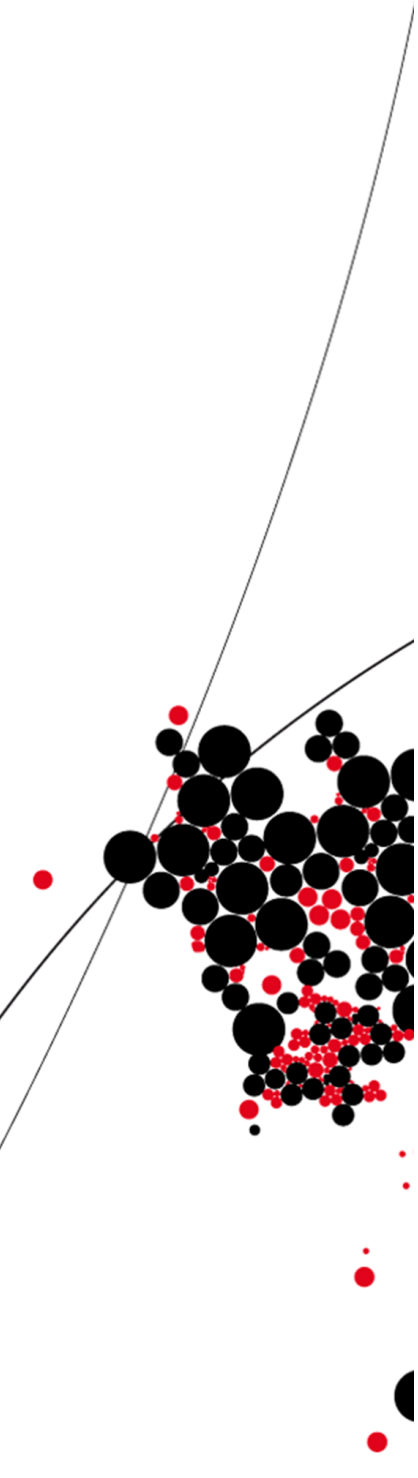
UNIVERSITY OF TWENTE.

Faculty of Engineering Technology
Departement of Thermal and Fluid Engineering
Engineering Fluid Dynamics research group

A physics-compatible dual field discretization using domain decomposition

S.D.M. de Jong

M.Sc. Thesis
04-2023



Chair:
prof. dr. ir. C.H. Venner
Internal member:
dr. H. Ozdemir
External member:
dr. I. Ostanin
Supervisors:
dr. A. Brugnoli
dr. ir. R. Rashad
dr. ir. Y. Zhang

Report number:
425

Contents

Summary **4**

1 Introduction **5**

1.1 Literature study 6

1.2 Proposed method & Research questions 6

1.3 Structure of the thesis 8

2 Mathematical background **9**

2.1 Exterior Calculus 9

2.2 Basis Functions 12

3 Time independent Poisson equation **15**

3.1 Standard finite element formulation 15

3.1.1 Strong formulation 15

3.1.2 Weak formulations 16

3.2 Space discretization of mixed boundary conditions via a domain decomposition method . 18

3.2.1 Weak formulation 19

3.3 Numerical simulations 21

3.3.1 Analytical solution 21

3.3.2 Results 22

3.3.3 Conclusion 28

4 Time dependent Poisson equation **29**

4.1 Standard finite element formulation 29

4.1.1 Strong formulations 29

4.1.2 Weak formulations 30

4.2 Space discretization of mixed boundary conditions via a domain decomposition method . 31

4.2.1 The weak formulations for a decomposed domain 31

4.3 Time integration using the implicit midpoint scheme . . . 33
4.4 Numerical simulations . . . 34
4.4.1 Analytical solution . . . 35
4.4.2 Results . . . 35
4.4.3 Curl free flow . . . 37
4.4.4 Power Balance . . . 38
4.4.5 Conclusion . . . 41

