figure 6.11. It is both dimensionally as well as topologically exactly the

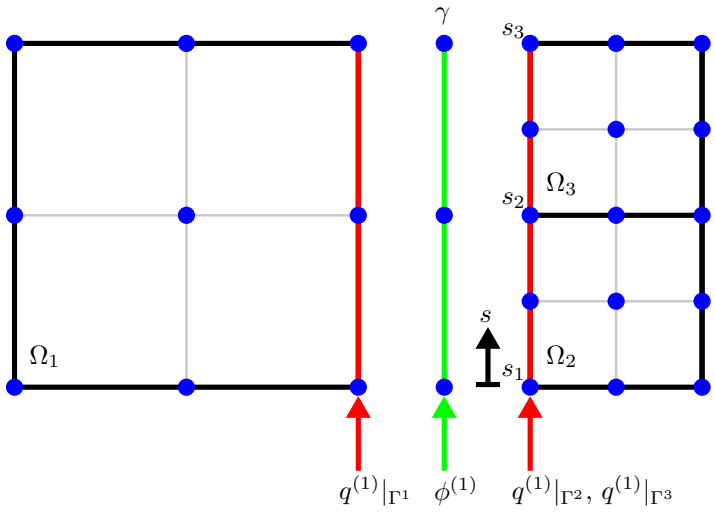


Figure 6.11 – h-Refined mesh for the 2-form Poisson problem.

same, up to and including the mortar γ which connects the elements. Elements Ω_1 through Ω_3 are of equal order N. As with p-refinement, using the 1-form $q^{(1)}$ it is possible to connect the three elements. Repeating the integral matching condition:

$$((q^{(1)}|_\Gamma - \phi^{(1)}), \psi^{(1)}) = \int_\gamma (q^{(1)}|_\Gamma - \phi^{(1)}) \wedge \star\psi^{(1)} = 0.$$

A relation of the form given by equation (6.45) between $q^{(1)}$ and $\phi^{(1)}$ is needed:

$$\mathbf{q}|_\Gamma = Z\boldsymbol{\phi}.$$

With the knowledge of the previous (sub)sections, it is clear that the projection matrix for element Ω_1 will simply be the identity matrix. Therefore, what follows is the concurrent derivation of the projection matrices of elements Ω_2 and Ω_3. Introducing one last time the mortar function $\phi^{(1)}$ (of order N) as well as a test function $\psi^{(1)}$ (also of order N) and substituting into the integral matching condition (again order of the reconstruction en degree of the differential from are not indicated to ensure readability):

$$\int_{\Gamma^2} q_2|_{\Gamma^2}\psi|_{\Gamma^2}\mathrm{d}s = \int_{\Gamma^2} \phi\psi|_{\Gamma^2}\mathrm{d}s = 0, \tag{6.99}$$

$$\int_{\Gamma^3} q_3|_{\Gamma^3}\psi|_{\Gamma^3}\mathrm{d}s = \int_{\Gamma^3} \phi\psi|_{\Gamma^3}\mathrm{d}s = 0. \tag{6.100}$$

These are again expanded using the edge functions (as opposed to Lagrange polynomials):

$$e_i^N \in \mathbb{P}_N([-1,1]),\ \int_{\xi_{j-1}}^{\xi_j} e_i\left(\xi_j\right) = \delta_{ij},\ \forall i,j \in \{1,\ldots,N\}^2.$$

The mapping given by equation (6.65) is used to perform a change of coordinates. This yields:

$$\frac{|\Gamma^2|}{2}\int_{-1}^{1} q_2|_{\Gamma^2}(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi = \frac{|\Gamma^2|}{2}\int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi, \tag{6.101}$$

$$\frac{|\Gamma^3|}{2}\int_{-1}^{1} q_3|_{\Gamma^3}(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi = \frac{|\Gamma^3|}{2}\int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi. \tag{6.102}$$

Canceling constants on both sides:

$$\int_{-1}^{1} q_2|_{\Gamma^2}(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi = \int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi, \tag{6.103}$$

$$\int_{-1}^{1} q_3|_{\Gamma^3}(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi = \int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi. \tag{6.104}$$

Defining the integrals as:

$$I_{11} = \int_{-1}^{1} q_2|_{\Gamma^2}(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi, \tag{6.105}$$

$$I_{12} = \int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^2}(\xi)\mathrm{d}\xi, \tag{6.106}$$

$$I_{21} = \int_{-1}^{1} q_3|_{\Gamma^3}(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi, \tag{6.107}$$

$$I_{22} = \int_{-1}^{1} \phi(\xi)\psi|_{\Gamma^3}(\xi)\mathrm{d}\xi. \tag{6.108}$$

Which results in:

$$I_{11} = I_{12}, \tag{6.109}$$

$$I_{21} = I_{22}. \tag{6.110}$$

Start by discretizing the integrals on the left:

$$q_2|_{\Gamma^2} = \sum_{i=1}^{N} q_2(\xi_0,\eta_i)e_i(\xi) = \sum_{i=1}^{N} \hat{q}_i e_i(\xi), \tag{6.111}$$

$$q_3|_{\Gamma_2} = \sum_{i=1}^{N} q_3(\xi_0,\eta_i)e_i(\xi) = \sum_{i=1}^{N} \check{q}_i e_i(\xi), \tag{6.112}$$

$$\psi|_{\Gamma^2} = \psi|_{\Gamma^3} = \sum_{i=1}^{N} \psi_i e_i(\xi). \tag{6.113}$$

Substitution and numerical integration in the $(N+1)$ GLL nodes yields:

$$I_{11} = \sum_{i=1}^{N} \psi_i \sum_{j=1}^{N} \hat{q}_j \sum_{p=0}^{N} w_p e_i(\xi_p) e_j(\xi_p), \tag{6.114}$$

$$I_{21} = \sum_{i=1}^{N} \psi_i \sum_{j=1}^{N} \check{q}_j \sum_{p=0}^{N} w_p e_i(\xi_p) e_j(\xi_p). \tag{6.115}$$

The right-hand side integral requires the additional mappings given by equations (6.84) and (6.85), yielding for the discretization of $\phi^{(1)}$:

$$\phi_{\mu_2} = \sum_{i=0}^{N} \phi_i e_i(\mu_2(\xi)) J, \tag{6.116}$$

$$\phi_{\mu_3} = \sum_{i=0}^{N} \phi_i e_i(\mu_3(\xi)) J. \tag{6.117}$$

Here J represents the Jacobian (a scaling constant in this case) which ensures the edge functions are mapped correctly. With the test function $\psi^{(1)}$ as defined previously. Substitution and integration yields:

$$I_{12} = \sum_{i=1}^{N} \psi_i \sum_{j=1}^{N} \phi_j \sum_{p=0}^{N} w_p e_i(\xi_p) e_j(\mu_1(\xi_p)), \tag{6.118}$$

$$I_{22} = \sum_{i=1}^{N} \psi_i \sum_{j=1}^{N} \phi_j \sum_{p=0}^{N} w_p e_i(\xi_p) e_j(\mu_2(\xi_p)). \tag{6.119}$$

Assembling the left- and right-hand side integrals yields:

$$\sum_{i=1}^{N} \psi_i \sum_{j=1}^{N} \hat{q}_j \sum_{p=0}^{N} w_p e_i(\xi_p) e_j(\xi_p) = \sum_{i=1}^{N} \psi_i \sum_{j=1}^{N} \phi_j \sum_{p=0}^{N} w_p e_i(\xi_p) e_j(\mu_1(\xi_p)), \tag{6.120}$$

$$\sum_{i=1}^{N} \psi_i \sum_{j=1}^{N} \check{q}_j \sum_{p=0}^{N} w_p e_i(\xi_p) e_j(\xi_p) = \sum_{i=1}^{N} \psi_i \sum_{j=1}^{N} \phi_j \sum_{p=0}^{N} w_p e_i(\xi_p) e_j(\mu_2(\xi_p)). \tag{6.121}$$

In ways similar to that of p-refinement the assembled system can be cast in to matrix-vector form:

$$[\psi^N]^T [W^{N\times N}][\hat{q}^N] = [\psi^N]^T [\tilde{W}^{N\times N}][\phi^N], \tag{6.122}$$

$$[\psi^N]^T [W^{N\times N}][\check{q}^N] = [\psi^N]^T [\tilde{W}^{N\times N}][\phi^N]. \tag{6.123}$$

Further simplified:

$$\boldsymbol{\psi}^T W \hat{\mathbf{q}} = \boldsymbol{\psi}^T \tilde{W} \boldsymbol{\phi}, \tag{6.124}$$

$$\boldsymbol{\psi}^T W \check{\mathbf{q}} = \boldsymbol{\psi}^T \tilde{W} \boldsymbol{\phi}. \tag{6.125}$$

Canceling the test function $\boldsymbol{\psi}$ and left-multiplying both sides by $\tilde{W}^{-1}$ yields:

$$\hat{\mathbf{q}} = \underbrace{W^{-1}\tilde{W}}_{\hat{Z}} \boldsymbol{\phi} \tag{6.126}$$

$$\check{\mathbf{q}} = \underbrace{W^{-1}\tilde{W}}_{\check{Z}} \boldsymbol{\phi} \tag{6.127}$$

The relation between $q^{(1)}$ and $\phi^{(1)}$ locally allows a global projections matrix to be built which eventually results in the (now all familiar) equation:

$$Z^T A Z \begin{bmatrix} \tilde{\mathbf{q}} \\ \mathbf{u} \end{bmatrix} = Z^T \mathbf{f}. \tag{6.128}$$

From which, with the help of the projection matrix, $\mathbf{q}$ can be obtained (containing the solution for all three elements).

As a final note, for both p- and h-refinement the approach can be extended to arbitrarily sized domains. Unlike the derivations shown here, edges or mortar elements will not always correspond to the standard element.

CHAPTER 7

RESULTS

With the theory of the MEM explained in addition to a brief qualitative and quantitative assessment, it is now time to take a more thorough look at some results. This chapter is therefore devoted to various test cases in order to assess the performance of the mimetic MEM implementation. The structure of this chapter is similar to the previous one. Hence, the first half of this chapter (Section 7.1) focuses on p-refinement and contains three test cases for both 0- and 2-forms. Similarly, in the second half (Section 7.2) h-refinement is tested with again another test case for both 0- and 2-forms.

7.1 p-refinement

The main goal of this section is to investigate the behavior of the p-refinement technique presented in the previous chapter. However, in order to choose a proper test case it is important to realize why p-refinement is implemented. Locally increasing the order is deemed necessary whenever the exact solution (the one to which the numerical simulation hopefully converges) contains (many) intricate flow details but remains smooth nonetheless. Evidently this should be reflected in the test cases presented here.

The problem of choice is the Poisson equation which has been described for both 0- and 2-forms in Chapter 5. Simply changing the right-hand side functions as well as the mesh itself allows for some interesting test cases.

The first test case will look familiar since it is taken from the previous chapter. This time however the analysis will be more elaborate.

7.1.1 Test case 1: $u(x, y) = \sin(\pi x^2) \sin(\pi y)$

The domain $\Omega : x, y \in (0, 2) \times (0, 1)$ consists of two elements:

$$\Omega_1 : x, y \in (0, 1) \times (0, 1), \tag{7.1}$$
$$\Omega_2 : x, y \in (1, 2) \times (0, 1). \tag{7.2}$$

This domain is visualized in Figure 7.1. The exact solution u is given by the expression:

$$u(x, y) = \sin\left(\pi x^2\right) \sin(\pi y). \tag{7.3}$$

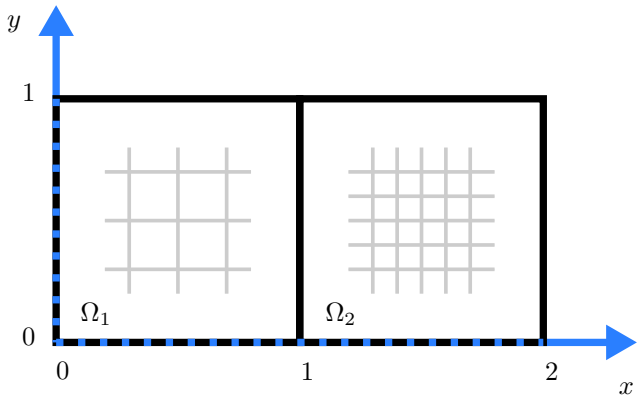
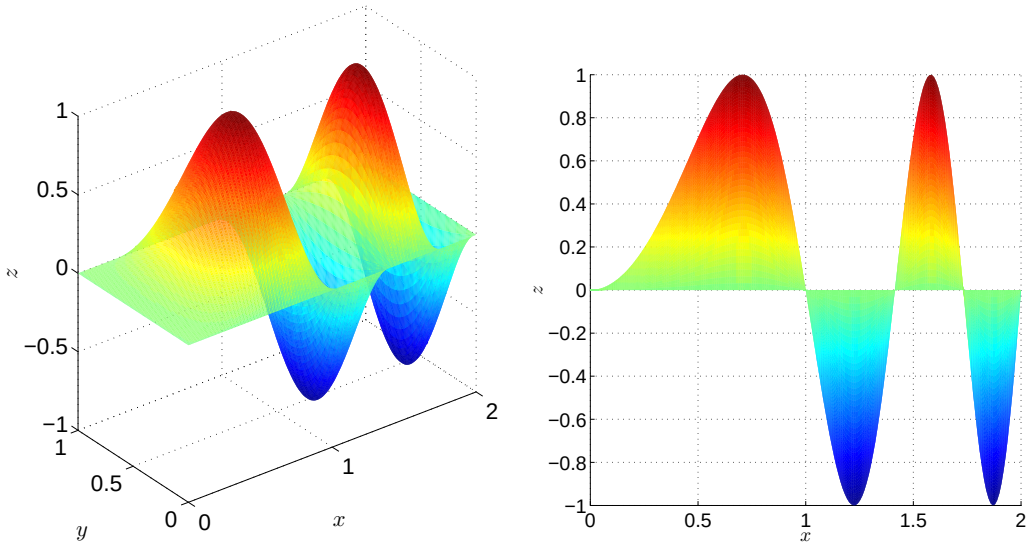


Figure 7.1 – The refined mesh of p-refinement test case 1.

The associated right-hand side function f becomes:

$$f(x,y) = \pi \sin(\pi y)\left(2\cos(\pi x^2) - \pi(4x^2+1)\sin(\pi x^2)\right). \tag{7.4}$$

The exact solution is visualized in Figure 7.2. As mentioned before, it can readily be concluded



(a) Isometric view. **(b)** Side view.

Figure 7.2 – Graphical representation of the exact solution of p-refinement test case 1.

that the right element (containing three extrema as opposed to only one on the left) would benefit from an increase in order. This has been shown to be true in Chapter 6 for both 0- and 2-forms and those results will not be repeated here. More interesting is to take a look at the convergence (as a function of the degrees of freedom) and in what way the left or right element enhances or limits the accuracy of the solution of its neighbor. Additionally, another important aspect is the effect of mortaring on the condition number and sparsity.

This particular test case compares a uniform mesh to a refined mesh. For the uniform approach the order (of both elements) is varied between 2 and 16 with steps of 2. In the refined case the order of element Ω_2 is increased with either 1, 2, 3 or 4 with respect to its left neighbor. Note that these are not necessarily optimal solutions, but solely the choice of the author. Additionally, Tables B.1 and B.2 in Appendix B.1 contain the orders and corresponding number of degrees of freedom for the uniform and refined meshes.

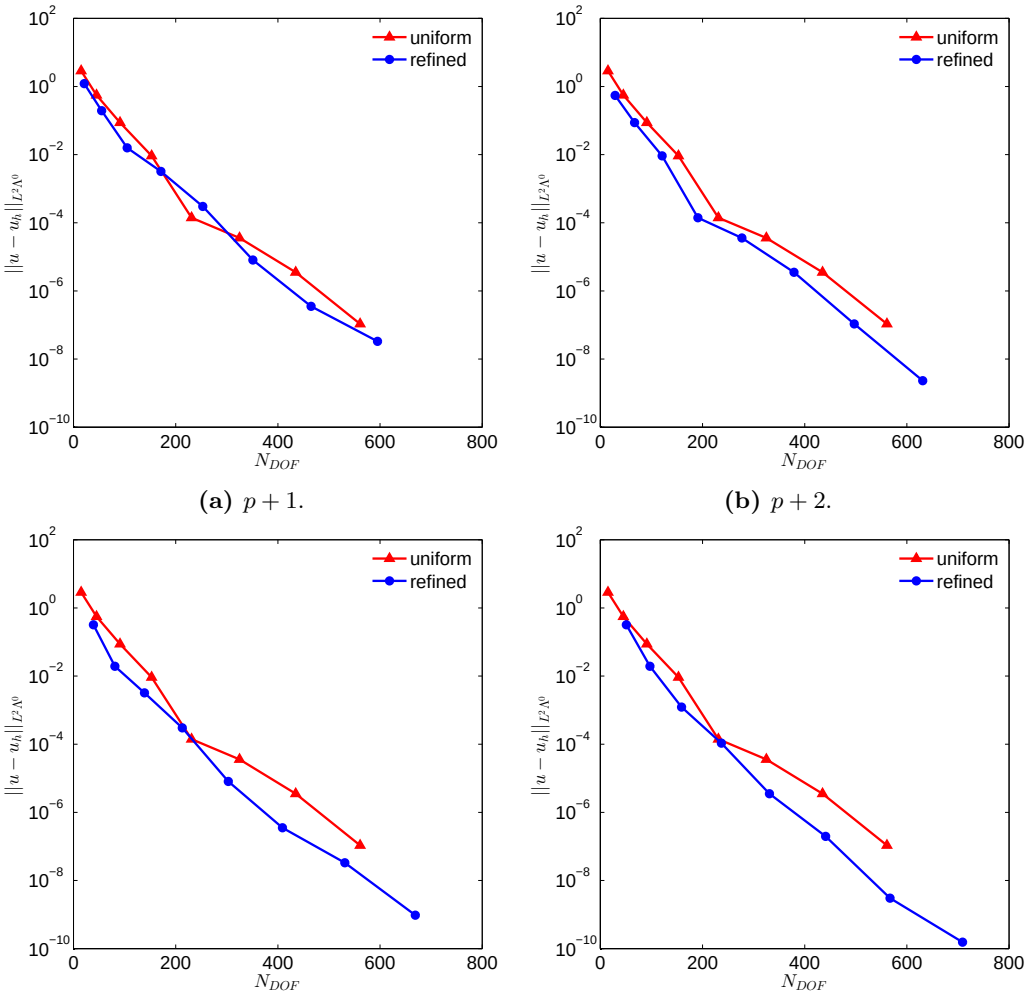
Convergence

The L^2-error investigated here is the total error (i.e. the sum of both element errors). This particular metric is deemed more useful than looking at either the minimum or maximum error. Without much difficulty it can be guessed that the error in element Ω_1 will be significantly lower than the error in element Ω_2 when they are of equal order (as has already been shown). Increasing the order on the right will hopefully bring these values closer together. This will of course bring the maximum error down (and in some cases slightly affect the minimum error as well). Therefore the total error or error sum yields a more consistent indication of the accuracy of the solution. Furthermore, compared to the total error the average error does not provide any new insights.

In Figure 7.3 the $L^2\Lambda^0$-error is shown as a function total number of degrees of freedom N_{DOF} for the 0-form Poisson problem. Ideally, the refined curve (blue dots) should be located below the uniform curve (red triangles). This is only the case when the order of element Ω_2 is increased by 2 with respect to element Ω_1. The difference in error is slightly less than 1 order in magnitude. In other cases the refined error is often less than the uniform one, except at around $p = 8, 10$. In that particular region the graphs converge and in some cases even coincide or intersect each other (making the uniform approach more accurate in that case). This anomaly, visible in all four cases, is found to be specific to this particular test case.

In Figure 7.4 the $L^2\Lambda^0$-error is shown, however this time just for element Ω_1 and as a function of its order p. These graphs show that when the order of the neighboring element Ω_2 is increased, it affects the error in element Ω_1. This effect is limited however, as signified by the graphs of $p + 4$ and $p + 5$ practically coinciding.

Similar results can be obtained for the 2-form Poisson problem. They are depicted in Figures 7.5 and 7.6. First of all, Figure 7.5 shows a trend similar to the one observed for the 0-form problem. The results even appear to be slightly more favorable as the refined graph seems to move down when the order of element Ω_2 is increased further with respect to element Ω_1. Additionally, Figure 7.6 shows the error of just element Ω_1. Once more it decreases when the order of its neighboring element is increased, but eventually reaches a limit as well.



(a) $p + 1$. **(b)** $p + 2$.

(c) $p + 3$. **(d)** $p + 4$.

Figure 7.3 – Error convergence plotted as a function of the number of degrees of freedom for p-refinement test case 1 (0-form) on a uniform and refined mesh.

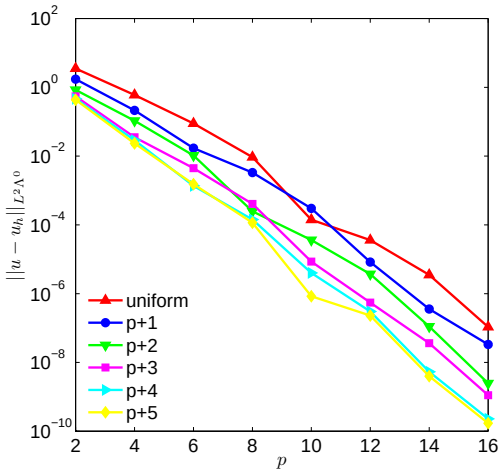
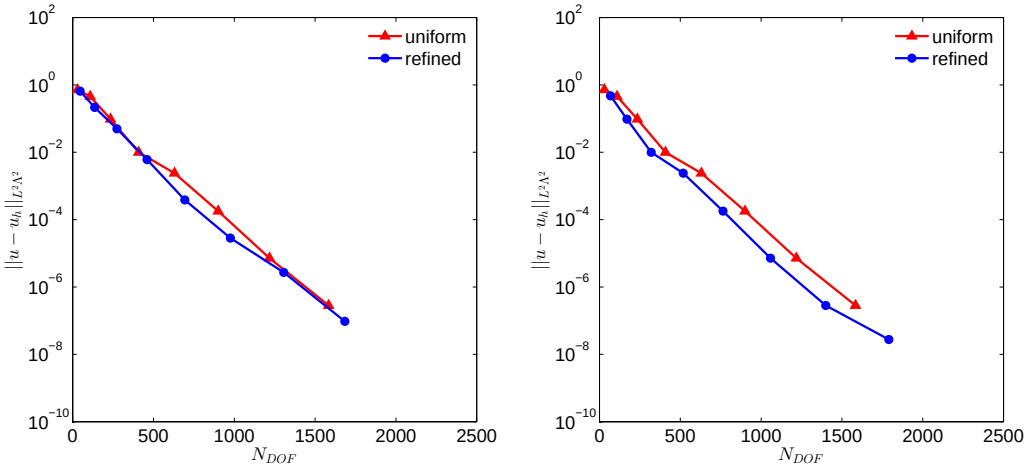


Figure 7.4 – Error convergence plot corresponding to element Ω_1 of p-refinement test case 1 (0-form).



(a) $p + 1$.

(b) $p + 2$.

(c) $p + 3$.

(d) $p + 4$.

Figure 7.5 – Error convergence plotted as a function of the number of degrees of freedom for p-refinement test case 1 (2-form) on a uniform and refined mesh.

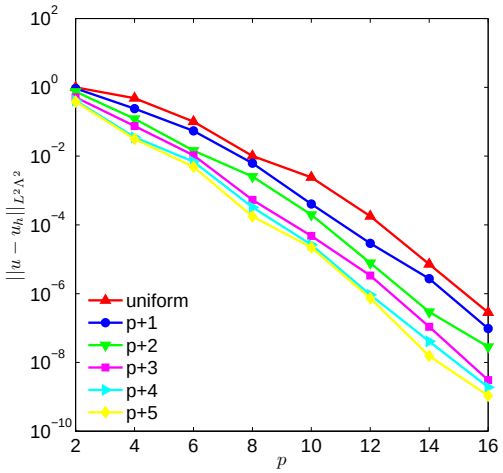


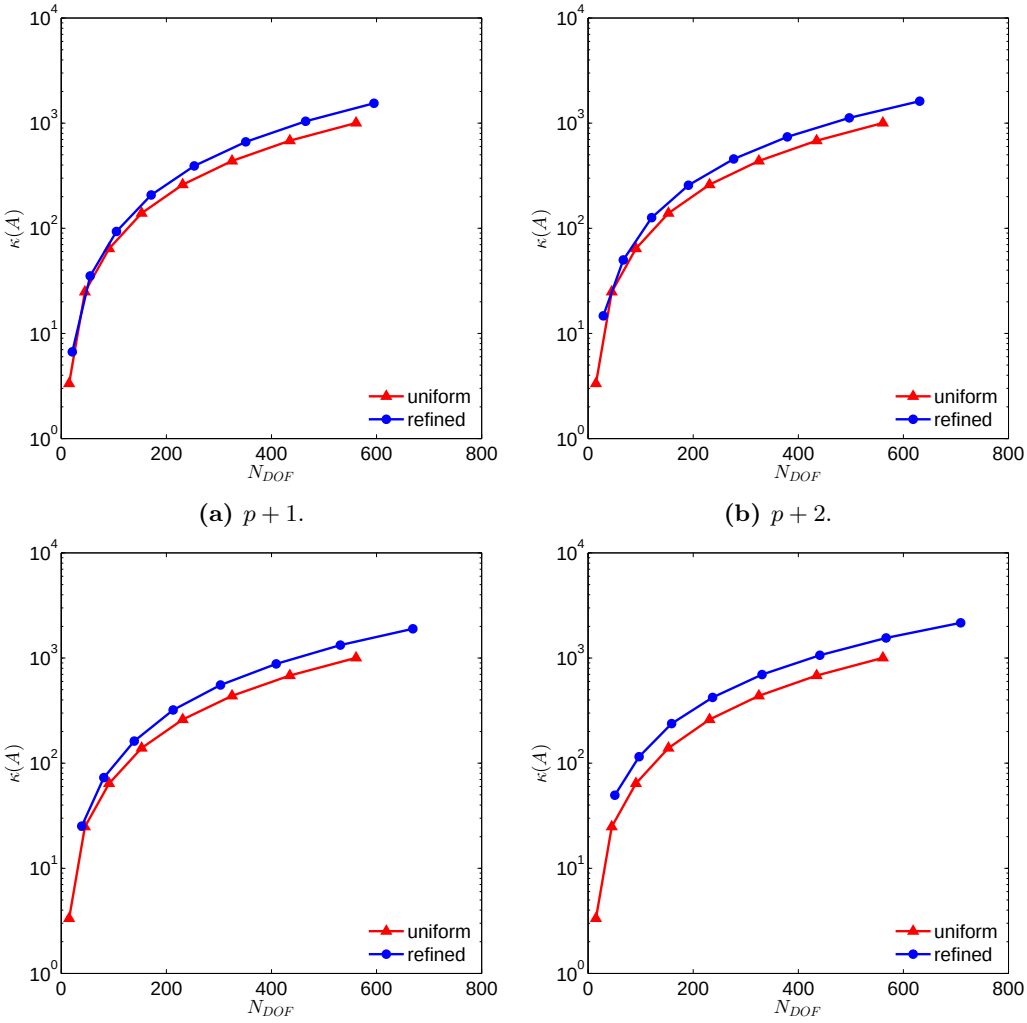
Figure 7.6 – Error convergence plot corresponding to element Ω_1 of p-refinement test case 1 (2-form).

Condition number

Since mortaring alters the system matrix it is considered worthwhile to take a closer look at the condition number. The condition number of a matrix A is defined as:

$$\mathcal{K}(A) = \frac{\lambda_N}{\lambda_1} = ||A|| \cdot ||A^{-1}|| \in [1, \infty[\tag{7.5}$$

Defined as the ratio of the largest over the smallest eigenvalue, the condition number is indicative of the convergence speed with low values being more beneficial. For both the 0-form and 2-form Poisson problem the condition number is given as a function of the total number of degrees of freedom in Figures 7.7 and 7.8. Clearly, the condition number is increased due to the action



(a) $p+1$.

(b) $p+2$.

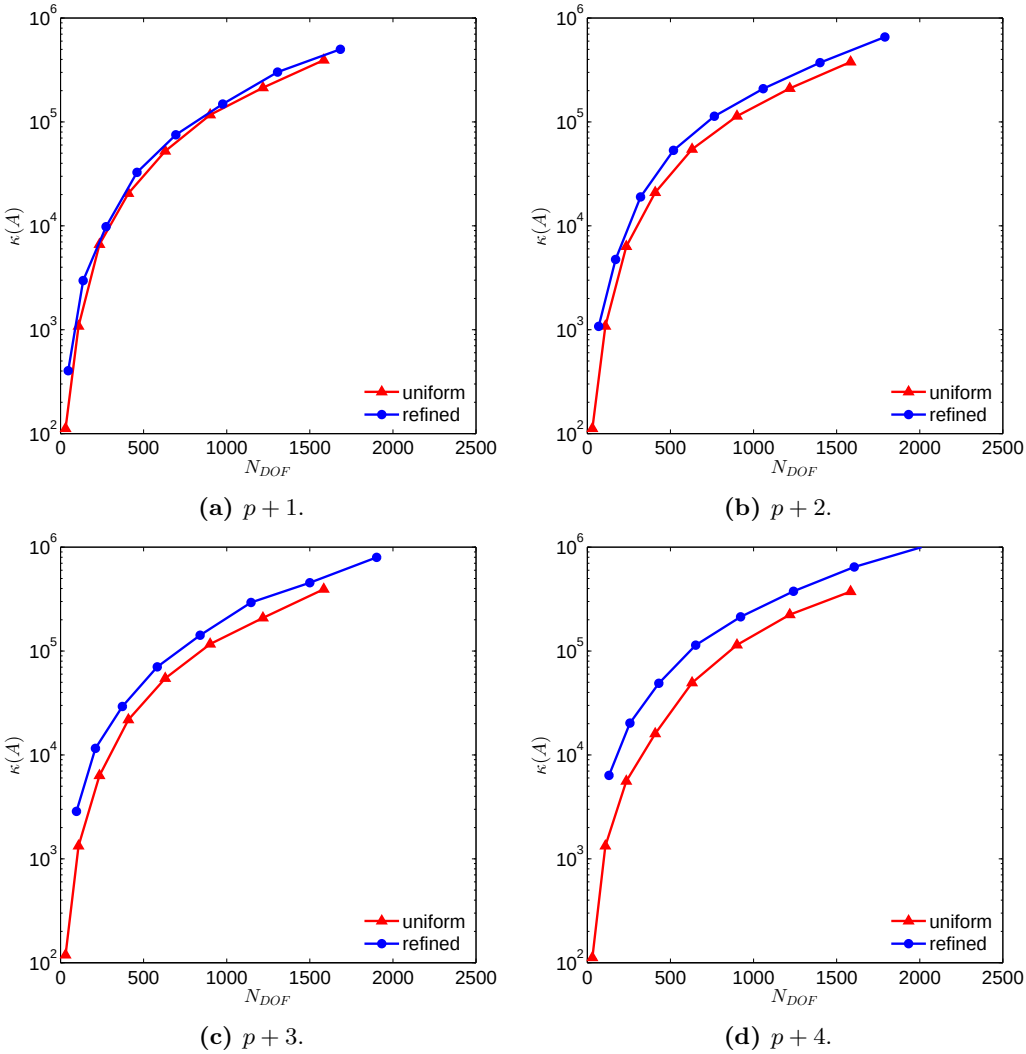
(c) $p+3$.

(d) $p+4$.

Figure 7.7 – Condition number plotted as a function of the number of degrees of freedom for p-refinement test case 1 (0-form) on a uniform and refined mesh.

of the projection matrix. For an equal number of degrees of freedom the condition number is

higher. The same plots have been generated for the 2-form case in Figure 7.8. Clearly, the

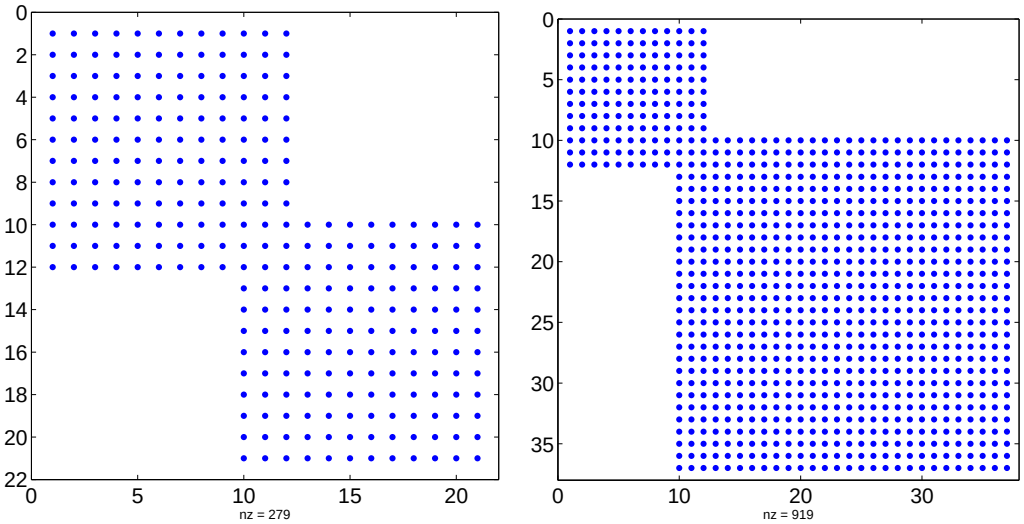


(a) $p+1$. **(b)** $p+2$. **(c)** $p+3$. **(d)** $p+4$.

Figure 7.8 – Condition number plotted as a function of the number of degrees of freedom for p-refinement test case 1 (2-form) on a uniform and refined mesh.

behavior observed for the 0-form problem is also present in the 2-form case.

It is considered worthwhile to take a closer look at the matrix structure (and in particular the sparsity) of both the 0-form and 2-form problem to investigate the effect of the projection matrix. Figure 7.9 displays four spy plots: the upper row is for the 0-form case while the bottom row is for the 2-form case. Each plot on the left corresponds to uniform order $p = 4$ while the one on the right corresponds to the refined case in which the order of the element Ω_2 is increased by 2. At first sight the sparsity in the 2-form case appears to be more significant. Measuring sparsity as the number of non-zero entries NNZ w.r.t. the total number of matrix elements N_{total} yields the results listed in Table 7.1. These numbers confirm that by default the 2-form case yields a significantly more sparse system matrix, which becomes only slightly less sparse upon applying the MEM. This pattern holds true for both 0-forms and 2-forms. For completeness the sparsity



(a) Uniform $p = 4$ (0-form).

(b) Refined $p + 2$ (0-form).

(c) Uniform $p = 4$ (2-form).

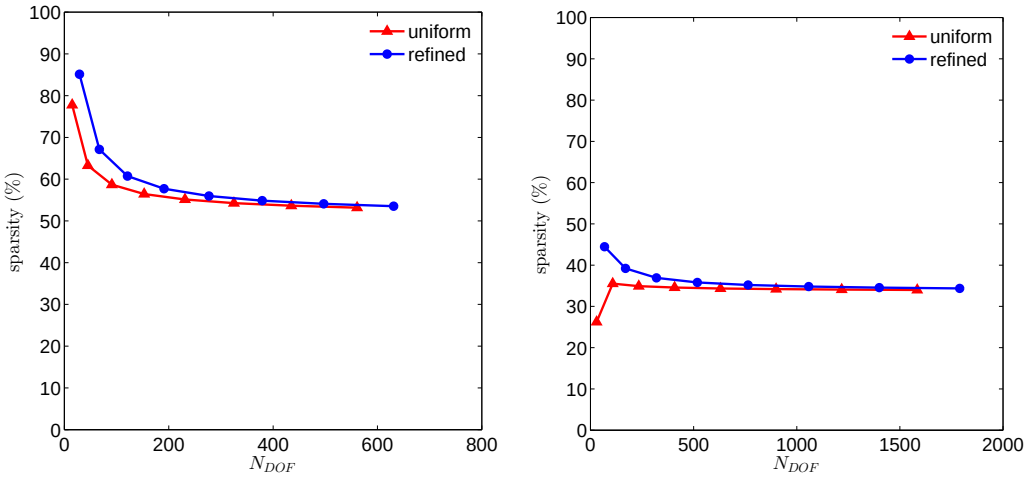
(d) Refined $p + 2$ (2-form).

Figure 7.9 – Comparison of spy plots of the system matrices of p-refinement test case 1 between a uniform and refined mesh.

Table 7.1 – Sparsity of the system matrices of p-refinement test case 1.

	0-form		2-form	
Quantity	$p = 4$	$p + 2$	$p = 4$	$p + 2$
NNZ	279	919	4144	11332
N_{total}	441	1369	11664	28900
%	63	67	36	39

has been plotted as a function of the total number of degrees of freedom in Figure 7.10. The



(a) 0-form. **(b)** 2-form.

Figure 7.10 – Sparsity plotted as a function of the degrees of freedom for p-refinement test case 1. These graphs represent a comparison between a uniform and refined mesh ($p + 2$ case).

general trend is an increase in sparsity with increasing number of degrees of freedom (and thus order).

7.1.2 Test case 2: $u(x,y) = e^{c_1 x}\cos\left(\frac{\pi}{2}x\right)e^{c_2 y}\cos\left(\frac{\pi}{2}y\right)$

The domain $\Omega : x, y \in (-1,1) \times (-1,1)$ is split up into nine square elements, with three in each direction. This domain is visualized in Figure 7.11. The exact solution u is given by the

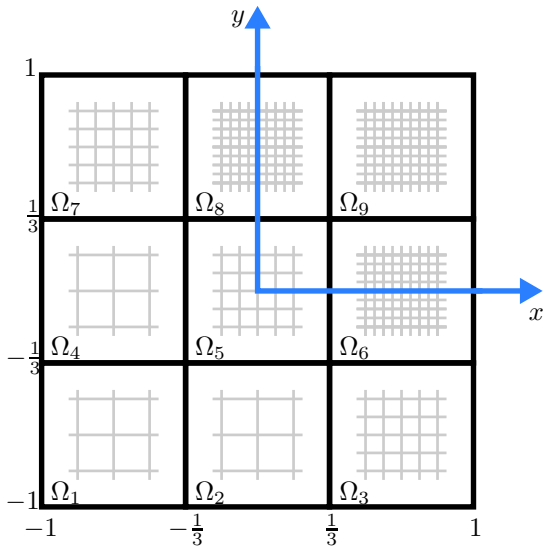


Figure 7.11 – The refined mesh of p-refinement test case 2.

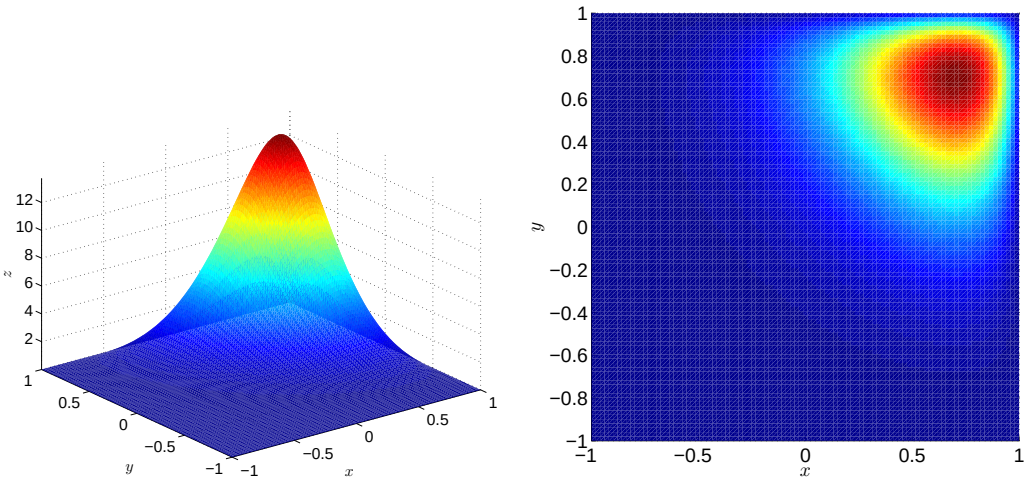
expression:

$$u(x,y) = e^{c_1 x}\cos\left(\frac{\pi}{2}x\right)e^{c_2 y}\cos\left(\frac{\pi}{2}y\right). \tag{7.6}$$

The associated right-hand side function f is:

$$\begin{aligned} f(x,y) =& e^{c_2 y}\cos(\frac{\pi}{2}y)e^{c_1 x}\left((c_1^2 - (\frac{\pi}{2}^2)\cos(\frac{\pi}{2}x) - c_1\pi\sin(\frac{\pi}{2}x)\right) + \\ & e^{c_1 x}\cos(\frac{\pi}{2}x)e^{c_2 y}\left((c_2^2 - (\frac{\pi}{2}^2)\cos(\frac{\pi}{2}y) - c_2\pi\sin(\frac{\pi}{2}y)\right). \end{aligned} \tag{7.7}$$

The exact solution when $c_1 = c_2 = 3$ is visualized in Figure 7.12. Considering the large peak in the upper right corner of the domain, it seems worthwhile to increase the order gradually in that direction. To investigate once more the error convergence as well as the condition number a uniform and refined mesh are proposed. These meshes are depicted in Figure 7.13. It is expected that in order to reach the same error, in the uniform case more degrees of freedom (i.e. a higher p) will be needed. For both the uniform and refined mesh the order p is varied between 2 and 10 with steps of 2. Tables B.3 and B.4 in Appendix B.1 contain the orders and corresponding number of degrees of freedom for the uniform and refined meshes.



(a) Isometric view. **(b)** Top view.

Figure 7.12 – Graphical representation of the exact solution of p-refinement test case 2.

p	p	p
p	p	p
p	p	p

(a) Uniform mesh.

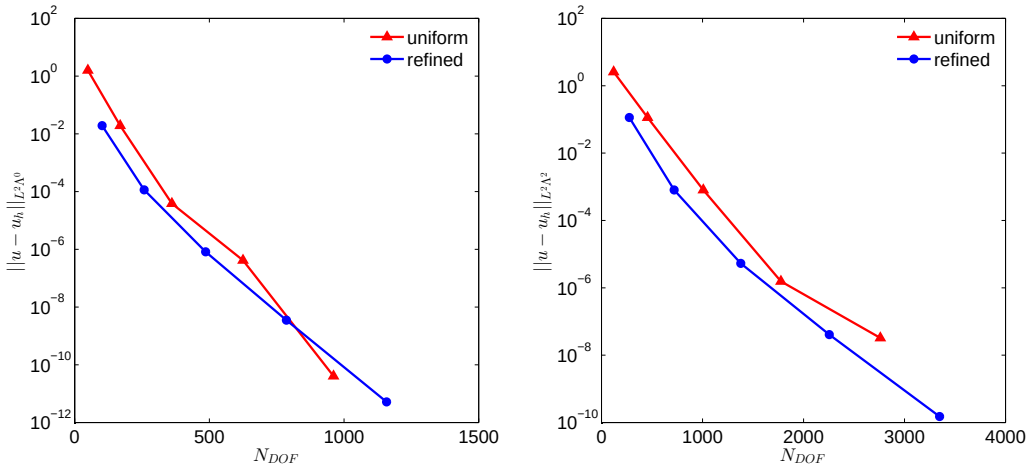
$p+1$	$p+2$	$p+2$
p	$p+1$	$p+2$
p	p	$p+1$

(b) Refined mesh.

Figure 7.13 – Element order for the uniform and refined meshes of p-refinement test case 2.

Convergence

Once the again the sum of the element errors is plotted against the number of degrees of freedom. Convergence results for both the 0-form and 2-form Poisson problem are given in Figure 7.14. For this particular test case the improvement appears to be relatively small for 0-forms. There

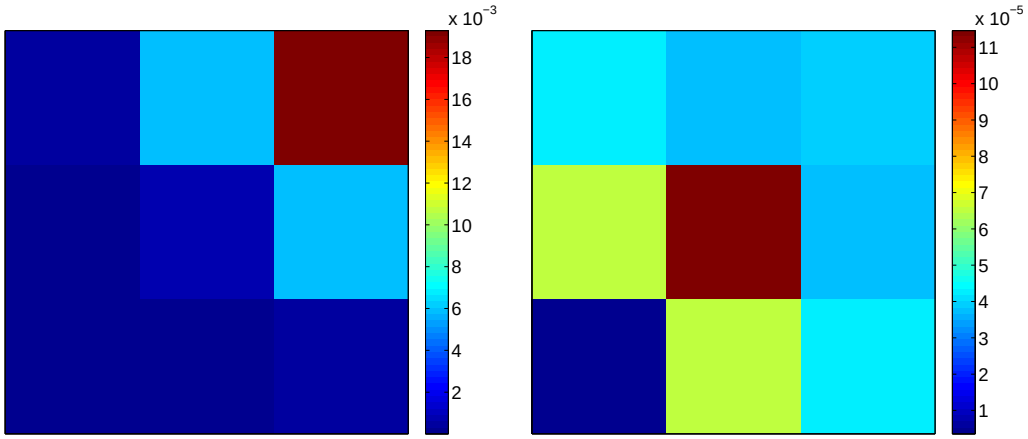


(a) 0-form. **(b)** 2-form.

Figure 7.14 – Error convergence plotted as a function of the number of degrees of freedom for p-refinement test case 2 (0-form and 2-form) on a uniform and refined mesh.

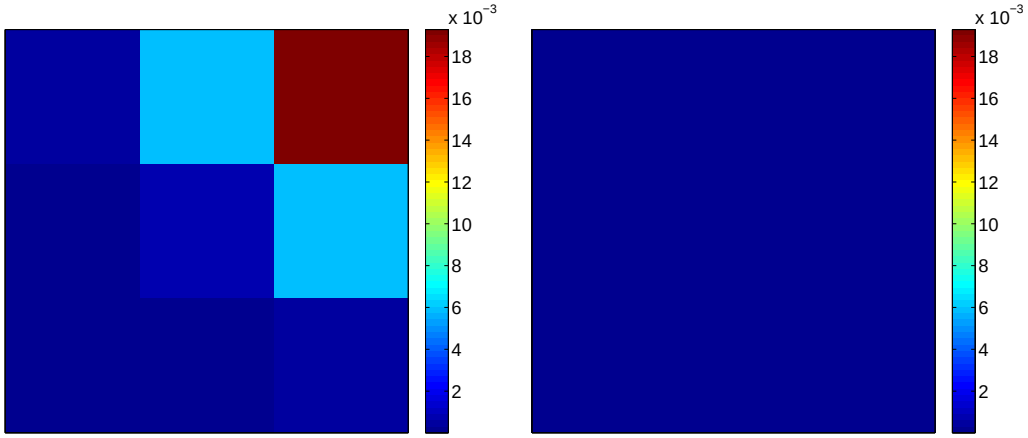
are still some intersections and points where the convergence graphs intersect of coincide. For 2-forms the results slightly better, but leave room for improvement nonetheless. It is expected that this is caused by the small scale of this problem.

At this point it is considered interesting to take a closer look at the individual element errors. To that end Figure 7.15 displays the element error for a uniform and refined mesh with $p = 4$ for 0-forms. On the upper row the colorbar is scaled relative to itself, while on the lower row the colorbar is set equal for both the uniform and refined case. The former gives insight into which individual elements have the greatest error in each case while the latter allows a comparison to be made between the uniform and refined case. In Figure 7.15a it can readily be seen that the largest error occurs where the peak is. In Figure 7.15b the greatest error is now encountered in the element at the center of the domain, which is clearly the result of p-refinement. Looking at the row of figures below that, the solution on the refined mesh shows a significant improvement over the uniform one. For 2-forms a similar result is displayed in Figure 7.16. Here the same trend is visible.



(a) Uniform, $p = 4$ (relative error).

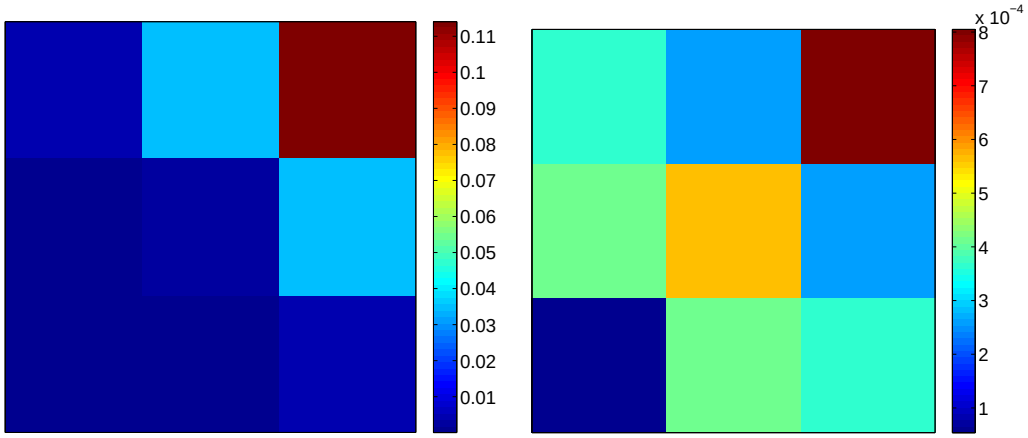
(b) Refined (relative error).



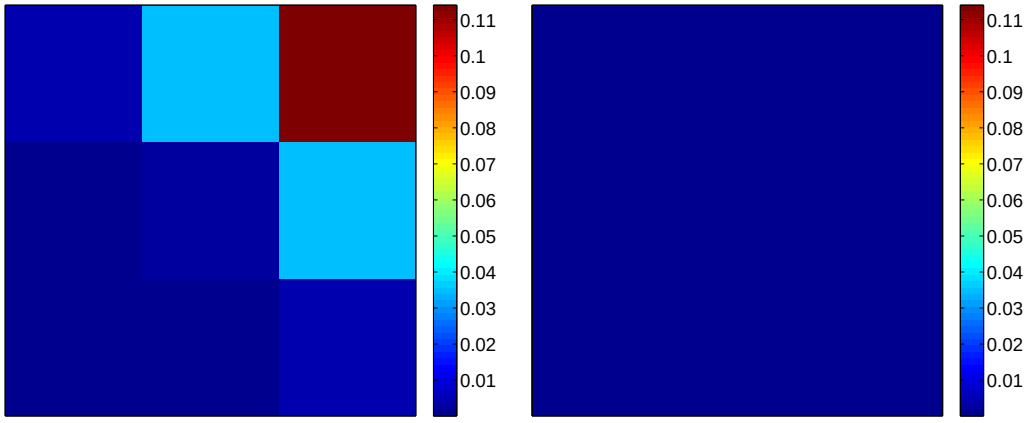
(c) Uniform $p = 4$, (absolute error).

(d) Refined (absolute error).

Figure 7.15 – Individual element error comparison for p-refinement test case 2 (0-form).



(a) Uniform $p = 4$ (relative error). **(b)** Refined (relative error).

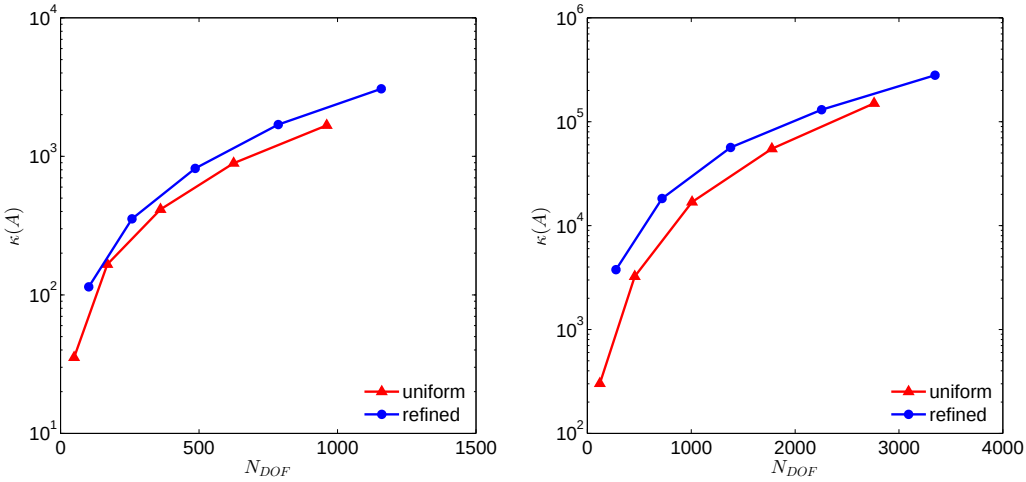


(c) Uniform $p = 4$ (absolute error). **(d)** Refined (absolute error).

Figure 7.16 – Individual element error comparison for p-refinement test case 2 (2-form).

Condition number

Like test case 1, the condition number is investigated in order to see how the mortaring affects the system matrix of a larger system. Again for both the 0-form and 2-form Poisson problem these results are given in Figure 7.17. The behavior of the condition number is similar as to



(a) 0-form. **(b)** 2-form.

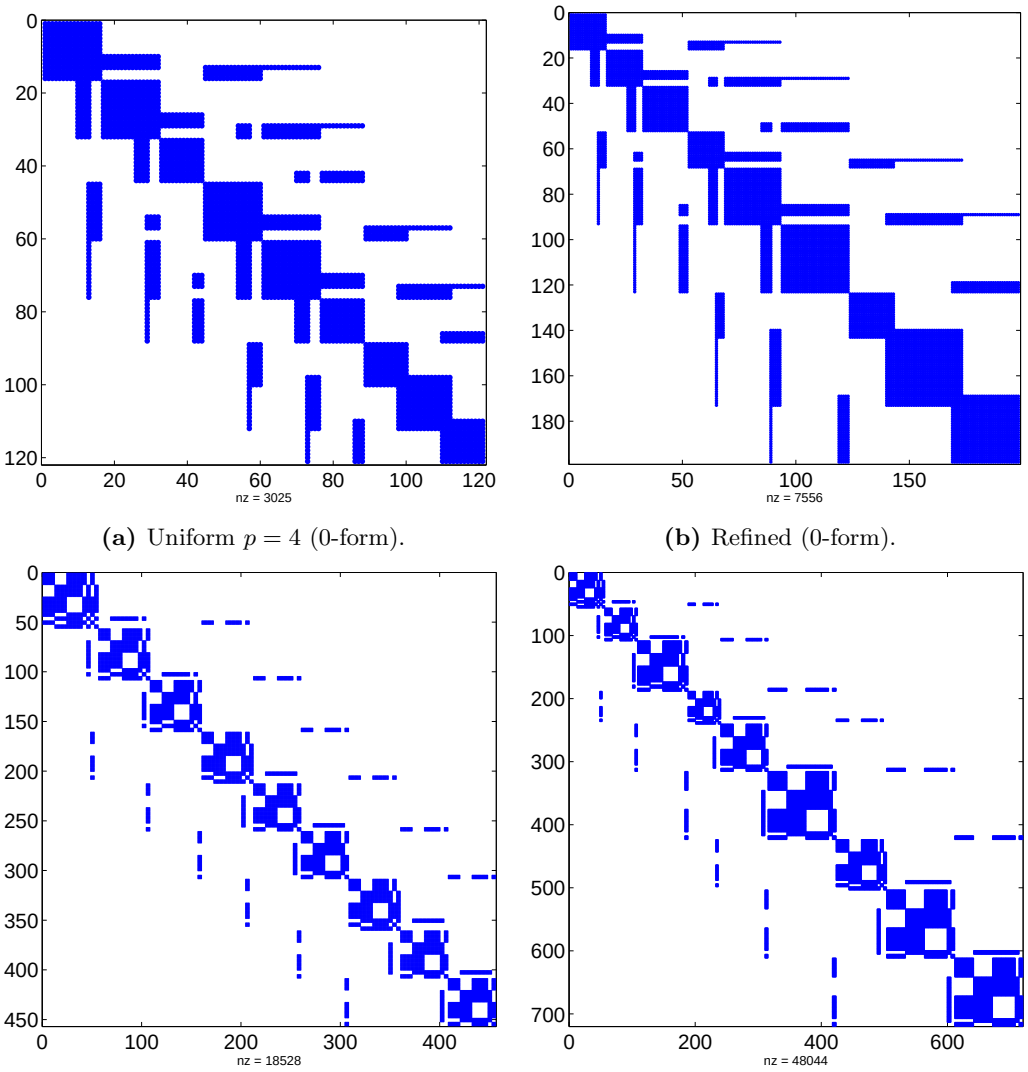
Figure 7.17 – Condition number plotted as a function of the number of degrees of freedom for p-refinement test case 2 (0-form and 2-form) on a uniform and refined mesh.

what is observed for test case 1. Mortaring increases the condition number by shifting the curve (slightly) upwards to higher values. The sparsity pattern is of course structurally similar to the one observed for test case 1 as can be seen in figure (it is still a Poisson problem, only the mesh has changed). At first sight the sparsity appears to have increased due to the increase in number of elements. Additionally, between the 0-form and 2-form problem, the latter is again more sparse. Table 7.2 summarizes sparsity data by listing the number of non-zero entries, the total number of matrix elements and sparsity (the number of non-zero entries expressed as a percentage of the total number of matrix elements). Figure 7.19 displays the sparsity as a function of the total

Table 7.2 – Measuring sparsity for the 0-form and 2-form Poisson test case 2.

	0-form		2-form	
quantity	$p=4$	refined	$p=4$	refined
NNZ	3025	7556	18528	48044
N_{total}	14641	39204	207936	516961
%	21	19	9	9

number of degrees of freedom. The general trend is an increase in sparsity with an increasing number of degrees of freedom (and thus order). This was also observed in the previous test case.



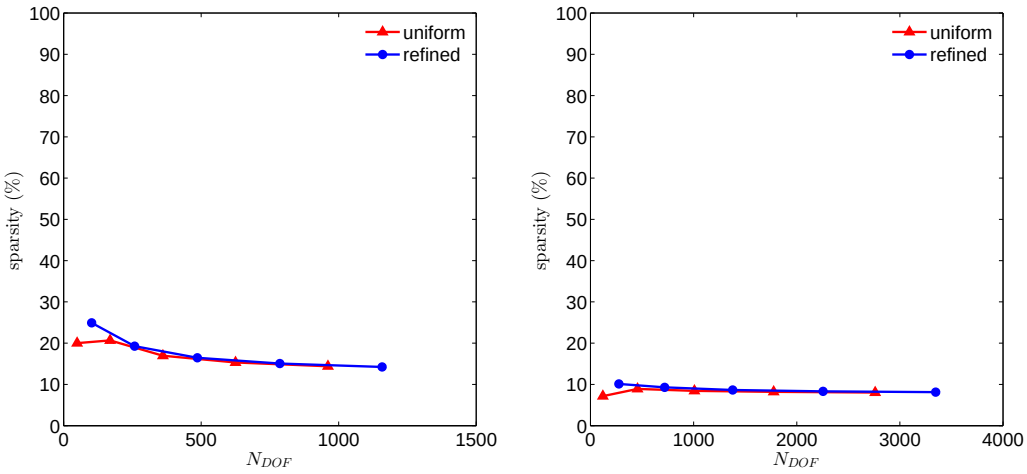
(a) Uniform $p = 4$ (0-form).

(b) Refined (0-form).

(c) Uniform $p = 4$ (2-form).

(d) Refined (2-form).

Figure 7.18 – Comparison of spy plots of the system matrices of p-refinement test case 2 between a uniform and refined mesh.



(a) 0-form. **(b)** 2-forms.

Figure 7.19 – Sparsity plotted as a function of the degrees of freedom for p-refinement test case 2. These graphs represent a comparison between a uniform and refined mesh.

7.1.3 Test case 3: $u(x,y) = 100(x-x^2)(y-y^2)e^{-50((x-1)^2+(y-1)^2)}$

The third and final test case for p-refinement is similar to test case 2. Once again there is a large peak located in the upper right corner of the domain. However, this time the peak does not span multiple elements. It resides solely in a single element. The aim of this test case is to determine the value of the maximum as accurately as possible.

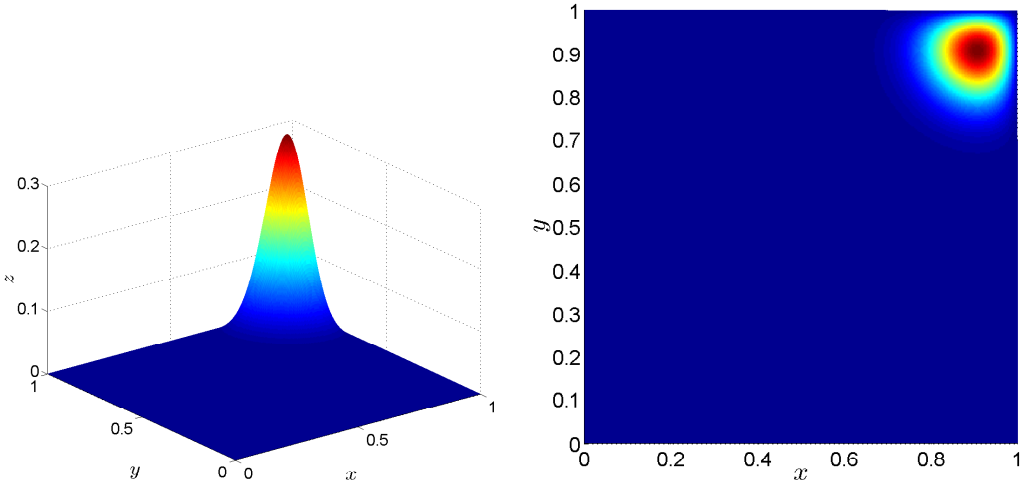
The domain $\Omega : x, y \in (-1,1) \times (-1,1)$ is once more split up into nine square elements, with three in each direction (as depicted in Figure 7.11 in the previous section). The exact solution u is given by the expression:

$$u(x,y) = 100(x-x^2)(y-y^2)e^{-50((x-1)^2+(y-1)^2)}. \tag{7.8}$$

And the associated right-hand side function f by:

$$\begin{aligned} f(x,y) =& 200e^{-50((x-1)^2+(y-1)^2)}(5000x^4(y-1)y - 15000x^3(y-1)y + \\ & x^2(5000y^4 - 15000y^3 + 29500y^2 - 19400y - 99) + \\ & (-5000y^4 + 15000y^3 - 19400y^2 + 9300y + 99) - 99(y-1)y. \end{aligned} \tag{7.9}$$

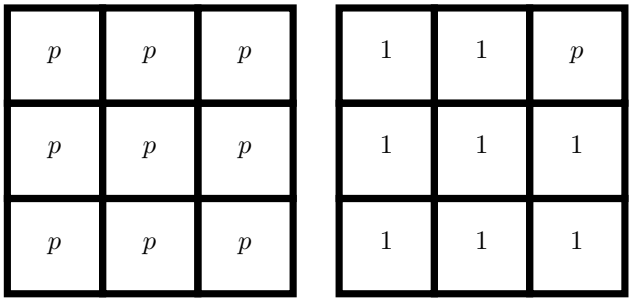
This is also visualized in Figure 7.20. The exact location of the maximum is at $x = y = 0.905$



(a) Isometric view. **(b)** Top view.

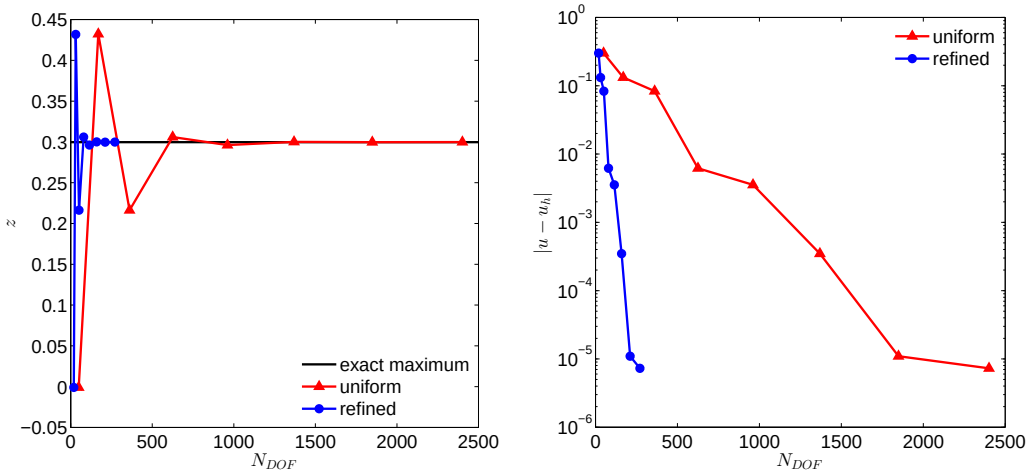
Figure 7.20 – Graphical representation of the exact solution of p-refinement test case 3.

where it attains a value of $z = 0.2998$. Aside from the peak located in the upper right corner element, the solution is zero everywhere else. The refined mesh is therefore built up of linear elements, except for the one in the upper right corner. The uniform and refined mesh are depicted in Figure 7.21. Once again the order p is varied between 2 and 16 with steps of 2. Results of the 0-form Poisson problem are depicted in Figure 7.22. Figure 7.22a shows the convergence of the solution on both meshes to the exact maximum while Figure 7.22b shows the error between the approximated and exact maximum (absolute difference). On the refined mesh the solution convergences rapidly to the exact maximum compared to the uniform one. This is mirrored by the progression of the error as well. Hence, the use of linear elements throughout large parts



(a) Uniform mesh. **(b)** Refined mesh.

Figure 7.21 – Element order for the uniform and refined meshes of p-refinement test case 3.

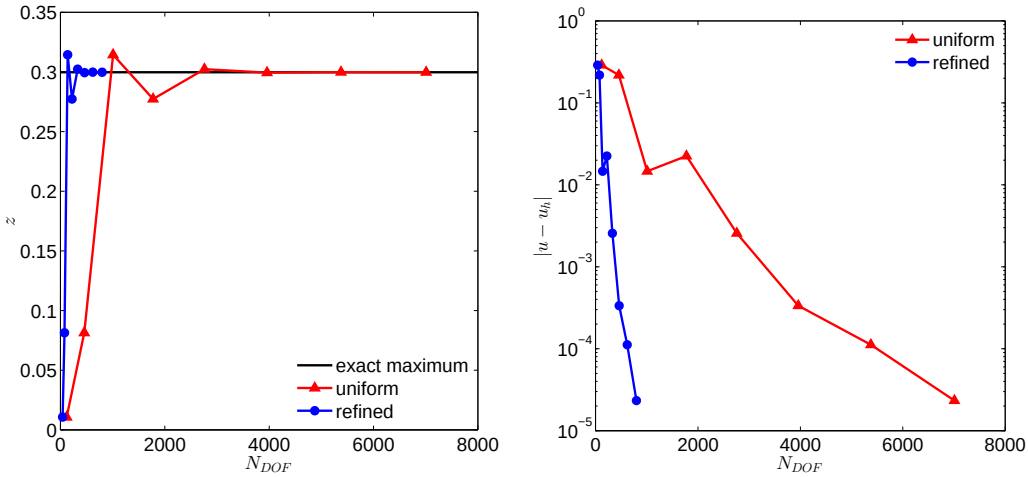


(a) Peak value. **(b)** Peak error.

Figure 7.22 – Peak value and error convergence plotted as a function of the degrees of freedom for p-refinement test case 3 (0-forms).

of the domain appears to be justified. For reference purposes, the order p and corresponding number of degrees of freedom are listed in Table B.5 in Appendix B.1.

Results of the 2-form Poisson problem are depicted in Figure 7.23. Figure 7.23a shows the



(a) Peak value. **(b)** Peak error.

Figure 7.23 – Peak value and error convergence plotted as a function of the degrees of freedom for p-refinement test case 3 (2-forms).

convergence of the solution on both meshes to the exact maximum while Figure 7.23b shows the error between the approximated and exact maximum (absolute difference). The same trend reappears: convergence to the exact maximum is much faster while the error is much lower for a given number of degrees of freedom. The use of a refined mesh proves its use. Once again, the order p and corresponding number of degrees of freedom are listed in Table B.6 in Appendix B.1.

In addition to yielding more accurate results for a given number of degrees of freedom, the performance is improved significantly as well. Computation times (on an AMD Phenom II X4 965 Processor (3.40 GHz)) for both the 0-form and 2-form problem are listed in Table 7.3. The

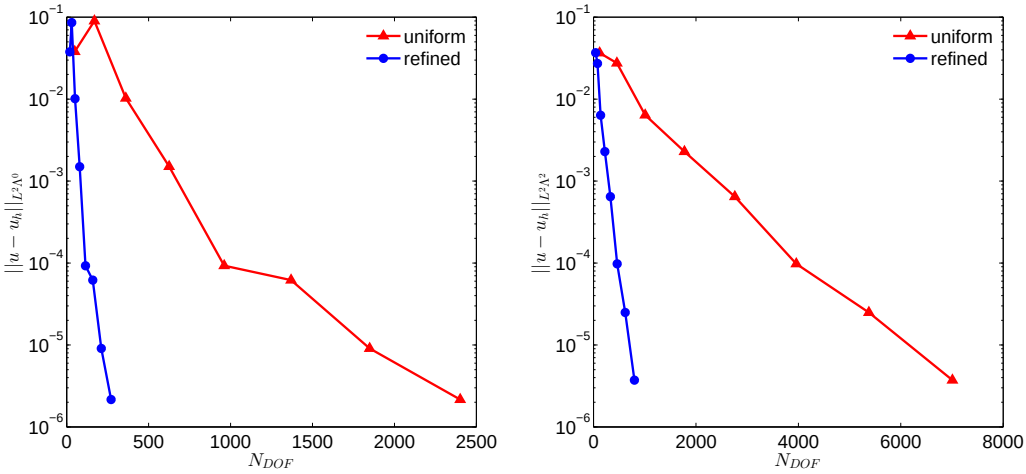
Table 7.3 – Performance measurements for p-refinement test case 3.

mesh	0-form	2-form
uniform	5.8 s	87.4 s
refined	0.5 s	1.8 s

computation times listed in this table correspond to the complete run from order 2 to 16. These numbers show a significant speed up, especially in the case of the 2-form Poisson problem.

For completeness, the total error sum (element errors summed up) for both problems is given in Figure 7.24. The improvement is much more pronounced compared to previous test cases. This demonstrates the merits of p-refinement and more importantly, that the mimetic variant of the MEM works.

As a final note, for this particular test case (and the one in the section following this one) condition number and sparsity are not investigated further as previous test cases already provide sufficient insight.



(a) 0-form. **(b)** 2-form.

Figure 7.24 – Total error convergence as a function of the number of degrees of freedom for p-refinement test case 3 (0-form and 2-form) on a uniform and refined mesh.

7.2 h-refinement

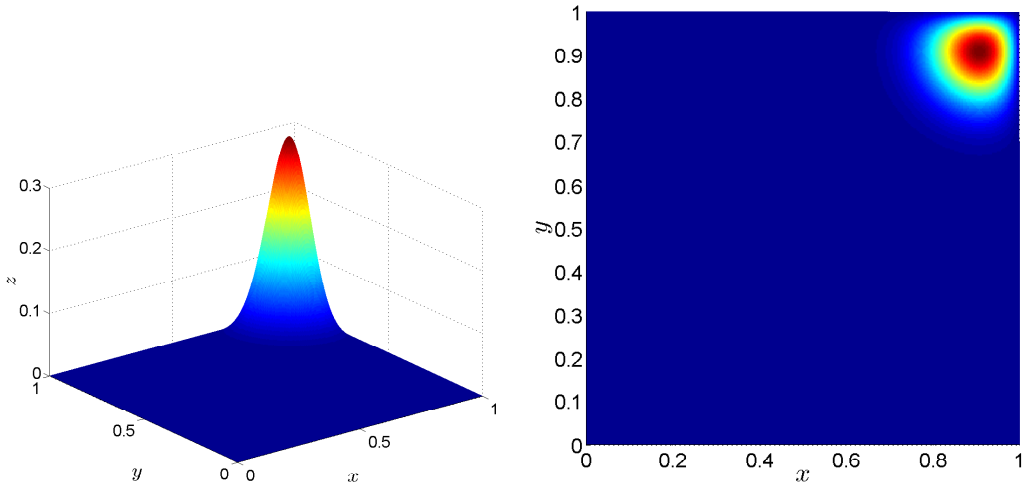
Although the discretizations for h-refinement are very much similar to the ones for p-refinement, its area of application is different. By reusing test case 3 for p-refinement from the previous section, these differences can be highlighted.

7.2.1 Test case 1: $u(x,y) = 100(x-x^2)(y-y^2)e^{-50((x-1)^2+(y-1)^2)}$

Although this test case has been used before, its details will be repeated here for convenience. First recall the exact solution:

$$\begin{aligned} u(x,y) =& 100(x-x^2)(y-y^2)e^{-50((x-1)^2+(y-1)^2)}, \\ f(x,y) =& 200e^{-50((x-1)^2+(y-1)^2)}(5000x^4(y-1)y - 15000x^3(y-1)y + \\ & x^2(5000y^4 - 15000y^3 + 29500y^2 - 19400y - 99) + \\ & (-5000y^4 + 15000y^3 - 19400y^2 + 9300y + 99) - 99(y-1)y. \end{aligned}$$

As mentioned before, it exhibits a localized peak at $x = y = 0.905$ where it reaches its maximum value of $z = 0.2998$ as depicted in Figure 7.25. The aim of this test case is to determine again the value of the maximum as accurately as possible. The domain $\Omega : x, y \in (-1,1) \times (-1,1)$ is split up in elements differently now. In total six configurations are used, all of which are depicted in Figure 7.26. The captions to each mesh denotes the element configuration. For example, [21] mean it has 2 elements in x-direction and 1 in y-direction. Moreover, [h] denotes the h-refined mesh. Finally, the addition -f indicates a further refinement (in this case a resizing of the elements; something which is not always necessarily done in practice) of the original mesh. Note that the current implementation for h-refinement only allows uniform element order. Once again this is varied between 2 and 16 with steps of 2. Results of the 0-form Poisson problem are depicted in Figure 7.27. Figure 7.27a shows the convergence of the solution on the different meshes to the exact maximum while Figure 7.27b shows the error between the approximated



(a) Isometric view. **(b)** Top view.

Figure 7.25 – Graphical representation of the exact solution of h-refinement test case 1.

and exact maximum (absolute difference). Unlike p-refinement test case 3, these graphs depict a less dramatic improvement. First focus on the (cyan colored) graph corresponding to the basic h-refined mesh (Figure 7.26e). Although it is on par (or slightly better than) most of the other meshes, its results are not better than those corresponding to the situation in which the complete domain is covered by just a single element (red colored curve). However, when the elements are sized more appropriately (resulting in the mesh of Figure 7.26f), the error (signified by the yellow colored graph) improves significantly to the point where it becomes better than the single element case. It should be noted however that a similar improvement is observed for the mesh of Figure 7.26d (signified by the magenta colored graph). For reference purposes, the order p and corresponding number of degrees of freedom are listed in Table B.7 in Appendix B.2.

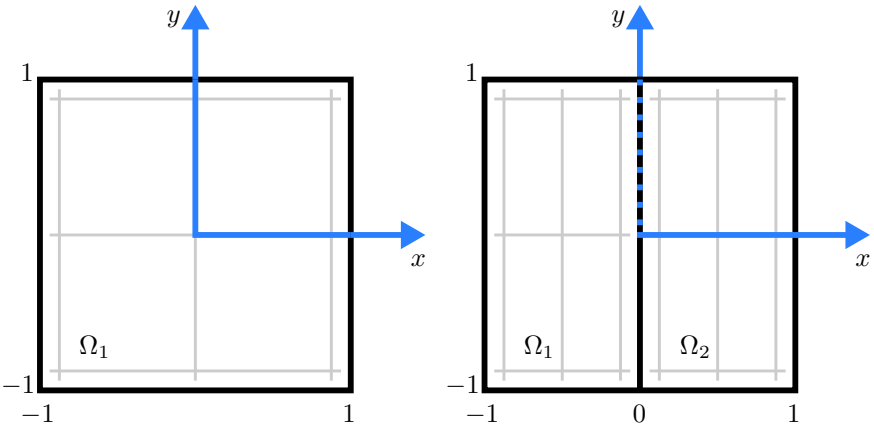
Results of 2-form Poisson problem are depicted in Figure 7.28. Figure 7.28a shows the convergence of the solution on the different meshes to the exact maximum while Figure 7.28b shows the error between the approximated and exact maximum (absolute difference). These graphs mirror much of trends seen in the 0-form case. The convergence does however exhibit slightly erratic behavior. Once again, the order p and corresponding number of degrees of freedom are listed in Table B.8 in Appendix B.2.

Performance-wise, improvements are obtained as well as shown in Table 7.4. These compu-

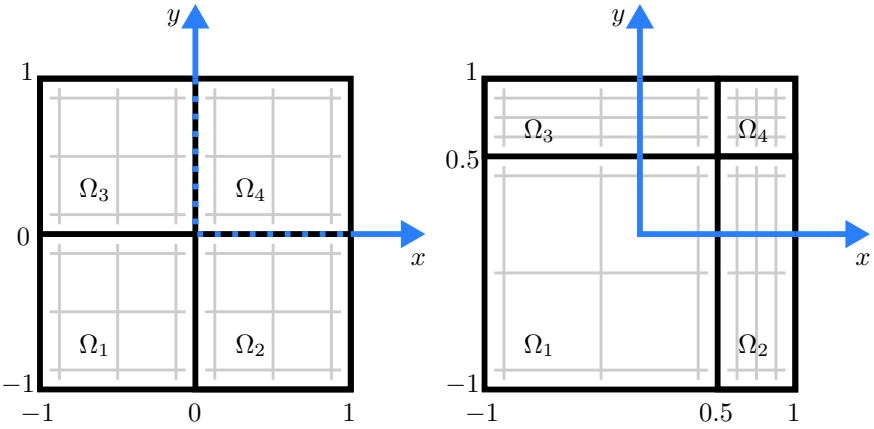
Table 7.4 – Performance measurements for h-refinement test case 1.

mesh	0-form	2-form
[11]	0.3 s	1.5 s
[21]	0.5 s	4.8 s
[22]	1.2 s	16.4 s
[22]-f	1.4 s	16.9 s
[h](-f)	0.9 s	9.9 s

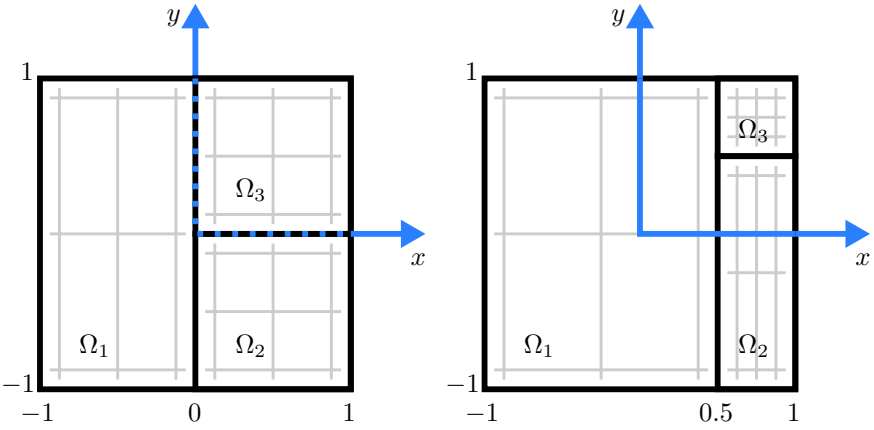
tation times correspond to the complete run from order 2 to 16. The speed up is less significant



(a) [11]. **(b)** [21].

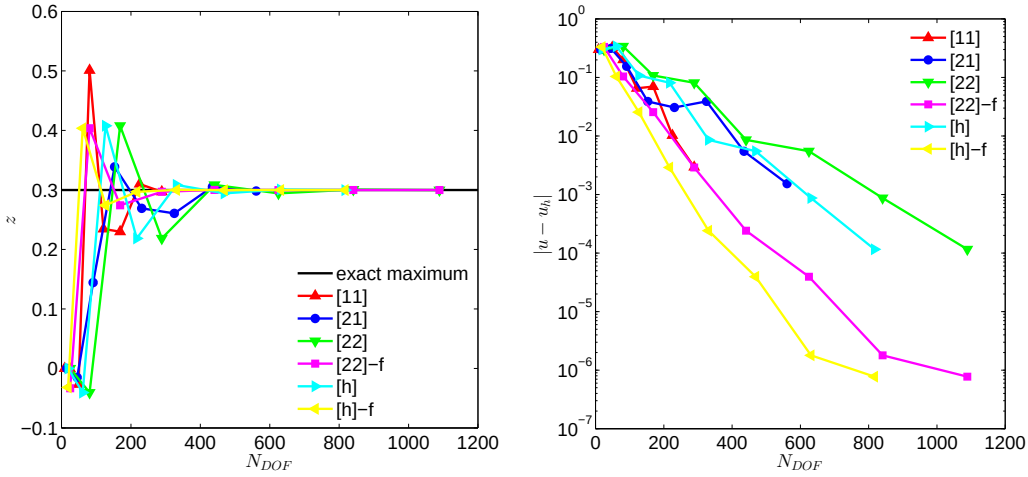


(c) [22]. **(d)** [22]-f.



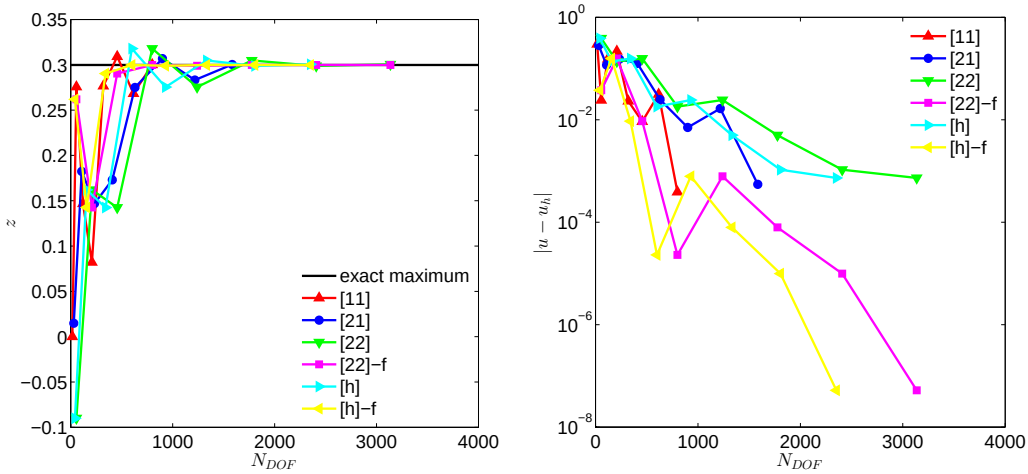
(e) [h]. **(f)** [h]-f.

Figure 7.26 – Refined meshes of h-refinement test case 1.



(a) Peak value. **(b)** Peak error.

Figure 7.27 – Peak value and error convergence plotted as a function of the degrees of freedom for h-refinement test case 1 (0-forms).

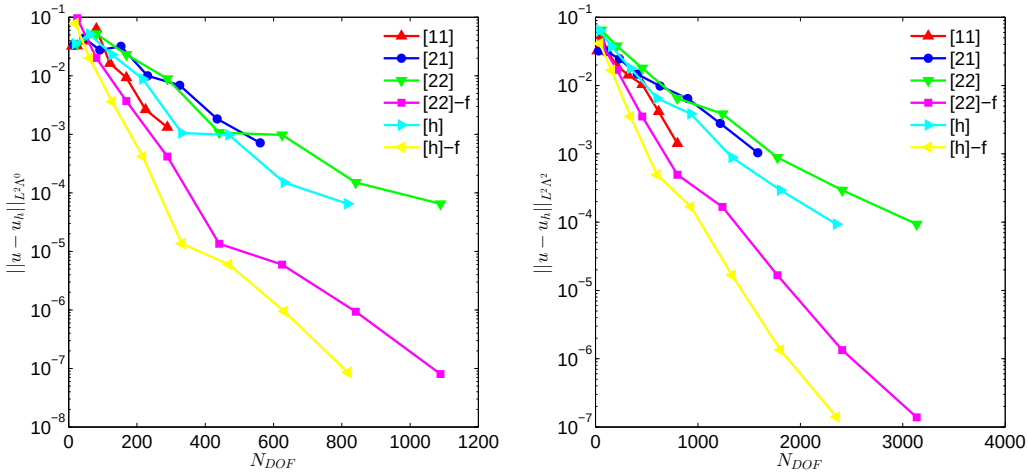


(a) Peak value. **(b)** Peak error.

Figure 7.28 – Peak value and error convergence plotted as a function of the degrees of freedom for h-refinement test case 1 (2-forms).

compared to the p-refinement approach. The difference between meshes [22] and [22]-f is caused by the fact that in case of the former the mass matrix is determined only once and reused after that. This is not done when elements are resized, which requires different mappings for each.

For completeness, the total error sum (element errors summed up) for both problems is given in Figure 7.29. In principle the same conclusion can be drawn from this plot. In the end, this



(a) 0-form. **(b)** 2-form.

Figure 7.29 – Total error convergence as a function of the number of degrees of freedom for h-refinement test case 1 (0-form and 2-form) on various meshes.

particular test case is not as convincing as it was for p-refinement. Nevertheless, improvements were visible. In order to better assess its performance, the current implementation should be expanded to allow for test problems more suited for h-refinement (ones that exhibit non-smooth behavior). Right now however, the code is limited to the 3-element mesh depicted here.

CHAPTER 8

CONCLUSION AND RECOMMENDATIONS

This chapter contains the main conclusions as well as a number of recommendations for future work.

8.1 Conclusion

The main goal of this thesis is to introduce mesh refinement in the MSEM. This should, amongst others, provide for increased flexibility, accuracy and performance. Moreover, it should adhere to theory of mimetic methods as presented in Chapters 2, 3 and 4. The selected operator is the scalar Laplacian, giving rise to the 0-form and 2-form Poisson problem.

It can be concluded that the main goal has been achieved, as the results from Chapter 7 have shown. This was accomplished by closely looking at an existing method, the MEM. Starting with p-refinement, the Poisson problem for 0-forms could be readily solved as it relies on nodal coupling as well (see Section 6.2.1). Save some notational differences, it is found to be largely similar to the original MEM. However, for the 2-form Poisson problem the approach is certainly different as 0-forms (or nodes) do not appear in this type of problem. Therefore an alternate mortar method is formulated which makes use of the 1-forms (see Section 6.2.2). For h-refinement, similar discretizations applied to the 0-form and 2-form Poisson problem were given in Sections 6.3.1 and 6.3.2 respectively.

As mentioned, the final results are presented in Chapter 7. Especially for p-refinement they look promising. In total three test cases were used to assess its performance. Of test case 1 (which is limited in size as the computational domain consists solely of two elements), Figures 7.3 (0-form) and 7.5 (2-form) show a decrease in the required number of degrees of freedom for a given error tolerance. However, the effect is not always profound. It has also been shown that there are limits to the error convergence when increasing the error in one element and keeping the order in its neighbor fixed, as depicted clearly in Figure 7.4 (0-form) and Figure 7.6 (2-form). Additionally, the condition number appears to be only slightly affected (i.e. it increased) by the mortaring procedure, as visualized in Figure 7.7 (0-form) and Figure 7.8 (2-form). The same holds true for its matrix structure (Figure 7.9) and sparsity (Figure 7.10).

The second test case, which is slightly larger in scope and features a peak with a large base

(i.e. it covers multiple elements), mirrored much of these observations although in Figure 7.14 it appeared to favor the 2-form approach. Additionally, visualization and comparison of the individual element errors on a uniform and refined mesh were clearly depicted in Figures 7.15 (0-form) and 7.16 (2-form).

The final test case for p-refinement showed the best results. This test case was aimed at accurately determining the maximum value of a specific peak residing within a single element. Figures 7.22 (0-form) and 7.23 (2-form) reveal a significant decrease in the required number of degrees of freedom for a given error tolerance. Also performance-wise the comparison in Table 7.3 leaves little to be desired, showing a significant speed-up in the required computation time.

For testing the h-refinement implementation the third test case for p-refinement was repeated. Even though for the mesh configuration considered this test case is not optimal, results looked promising. Given the error convergence as depicted in Figure 7.28 (0-form) and Figure 7.28 (2-form), the trend is to decrease the error for a given number of degrees of freedom. However, it required additional resizing of the elements for the h-refinement method to come out better than the single element mesh configuration.

In conclusion, the requirements set at the start of this thesis have been met. That is, improving accuracy while reducing the required number of degrees of freedom while at the same time adhering to mimetic framework. The results are promising, showing increased accuracy as well as performance. There is however room for improvement which justifies the next section which contains a number of recommendations.

8.2 Recommendations

Although promising results were shown, some recommendations are still in place. These are summarized below:

1. **Solve the Poisson problem for 1-forms on a (h- or p-) refined mesh.** From the start this thesis was limited to the scalar Laplacian, which in differential geometry corresponds to the 0-form and 2-form Poisson problem. The next step would be to derive a similar method for 1-forms or the vector Laplacian. It is predicted, given that the mimetic mortar approach presented in this thesis works with both 0-forms and 1-forms, that all the necessary ingredients might already be available.

2. **Combine h- and p-refinement in a true hp-refinement approach.** Unlike the code for p-refinement, the one for h-refinement first needs to be generalized. Once that has been done, both should be combined increasing both flexibility and performance.

3. **Implement a fully hp-adaptive method.** For the test cases presented in this work each refined mesh is configured manually. It would be better to complement it with an adaptive algorithm to take care of this task automatically. This would require a suitable modification strategy combined with error estimators and a stopping criterion. Currently work is already being done one error estimators in the MSEM framework.

4. **Add an extra dimension.** Whether spatial or temporal, adding another dimension would greatly expand the possibilities of the method. With new test problems coming within reach, its use becomes more applied resembling more closely applications encountered in industry.

5. **Confirm mimetic properties.** Even though the method presented in this thesis has been shown to work, it is considered well worth the effort to take a closer look at whether

or not it violates any of the nice properties of the MSEM. If some of them are lost, they should be corrected first.

BIBLIOGRAPHY

[1] G. Anagnostou, Y. Maday, C. Mavriplis, and A. Patera. On the mortar element method: Generalizations and implementation. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 157–173. SIAM, 1989.

[2] P. Bochev. A discourse on variational and geometric aspects of stability of discretizations. In H. Deconinck, editor, *33rd Computational Fluid Dynamics Lecture Series*, VKI LS 2003-05, Chaussee de Waterloo, 72, B-1640 Rhode Saint Genese, Belgium, 2003. Von Karman Institute for Fluid Dynamics. 90 pages.

[3] P. Bochev and J. Hyman. Principles of mimetic discretizations of differential operators. In D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, and M. Shashkov, editors, *Compatible Spatial Discretizations*, volume 142 of *The IMA Volumes in Mathematics and its Applications*, pages 89–119. Springer New York, 2006.

[4] A. Bossavit. On the geometry of electromagnetism. *Journal of Japanese Society of Applied Electromagnetics and Mechanics*, 6:17–28, 1998.

[5] A. Bossavit. Computational electromagnetism and geometry: Building a finite-dimensional" maxwell's house"(1): Network equations. *Journal of Japanese Society of Applied Electromagnetics and Mechanics*, 7(2):150–159, 1999.

[6] M. Bouman. Mimetic spectral element method for elliptic problems. Master's thesis, Delft University of Technology, 2010.

[7] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Fundamentals in Single Domains*, volume 21 of *Scientific Computation*. Springer, 2006.

[8] M. Desbrun, A. Hirani, M. Leok, and J. Marsden. Discrete exterior calculus. *Arxiv preprint math/0508341*, 2005.

[9] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta numerica*, 4:105–158, 1995.

[10] H. Flanders. *Differential Forms with Applications to the Physical Sciences*. Dover Publications, Inc., 1989.

[11] T. Frankel. *The Geometry of Physics: An Introduction*. Cambridge University Press, 2nd edition, 2004.

[12] M. Gerritsma. Edge functions for spectral element methods. In J. S. Hesthaven and E. M. Rnquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76 of *Lecture Notes in Computational Science and Engineering*, pages 199–207. Springer Berlin Heidelberg, 2011.

[13] W. Gui and I. Babuška. The h, p and h-p versions of the finite element method in 1 dimension (p1). *Numerische Mathematik*, 49:577–612, 1986. 10.1007/BF01389733.

[14] W. Gui and I. Babuška. The h, p and h-p versions of the finite element method in 1 dimension (p2). *Numerische Mathematik*, 49:613–657, 1986. 10.1007/BF01389734.

[15] W. Gui and I. Babuška. The h, p and h-p versions of the finite element method in 1 dimension (p3). *Numerische Mathematik*, 49:659–683, 1986. 10.1007/BF01389735.

[16] R. Hiemstra. Mimetic isogeometric discretization method: Applied geometry in cfd. Master's thesis, Delft University of Technology, 2011.

[17] R. Hiptmair. Discrete hodge-operators: An algebraic perspective. *Journal of Electromagnetic Waves and Applications*, 32(3):247–269, 2001.

[18] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp element methods for CFD*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2005.

[19] J. Kreeft and M. Gerritsma. Mixed mimetic spectral element method for stokes flow: A pointwise divergence-free solution. *Arxiv preprint arXiv:1201.4409v2*, 2012.

[20] J. Kreeft and M. Gerritsma. A priori error estimates for compatible spectral discretization of the stokes problem for all admissible boundary conditions. *Arxiv preprint arXiv:1206.2812*, 2012.

[21] J. Kreeft, A. Palha, and M. Gerritsma. Mimetic spectral element method for generalized convection-diffusion problems. *V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*, 2010.

[22] J. Kreeft, A. Palha, and M. Gerritsma. Mimetic framework on curvilinear quadrilaterals of arbitrary order. *Arxiv preprint arXiv:1111.4304*, 2011.

[23] Y. Maday, C. Mavriplis, and A. T. Patera. Non-conforming mortar element methods: Application to spectral discretizations. In *Domain Decomposition Methods*, pages 392–418. SIAM, 1989.

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

[25] C. Mavriplis. *Nonconforming discretizations and a posteriori error estimators for adaptive spectral element techniques*. PhD thesis, Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics., 1989.

[26] P. Nithiarasu and O. Zienkiewicz. Adaptive mesh generation for fluid mechanics problems. *International Journal for Numerical Methods in Engineering*, 47(1-3):629–662, 2000.

[27] G. Oud. Discrete differential geometry. Master's thesis, Delft University of Technology, 2011.

[28] A. Palha and M. Gerritsma. Mimetic least-squares spectral/hp finite element method for the poisson equation. In I. Lirkov, S. Margenov, and J. Wasniewski, editors, *Large-Scale Scientific Computing*, volume 5910 of *Lecture Notes in Computer Science*, pages 662–670. Springer Berlin / Heidelberg, 2010.

[29] A. Patera, G. Anagnostou, et al. *Nonconforming sliding spectral element methods for the unsteady incompressible Navier-Stokes equations*. PhD thesis, Massachusetts Institute of Technology, 1990.

[30] P. P. Rebelo. Physically accurate advection: A discrete representation of the lie derivative. Master's thesis, Delft University of Technology, 2011.

[31] N. Robidoux. Polynomial histopolation, superconvergent degrees of freedom, and pseudospectral discrete hodge operators. *Unpublished*, 2003.

[32] E. Tonti. *On the Formal Structure of Physical Theories*. Istituto de matematica, Politecnico, 1975. monograph of the Italian National Research Council.

Appendices

APPENDIX A

DISCRETIZATIONS OF THE SCALAR POISSON PROBLEM

This appendix contains the derivations for the 0-form and 2-form Poisson problem presented in Chapter 5. They are meant to provide additional insight into the discretization.

A.1 0-form Poisson problem

Section 5.2 stopped at:

$$(\mathrm{d}v^{(0)}, \mathrm{d}u^{(0)}) = (v^{(0)}, f^{(0)}). \tag{A.1}$$

Where the left- and right-hand side could be expanded as:

$$\left(\mathrm{d}v^{(0)}, \mathrm{d}u^{(0)}\right)_\Omega = \int_\Omega \mathrm{d}v^{(0)} \wedge \star \mathrm{d}u^{(0)}, \tag{A.2}$$

$$\left(v^{(0)}, f^{(0)}\right)_\Omega = \int_\Omega v^{(0)} \wedge \star f^{(0)}. \tag{A.3}$$

In order to map these expressions back to the standard element $\Omega_0 : \xi, \eta \in (-1,1) \times (-1,1)$, the pullback is applied. The inner product from equation (A.2) is expanded as follows:

$$\begin{aligned}
&\int_\Omega \mathrm{d}v^{(0)} \wedge \star \mathrm{d}u^{(0)} = \int_\Omega \left(\frac{\mathrm{d}v}{\mathrm{d}x}\mathrm{d}x + \frac{\mathrm{d}v}{\mathrm{d}y}\mathrm{d}y\right) \wedge \left(\frac{\mathrm{d}u}{\mathrm{d}x}\mathrm{d}y - \frac{\mathrm{d}u}{\mathrm{d}y}\mathrm{d}x\right) = \\
&\int_{\Omega_0} \left(\frac{1}{J}\begin{bmatrix} v_\xi \\ v_\eta \end{bmatrix}^T \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{bmatrix}\right) \left(\frac{1}{J}\begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \begin{bmatrix} u_\xi \\ u_\eta \end{bmatrix}\right) J \mathrm{d}\xi \mathrm{d}\eta.
\end{aligned} \tag{A.4}$$

First of all note the appearance of the term J. This represents the Jacobian and is defined as:

$$J = \frac{\mathrm{d}x}{\mathrm{d}\xi}\frac{\mathrm{d}y}{\mathrm{d}\eta} - \frac{\mathrm{d}x}{\mathrm{d}\eta}\frac{\mathrm{d}y}{\mathrm{d}\xi}. \tag{A.5}$$

Additionally, note as well the minus sign in the second step due to the application of the Hodge $\star$ operator. It cancels at the last step due to the interchange of $\mathrm{d}y\mathrm{d}x$ to $\mathrm{d}x\mathrm{d}y$ (anticommutativity

property of the wedge product). Additionally, terms containing $\mathrm{d}x\mathrm{d}x$ or $\mathrm{d}y\mathrm{d}y$ vanish as described in Chapter 2. Equation (A.3) is evaluated in a similar fashion:

$$\int_\Omega v^{(0)} \wedge \star f^{(0)} = \int_\Omega v \wedge (f \mathrm{d}x\mathrm{d}y) = \int_{\Omega_0} v f J \mathrm{d}\xi\mathrm{d}\eta. \tag{A.6}$$

The next step is the discretization of the individual terms. This is done using the familiar edge and Lagrange polynomials (see Chapter 4) evaluated in the $(N+1)$ GLL nodes:

$$(v^{(0)})_h = \sum_{i=0}^{N}\sum_{j=0}^{N} v_i h_i(\xi) h_j(\eta), \tag{A.7}$$

$$(f^{(0)})_h = \sum_{i=0}^{N}\sum_{j=0}^{N} f_i h_i(\xi) h_j(\eta), \tag{A.8}$$

$$\begin{aligned}(\mathrm{d}v^{(0)})_h = &\sum_{i=1}^{N}\sum_{j=0}^{N} (v^\xi_{ij} - v^\xi_{i-1,j}) e_i(\xi) h_j(\eta) + \\ &\sum_{i=0}^{N}\sum_{j=1}^{N} (v^\eta_{ij} - v^\eta_{i,j-1}) h_i(\xi) e_j(\eta),\end{aligned} \tag{A.9}$$

$$\begin{aligned}(\mathrm{d}u^{(0)})_h = &\sum_{i=1}^{N}\sum_{j=0}^{N} (u^\xi_{ij} - u^\xi_{i-1,j}) e_i(\xi) h_j(\eta) + \\ &\sum_{i=0}^{N}\sum_{j=1}^{N} (u^\eta_{ij} - u^\eta_{i,j-1}) h_i(\xi) e_j(\eta).\end{aligned} \tag{A.10}$$

Note the superscript h which denotes a discretization parameter. Using these expressions as well as the definition of the inner product, equation (A.1) can be evaluated numerically. Further elaborations will be restricted to affine mappings only, as they are the only types of mappings considered in the test cases. Additionally, in order to reduce the size of the expression some notational simplifications are introduced first:

$$(v^{(0)})_h = \sum_{i=0}^{N}\sum_{j=0}^{N} v_{ij} R^{0,0}_{ij} \tag{A.11}$$

$$(f^{(0)})_h = \sum_{i=0}^{N}\sum_{j=0}^{N} f_{ij} R^{0,0}_{ij} \tag{A.12}$$

$$(\mathrm{d}v^{(0)})_h = \sum_{i=1}^{N}\sum_{j=0}^{N} \bar{v}^\xi_{ij} R^{1,0}_{ij} + \sum_{i=0}^{N}\sum_{j=1}^{N} \bar{v}^\eta_{ij} R^{0,1}_{ij} \tag{A.13}$$

$$(\mathrm{d}u^{(0)})_h = \sum_{i=1}^{N}\sum_{j=0}^{N} \bar{u}^\xi_{ij} R^{1,0}_{ij} + \sum_{i=0}^{N}\sum_{j=1}^{N} \bar{u}^\eta_{ij} R^{0,1}_{ij} \tag{A.14}$$

Here the overbar denotes a difference while R represents a combination of the interpolating edge

and Lagrange polynomials (indicated with the superscripts 1 and 0 respectively):

$$R_{ij}^{0,0} = h_i(\xi)h_j(\eta), \tag{A.15}$$
$$R_{ij}^{0,1} = h_i(\xi)e_j(\eta), \tag{A.16}$$
$$R_{ij}^{1,0} = e_i(\xi)h_j(\eta). \tag{A.17}$$

Substitution of these expressions into equation (A.1) yields:

$$\int_{\Omega_0} \left(\sum_{i=1}^{N}\sum_{j=0}^{N} \bar{v}_{ij}^{\xi} R_{ij}^{1,0} \sum_{i=1}^{N}\sum_{j=0}^{N} \bar{u}_{ij}^{\xi} R_{ij}^{1,0} \right) \left(\frac{\mathrm{d}y}{\mathrm{d}\eta}\right)^2 \frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta +$$
$$\int_{\Omega_0} \left(\sum_{i=0}^{N}\sum_{j=1}^{N} \bar{v}_{ij}^{\eta} R_{ij}^{0,1} \sum_{i=0}^{N}\sum_{j=1}^{N} \bar{u}_{ij}^{\eta} R_{ij}^{0,1} \right) \left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2 \frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta =$$
$$\int_{\Omega_0} \left(\sum_{i=0}^{N}\sum_{j=0}^{N} v_{ij} R_{ij}^{0,0} \sum_{i=0}^{N}\sum_{j=0}^{N} f_{ij} R_{ij}^{0,0} \right) J\mathrm{d}\xi\mathrm{d}\eta$$

Evaluating the integrals requires numerical integration or quadrature. What follows is a short intermezzo on how to perform quadrature, in particular Gaussian quadrature which is aimed at integrating polynomials.

***

Numerical integration (or quadrature) allows for an automated way of evaluating integrals. Take for example the following integral:

$$\int_{-1}^{1} u(\xi)\mathrm{d}\xi. \tag{A.18}$$

The concept of quadrature is to approximate an integral by a finite summation:

$$\int_{-1}^{1} u(\xi)\mathrm{d}\xi \approx \sum_{i=0}^{N} \omega_i u(\xi_i). \tag{A.19}$$

Here ω_i are the integration weights and ξ_i are the $(N+1)$ quadrature nodes on the standard domain $\Omega_0 = (-1, 1)$. Although the type of quadrature nodes can vary, in this work the GLL nodes are used. Now consider again the function $u(\xi)$ but represented with the use of Lagrange polynomials:

$$u(\xi) = \sum_{i=0}^{N} u(\xi_i)h_i(\xi) + \epsilon(u). \tag{A.20}$$

Here $\epsilon(u)$ represents an approximation term which arises as a result of the finite summation. Integration then yields:

$$\int_{-1}^{1} u(\xi)\mathrm{d}x = \sum_{i=0}^{N} \omega_i u(\xi_i) + R(u). \tag{A.21}$$

With:

$$\omega_i = \int_{-1}^{1} h_i(\xi)\mathrm{d}\xi, \tag{A.22}$$

$$R(u) = \int_{-1}^{1} \epsilon(u)\mathrm{d}\xi. \tag{A.23}$$

The integration weights associated with the GLL nodes (as defined by equation (A.22)) are given by the following expression:

$$\omega_i = \frac{2}{N(N+1)[L_N(\xi_i)]^2},\ i = 0, \ldots, N. \tag{A.24}$$

Note that the numerical integral is exact (i.e. $R(u) = 0$) whenever $u(\xi) \in \mathbb{P}_{2N-1}([-1, 1])$.

Naturally, numerical integration can be extended to higher dimensions as well. To that end, consider the function $u(\xi, \eta)$ which must be integrated:

$$\int_{-1}^{1}\int_{-1}^{1} u(\xi, \eta)\mathrm{d}\xi\mathrm{d}\eta. \tag{A.25}$$

Once more approximate $u(\xi, \eta)$ with Lagrange interpolants:

$$u(\xi, \eta) \approx \sum_{i=0}^{N}\sum_{j=0}^{M} u(\xi_i, \eta_j)h_i(\xi_i)h_j(\eta_j). \tag{A.26}$$

Which after integrating yields:

$$\int_{-1}^{1}\int_{-1}^{1} u(\xi, \eta)\mathrm{d}\xi\mathrm{d}\eta \approx \sum_{i=0}^{N}\sum_{j=0}^{M} u(\xi_i, \eta_j)\omega_i\omega_j. \tag{A.27}$$

Note that the order of integration can be chosen differently in either direction (which is not done in this thesis).

This ends the intermezzo on quadrature.

***

Applying quadrature allows these integrals to be evaluated numerically resulting in further simplifications:

$$\sum_{i=1}^{N}\sum_{j=0}^{N} \bar{v}_{ij}^{\xi} \sum_{p=1}^{N}\sum_{q=0}^{N} \bar{u}_{pq}^{\xi} \sum_{r=0}^{N}\sum_{s=0}^{N} e_i(\xi_r)e_p(\xi_r)\omega_r\omega_s \left(\frac{\mathrm{d}y}{\mathrm{d}\eta}\right)^2_{\substack{\xi=\xi_r \\ \eta=\eta_s}} \left(\frac{1}{J}\right)_{\substack{\xi=\xi_r \\ \eta=\eta_s}} +$$
$$\sum_{i=0}^{N}\sum_{j=1}^{N} \bar{v}_{ij}^{\eta} \sum_{p=0}^{N}\sum_{q=1}^{N} \bar{u}_{pq}^{\eta} \sum_{r=0}^{N}\sum_{s=0}^{N} \omega_r\omega_s e_j(\eta_s)e_q(\eta_s) \left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2_{\substack{\xi=\xi_r \\ \eta=\eta_s}} \left(\frac{1}{J}\right)_{\substack{\xi=\xi_r \\ \eta=\eta_s}} =$$
$$\sum_{i=0}^{N}\sum_{j=0}^{N} v_{ij}f_{ij}\omega_i\omega_j J_{\substack{\xi=\xi_r \\ \eta=\eta_s}}.$$

For an affine mapping, as considered here, the derivatives and Jacobians are simply constants. Therefore the subscripts can be left out resulting in the expression:

$$\sum_{i=1}^{N}\sum_{j=0}^{N}\bar{v}_{ij}^{\xi}\sum_{p=1}^{N}\sum_{q=0}^{N}\bar{u}_{pq}^{\xi}\sum_{r=0}^{N}\sum_{s=0}^{N}e_i(\xi_r)e_p(\xi_r)\omega_r\omega_s\left(\frac{\mathrm{d}y}{\mathrm{d}\eta}\right)^2\left(\frac{1}{J}\right)+ \tag{A.28}$$

$$\sum_{i=0}^{N}\sum_{j=1}^{N}\bar{v}_{ij}^{\eta}\sum_{p=0}^{N}\sum_{q=1}^{N}\bar{u}_{pq}^{\eta}\sum_{r=0}^{N}\sum_{s=0}^{N}e_j(\eta_s)e_q(\eta_s)\omega_r\omega_s\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2\left(\frac{1}{J}\right)= \tag{A.29}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N}v_{ij}f_{ij}\omega_i\omega_j J. \tag{A.30}$$

The final step consists of writing this in a suitable matrix-vector format:

$$(\mathbb{D}^{\xi}\mathbf{v})^T M_1(\mathbb{D}^{\xi}\mathbf{u})+(\mathbb{D}^{\eta}\mathbf{v})^T M_1(\mathbb{D}^{\eta}\mathbf{u})=\mathbf{v}^T M_0\mathbf{f}. \tag{A.31}$$

Derivatives and Jacobians are contained within the mass matrices M_0 and M_1 (which are the Kronecker products of the coefficient matrices of the integrated Lagrange and edge polynomials respectively). Also note the superscripts on the incidence matrices $\mathbb{D}^{\xi}$ and $\mathbb{D}^{\eta}$ (as seen in Chapter 3). These denote the connections in ξ and η directions. Since this expression should hold for all possible test functions they can be canceled which yields:

$$((\mathbb{D}^{\xi})^T M_1\mathbb{D}^{\xi}+(\mathbb{D}^{\eta})^T M_1\mathbb{D}^{\eta})\mathbf{u}=M_0\mathbf{f}. \tag{A.32}$$

Which in turn simplifies to:

$$\mathbb{D}^T M_1\mathbb{D}\mathbf{u}=M_0\mathbf{f}. \tag{A.33}$$

Where the incidence matrix $\mathbb{D}$ is given by:

$$\mathbb{D}=\begin{bmatrix}\mathbb{D}^{\xi} & \mathbb{D}^{\eta}\end{bmatrix}. \tag{A.34}$$

A.2 2-form Poisson problem

Section 5.3 stopped at:

$$\left(v^{(2)},\mathrm{d}q^{(1)}\right)_{\Omega}=\left(v^{(2)},f^{(2)}\right)_{\Omega}, \tag{A.35}$$

$$\left(\mathrm{d}\tau^{(1)},u^{(2)}\right)_{\Omega}=\left(\tau^{(1)},q^{(1)}\right)_{\Omega}. \tag{A.36}$$

Where the left- and right-hand side of the first equation could be expanded as:

$$\left(v^{(2)},\mathrm{d}q^{(1)}\right)_{\Omega}=\int_{\Omega}v^{(2)}\wedge\star q^{(1)}, \tag{A.37}$$

$$\left(v^{(2)},f^{(2)}\right)_{\Omega}=\int_{\Omega}v^{(2)}\wedge\star f^{(2)}. \tag{A.38}$$

Applying the pullback to map everything to the standard element $\Omega_0 : \xi, \eta \in (-1,1) \times (-1,1)$ yields:

$$\begin{aligned}\int_\Omega v^{(2)} \wedge \star \mathrm{d}q^{(1)} &= \int_\Omega v \left(\frac{\mathrm{d}q^y}{\mathrm{d}x} - \frac{\mathrm{d}q^x}{\mathrm{d}y} \right) \mathrm{d}x\mathrm{d}y \\ &= \int_{\Omega_0} \left(\frac{1}{J} v \right) \left(\frac{1}{J} (q^y_\xi - q^x_\eta) \right) J \mathrm{d}\xi\mathrm{d}\eta, \end{aligned} \tag{A.39}$$

$$\int_\Omega v^{(2)} \wedge \star f^{(2)} = \int_\Omega v f \mathrm{d}x\mathrm{d}y = \int_{\Omega_0} \left(\frac{1}{J} v \right) \left(\frac{1}{J} f \right) J \mathrm{d}\xi\mathrm{d}\eta. \tag{A.40}$$

Similarly, the inner-products of equation (A.36) are expanded as:

$$\left(\mathrm{d}\tau^{(1)}, u^{(2)} \right)_\Omega = \int_\Omega \mathrm{d}\tau^{(1)} \wedge \star u^{(2)}, \tag{A.41}$$

$$\left(\tau^{(1)}, q^{(1)} \right)_\Omega = \int_\Omega \tau^{(1)} \wedge \star q^{(1)}. \tag{A.42}$$

Application of the pullback operator yields:

$$\begin{aligned}\int_\Omega \mathrm{d}\tau^{(1)} \wedge \star u^{(2)} &= \int_\Omega \left(\frac{\mathrm{d}\tau^y}{\mathrm{d}x} - \frac{\mathrm{d}\tau^x}{\mathrm{d}y} \right) u \mathrm{d}x\mathrm{d}y \\ &= \int_{\Omega_0} \left(\frac{1}{J} [\tau^y_\xi - \tau^x_\eta] \right) \left(\frac{1}{J} [u] \right) J \mathrm{d}\xi\mathrm{d}\eta, \end{aligned} \tag{A.43}$$

$$\begin{aligned}&\int_\Omega \tau^{(1)} \wedge \star q^{(1)} = \int_\Omega \left(\tau^x q^x + \tau^y q^y \right) \mathrm{d}x\mathrm{d}y = \\ &\int_{\Omega_0} \left(\frac{1}{J} \begin{bmatrix} \tau^x \\ \tau^y \end{bmatrix}^T \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{bmatrix} \right) \left(\frac{1}{J} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \begin{bmatrix} q^x \\ q^y \end{bmatrix} \right) J \mathrm{d}\xi\mathrm{d}\eta. \end{aligned} \tag{A.44}$$

Discretization of the various terms yields:

$$(v^{(2)})_h = \sum_{i=1}^{N} \sum_{j=1}^{N} v_{ij} e_i(\xi) e_j(\eta), \tag{A.45}$$

$$(f^{(2)})_h = \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij} e_i(\xi) e_j(\eta), \tag{A.46}$$

$$(u^{(2)})_h = \sum_{i=1}^{N} \sum_{j=1}^{N} u_{ij} e_i(\xi) e_j(\eta), \tag{A.47}$$

$$(q^{(1)})_h = \sum_{i=1}^{N} \sum_{j=0}^{N} q^\xi_{ij} e_i(\xi) h_j(\eta) + \sum_{i=0}^{N} \sum_{j=1}^{N} q^\eta_{ij} h_i(\xi) e_j(\eta), \tag{A.48}$$

$$(\tau^{(1)})_h = \sum_{i=1}^{N} \sum_{j=0}^{N} \tau^\xi_{ij} e_i(\xi) h_j(\eta) + \sum_{i=0}^{N} \sum_{j=1}^{N} \tau^\eta_{ij} h_i(\xi) e_j(\eta), \tag{A.49}$$

$$
\begin{aligned}
(\mathrm{d}q^{(1)})_h = &\sum_{i=1}^{N}\sum_{j=1}^{N}\left(q_{ij}^{\xi} - q_{i-1,j}^{\xi}\right) e_i(\xi)e_j(\eta) - \\
&\sum_{i=1}^{N}\sum_{j=1}^{N}\left(q_{ij}^{\eta} - q_{i,j-1}^{\eta}\right) e_i(\xi)e_j(\eta),
\end{aligned} \tag{A.50}
$$

$$
\begin{aligned}
(\mathrm{d}\tau^{(1)})_h = &\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\tau_{ij}^{\xi} - \tau_{i-1,j}^{\xi}\right) e_i(\xi)e_j(\eta) - \\
&\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\tau_{ij}^{\eta} - \tau_{i,j-1}^{\eta}\right) e_i(\xi)e_j(\eta).
\end{aligned} \tag{A.51}
$$

Once more some notational simplifications can be introduced:

$$(v^{(2)})_h = \sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij} R_{ij}^{1,1}, \tag{A.52}$$

$$(f^{(2)})_h = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij} R_{ij}^{1,1}, \tag{A.53}$$

$$(u^{(2)})_h = \sum_{i=1}^{N}\sum_{j=1}^{N} u_{ij} R_{ij}^{1,1}, \tag{A.54}$$

$$(q^{(1)})_h = \sum_{i=1}^{N}\sum_{j=0}^{N} q_{ij}^{\xi} R_{ij}^{1,0} + \sum_{i=0}^{N}\sum_{j=1}^{N} q_{ij}^{\eta} R_{ij}^{0,1}, \tag{A.55}$$

$$(\tau^{(1)})_h = \sum_{i=1}^{N}\sum_{j=0}^{N} \tau_{ij}^{\xi} R_{ij}^{1,0} + \sum_{i=0}^{N}\sum_{j=1}^{N} \tau_{ij}^{\eta} R_{ij}^{0,1}, \tag{A.56}$$

$$(\mathrm{d}q^{(1)})_h = \sum_{i=1}^{N}\sum_{j=1}^{N} \bar{q}_{ij}^{\xi} R_{ij}^{1,1} - \sum_{i=1}^{N}\sum_{j=1}^{N} \bar{q}_{ij}^{\eta} R_{ij}^{1,1}, \tag{A.57}$$

$$(\mathrm{d}\tau^{(1)})_h = \sum_{i=1}^{N}\sum_{j=1}^{N} \bar{\tau}_{ij}^{\xi} R_{ij}^{1,1} - \sum_{i=1}^{N}\sum_{j=1}^{N} \bar{\tau}_{ij}^{\eta} R_{ij}^{1,1}. \tag{A.58}$$

With $R_{ij}^{1,1}$ defined as:

$$R_{ij}^{1,1} = e_i(\xi)e_j(\eta). \tag{A.59}$$

Substitution of these expressions into equation (A.35) yields:

$$
\begin{aligned}
&\int_{\Omega_0}\left(\sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij} R_{ij}^{1,1} \sum_{i=1}^{N}\sum_{j=1}^{N} \bar{q}_{ij}^{\xi} R_{ij}^{1,1}\right)\frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta - \\
&\int_{\Omega_0}\left(\sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij} R_{ij}^{1,1} \sum_{i=1}^{N}\sum_{j=1}^{N} \bar{q}_{ij}^{\eta} R_{ij}^{1,1}\right)\frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta = \\
&\int_{\Omega_0}\left(\sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij} R_{ij}^{1,1} \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij} R_{ij}^{1,1}\right)\frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta.
\end{aligned} \tag{A.60}
$$

And into equation (A.36):

$$\begin{aligned}
&\int_{\Omega_0}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\bar{\tau}_{ij}^{\xi}R_{ij}^{1,1}\sum_{i=1}^{N}\sum_{j=1}^{N}u_{ij}R_{ij}^{1,1}\right)\frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta-\\
&\int_{\Omega_0}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\bar{\tau}_{ij}^{\eta}R_{ij}^{1,1}\sum_{i=1}^{N}\sum_{j=1}^{N}u_{ij}R_{ij}^{1,1}\right)\frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta=\\
&\int_{\Omega_0}\left(\sum_{i=1}^{N}\sum_{j=0}^{N}\tau_{ij}^{\xi}R_{ij}^{1,0}\sum_{i=1}^{N}\sum_{j=1}^{N}\bar{q}_{ij}^{\xi}R_{ij}^{1,1}\right)\left(\frac{\mathrm{d}y}{\mathrm{d}\eta}\right)^2\frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta+\\
&\int_{\Omega_0}\left(\sum_{i=1}^{N}\sum_{j=0}^{N}\tau_{ij}^{\eta}R_{ij}^{1,0}\sum_{i=1}^{N}\sum_{j=1}^{N}\bar{q}_{ij}^{\eta}R_{ij}^{1,1}\right)\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2\frac{1}{J}\mathrm{d}\xi\mathrm{d}\eta.
\end{aligned}\tag{A.61}$$

Next perform numerical integration:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N}v_{ij}\sum_{m=1}^{N}\sum_{n=1}^{N}\bar{q}_{mn}^{\xi}\sum_{r=0}^{N}\sum_{s=0}^{N}e_i(\xi_r)e_m(\xi_r)e_j(\eta_s)e_n(\eta_s)\omega_r\omega_s\left(\frac{1}{J}\right)_{\substack{\xi=\xi_r\\\eta=\eta_s}}-\\
&\sum_{i=1}^{N}\sum_{j=1}^{N}v_{ij}\sum_{m=1}^{N}\sum_{n=1}^{N}\bar{q}_{mn}^{\eta}\sum_{r=0}^{N}\sum_{s=0}^{N}e_i(\xi_r)e_m(\xi_r)e_j(\eta_s)e_n(\eta_s)\omega_r\omega_s\left(\frac{1}{J}\right)_{\substack{\xi=\xi_r\\\eta=\eta_s}}=\\
&\sum_{i=1}^{N}\sum_{j=1}^{N}v_{ij}\sum_{m=1}^{N}\sum_{n=1}^{N}f_{mn}\sum_{r=0}^{N}\sum_{s=0}^{N}e_i(\xi_r)e_m(\xi_r)e_j(\eta_s)e_n(\eta_s)\omega_r\omega_s\left(\frac{1}{J}\right)_{\substack{\xi=\xi_r\\\eta=\eta_s}}.
\end{aligned}\tag{A.62}$$

And for the second equation:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N}\bar{\tau}_{ij}^{\xi}\sum_{m=1}^{N}\sum_{n=1}^{N}u_{mn}\sum_{r=0}^{N}\sum_{s=0}^{N}e_i(\xi_r)e_m(\xi_r)e_j(\eta_s)e_n(\eta_s)\omega_r\omega_s\left(\frac{1}{J}\right)_{\substack{\xi=\xi_r\\\eta=\eta_s}}-\\
&\sum_{i=1}^{N}\sum_{j=1}^{N}\bar{\tau}_{ij}^{\eta}\sum_{m=1}^{N}\sum_{n=1}^{N}u_{mn}\sum_{r=0}^{N}\sum_{s=0}^{N}e_i(\xi_r)e_m(\xi_r)e_j(\eta_s)e_n(\eta_s)\omega_r\omega_s\left(\frac{1}{J}\right)_{\substack{\xi=\xi_r\\\eta=\eta_s}}=\\
&\sum_{i=1}^{N}\sum_{j=0}^{N}\bar{\tau}_{ij}^{\xi}\sum_{m=1}^{N}\sum_{n=0}^{N}\bar{q}_{mn}^{\xi}\sum_{r=0}^{N}\sum_{s=0}^{N}e_i(\xi_r)e_m(\xi_r)\omega_r\omega_s\left(\frac{\mathrm{d}y}{\mathrm{d}\eta}\right)^2_{\substack{\xi=\xi_r\\\eta=\eta_s}}\left(\frac{1}{J}\right)_{\substack{\xi=\xi_r\\\eta=\eta_s}}+\\
&\sum_{i=0}^{N}\sum_{j=1}^{N}\bar{\tau}_{ij}^{\eta}\sum_{m=0}^{N}\sum_{n=1}^{N}\bar{q}_{mn}^{\eta}\sum_{r=0}^{N}\sum_{s=0}^{N}e_j(\eta_s)e_n(\eta_s)\omega_r\omega_s\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2_{\substack{\xi=\xi_r\\\eta=\eta_s}}\left(\frac{1}{J}\right)_{\substack{\xi=\xi_r\\\eta=\eta_s}}.
\end{aligned}\tag{A.63}$$

Once more the subscripts on the Jacobians and derivatives can be dropped as they are simply constants for the affine mappings considered in this work:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij} \sum_{m=1}^{N}\sum_{n=1}^{N} \bar{q}^{\xi}_{mn} \sum_{r=0}^{N}\sum_{s=0}^{N} e_i(\xi_r) e_m(\xi_r) e_j(\eta_s) e_n(\eta_s) \omega_r \omega_s \left(\frac{1}{J}\right) - \\
&\sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij} \sum_{m=1}^{N}\sum_{n=1}^{N} \bar{q}^{\eta}_{mn} \sum_{r=0}^{N}\sum_{s=0}^{N} e_i(\xi_r) e_m(\xi_r) e_j(\eta_s) e_n(\eta_s) \omega_r \omega_s \left(\frac{1}{J}\right) = \\
&\sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij} \sum_{m=1}^{N}\sum_{n=1}^{N} f_{mn} \sum_{r=0}^{N}\sum_{s=0}^{N} e_i(\xi_r) e_m(\xi_r) e_j(\eta_s) e_n(\eta_s) \omega_r \omega_s \left(\frac{1}{J}\right).
\end{aligned} \tag{A.64}$$

And for the second equation:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N} \bar{\tau}^{\xi}_{ij} \sum_{m=1}^{N}\sum_{n=1}^{N} u_{mn} \sum_{r=0}^{N}\sum_{s=0}^{N} e_i(\xi_r) e_m(\xi_r) e_j(\eta_s) e_n(\eta_s) \omega_r \omega_s \left(\frac{1}{J}\right) - \\
&\sum_{i=1}^{N}\sum_{j=1}^{N} \bar{\tau}^{\eta}_{ij} \sum_{m=1}^{N}\sum_{n=1}^{N} u_{mn} \sum_{r=0}^{N}\sum_{s=0}^{N} e_i(\xi_r) e_m(\xi_r) e_j(\eta_s) e_n(\eta_s) \omega_r \omega_s \left(\frac{1}{J}\right) = \\
&\quad \sum_{i=1}^{N}\sum_{j=0}^{N} \bar{\tau}^{\xi}_{ij} \sum_{m=1}^{N}\sum_{n=0}^{N} \bar{q}^{\xi}_{mn} \sum_{r=0}^{N}\sum_{s=0}^{N} e_i(\xi_r) e_m(\xi_r) \omega_r \omega_s \left(\frac{\mathrm{d}y}{\mathrm{d}\eta}\right)^2 \left(\frac{1}{J}\right) + \\
&\quad \sum_{i=0}^{N}\sum_{j=1}^{N} \bar{\tau}^{\eta}_{ij} \sum_{m=0}^{N}\sum_{n=1}^{N} \bar{q}^{\eta}_{mn} \sum_{r=0}^{N}\sum_{s=0}^{N} e_j(\eta_s) e_n(\eta_s) \omega_r \omega_s \left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2 \left(\frac{1}{J}\right).
\end{aligned} \tag{A.65}$$

In matrix-vector format this becomes (having switched around the equations to bring out the symmetry of the operator):

$$\begin{bmatrix} -\begin{bmatrix} \boldsymbol{\tau}^{\xi} M_1 \mathbf{q}^{\xi} & \varnothing \\ \varnothing & \boldsymbol{\tau}^{\eta} M_1 \mathbf{q}^{\eta} \end{bmatrix} & \begin{bmatrix} (\mathbb{D}^{\xi}\boldsymbol{\tau}^{\xi})^T M_2 \mathbf{u} \\ -(\mathbb{D}^{\eta}\boldsymbol{\tau}^{\eta})^T M_2 \mathbf{u} \end{bmatrix} \\ \begin{bmatrix} \mathbf{v} M_2 \mathbb{D}^{\xi} \mathbf{q}^{\xi} & -\mathbf{v} M_2 \mathbb{D}^{\eta} \mathbf{q}^{\eta} \end{bmatrix} & \varnothing \end{bmatrix} = \begin{bmatrix} \varnothing \\ \varnothing \\ \mathbf{v} M_2 \mathbf{f} \end{bmatrix}. \tag{A.66}$$

Once again derivatives and Jacobians are contained within the mass matrices M_1 and M_2. The incidence matrices from Chapter 3 have returned as well. Canceling the test functions yields:

$$\begin{bmatrix} -\begin{bmatrix} M_1 & \varnothing \\ \varnothing & M_1 \end{bmatrix} & \begin{bmatrix} (\mathbb{D}^{\xi})^T M_2 \\ -(\mathbb{D}^{\eta})^T M_2 \end{bmatrix} \\ \begin{bmatrix} M_2 \mathbb{D}^{\xi} & -M_2 \mathbb{D}^{\eta} \end{bmatrix} & \varnothing \end{bmatrix} \begin{bmatrix} \mathbf{q}^{\xi} \\ \mathbf{q}^{\eta} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \varnothing \\ \varnothing \\ M_2 \mathbf{f} \end{bmatrix}. \tag{A.67}$$

The end result becomes:

$$\begin{bmatrix} -\bar{M}_1 & \mathbb{D}^T M_2 \\ \mathbb{D} M_2 & \varnothing \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \varnothing \\ M_2 \mathbf{f} \end{bmatrix}. \tag{A.68}$$

Where the incidence matrix $\mathbb{D}$ and mass matrix $\bar{M}_1$ are defined as:

$$\mathbb{D} = \begin{bmatrix} \mathbb{D}^{\xi} & -\mathbb{D}^{\eta} \end{bmatrix}, \tag{A.69}$$

$$\bar{M}_1 = \begin{bmatrix} M_1 & \varnothing \\ \varnothing & M_1 \end{bmatrix}. \tag{A.70}$$

APPENDIX B

TABLES WITH RESULTS

This chapter contains a number of tables with data from the test cases presented in Chapter 7.

B.1 p-refinement

This section contains the tables corresponding to the p-refinement test cases of Section 7.1.

B.1.1 Test case 1

Table B.1 – Number of degrees of freedom and element order. This table corresponds to p-refinement test case 1, the 0-form Poisson problem.

p	uniform	$p+1$	$p+2$	$p+3$	$p+4$	$p+5$
2	15	21	29	39	51	65
4	45	55	67	81	97	115
6	91	105	121	139	159	181
8	153	171	191	213	237	263
10	231	253	277	303	331	361
12	325	351	379	409	441	475
14	435	465	497	531	567	605
16	561	595	631	669	709	751

Table B.2 – Number of degrees of freedom and element order. This table corresponds to p-refinement test case 1, the 2-form Poisson problem.

p	uniform	$p+1$	$p+2$	$p+3$	$p+4$	$p+5$
2	30	46	68	96	130	170
4	108	136	170	210	256	308
6	234	274	320	372	430	494
8	408	460	518	582	652	728
10	630	694	764	840	922	1010
12	900	976	1058	1146	1240	1340
14	1218	1306	1400	1500	1606	1718
16	1584	1684	1790	1902	2020	2144

B.1.2 Test case 2

Table B.3 – Number of degrees of freedom and element order. This table corresponds to p-refinement test case 2, the 0-form Poisson problem.

p	uniform	refined
2	49	102
4	169	258
6	361	486
8	625	786
10	961	1158

Table B.4 – Number of degrees of freedom and element order. This table corresponds to p-refinement test case 2, the 2-form Poisson problem.

p	uniform	refined
2	120	275
4	456	719
6	1008	1379
8	1776	2255
10	2760	3347

B.1.3 Test case 3

Table B.5 – Number of degrees of freedom and element order. This table corresponds to p-refinement test case 3, the 0-form Poisson problem.

p	uniform	refined
2	49	19
4	169	31
6	361	51
8	625	79
10	961	115
12	1369	159
14	1849	211
16	2401	271

Table B.6 – Number of degrees of freedom and element order. This table corresponds to p-refinement test case 3, the 0-form Poisson problem.

p	uniform	refined
2	120	42
4	456	78
6	1008	138
8	1776	222
10	2760	330
12	3960	462
14	5376	618
16	7008	798

B.2 h-refinement

This section contains the tables corresponding to the h-refinement test case of Section 7.1.

B.2.1 Test case 1

Table B.7 – Number of degrees of freedom and element order. This table corresponds to h-refinement test case 1, the 0-form Poisson problem.

p	[11]	[21]	[22](-f)	[h](-f)
2	9	15	25	19
4	25	45	81	61
6	49	91	169	127
8	81	153	289	217
10	121	231	441	331
12	169	325	625	469
14	225	435	841	631
16	289	561	1089	817

Table B.8 – Number of degrees of freedom and element order. This table corresponds to h-refinement test case 1, the 2-form Poisson problem.

p	[11]	[21]	[22](-f)	[h](-f)
2	16	30	56	42
4	56	108	208	156
6	120	234	456	342
8	208	408	800	600
10	320	630	1240	930
12	456	900	1776	1332
14	616	1218	2408	1806
16	800	1584	3136	2352