5 Time Staggering **42**
5.1 Time integration using a staggering implicit midpoint scheme . . . 42
5.2 Numerical simulations . . . 43
5.2.1 Convergence . . . 43
5.2.2 Power balance . . . 44
5.2.3 Computational cost . . . 44

6 Conclusion **46**

7 Recommendation **47**

References **48**

A Appendix **50**
A.1 Firedrake simulation . . . 50

Summary

Numerical simulations are commonplace in many different applications. However, for a lot of numerical methods the conservation laws, such as mass, momentum, energy, etc., are not preserved. Recent developments of the dual field discretization make it possible to construct numerical schemes specifically designed to preserve the physical properties of a system. However, current physics-compatible methods are still facing issues when solving for mixed boundary conditions, because of the use of essential boundary conditions. In this work a numerical scheme is proposed for solving mixed boundary problems, while all boundary conditions are naturally imposed. The domain is decomposed using an internal interface such that each subdomain has homogeneous boundary conditions. First a mathematical framework is introduced capable of describing the geometrical and topological properties of the system. By taking the geometry and topology into account, the variables can be expressed in a more physical manner.

To show how the spatial discretization works, the proposed method is applied to time independent Poisson problem. The Poisson equation is first expressed in exterior calculus, for its inner and outer oriented forms. The equations are then converted into two different weak formulations based on the different orientations. One weak formulation is applied to one subdomain, while the other is applied to the other subdomain. The two subdomains are then coupled on the interface, which results in a dual field discretization. The simulations take place on a unit square domain, with one Dirichlet boundary condition and one Neumann boundary condition. The L^2-error and convergence rates are found to be similar to that of standard finite element discretizations. The curl free condition of the Poisson problem is found to be preserved on a discrete level for one of the formulations.

Then the time dependent Poisson problem is simulated to demonstrate a temporal discretization for physical-compatible methods. For the time integration an implicit midpoint scheme is used. The L^2-errors and convergence rates show comparable results to standard finite element methods, and the curl free condition is preserved on one subdomain. The conservation of energy is found to be preserved on each subdomain, and also on the entire domain coupled by the dual field method. Lastly, a time staggering scheme was applied to reduce the computational cost of the simulation, while having no detrimental effect on the performance of the scheme.

In conclusion, the proposed scheme is capable of meeting the curl free condition on part of the domain, the energy is preserved on each subdomain, and additionally the energy is also conserved on the entire domain. The time staggering scheme gave the same results as a non-staggering scheme, while significantly reducing the computational cost.

1 Introduction

Numerical simulations are used in many fields of engineering. For simulating the aerodynamics of aircraft, the hydrodynamics of ships, the electromechanical properties of robotics and many other applications. It is vital that these simulations perform well. However, the solutions of these numerical simulations do not always agree with the conservation laws. The absence of such conservation can lead to errors and instabilities in numerical simulations. For example, the well-known incompressible Navier-Stokes equations are notorious for having difficult to find analytical solutions, and thus numerical methods are employed to find approximate solutions. However, the use of such methods creates problems with the found numerical solutions. For instance, for the standard Taylor-Hood finite element discretization, the error for the mass conservation is dependent on the mesh size [1]. For a reasonable mesh size, the error can still be significant enough to cause problems. Other conservation laws such as the momentum, helicity, energy and others, are also found to no longer be preserved when not taking care to use a proper discretization. In some cases, this can lead to discrepancies between the numerical simulations and the experimental data. The cause of this difference is due to the energy balance not being conserved in the numerical schemes. For example, the lack of energy conservation can lead to earlier flow separation on airfoils in simulations than in experimental results [2]. Furthermore, Modesti and Pirozzoli [3] found that the shockwaves on an airfoil are in incorrect locations on an airfoil when compared to experimental data.

The mathematical equations, e.g. the Navier-Stokes equations, perfectly describe the physical system, yet it is not straightforward to capture the physical properties in numerical simulations. That is because the numerical simulation transforms the problem from a continuum to a discrete level. Some of the physical properties captured on the continuum level are found to be no longer preserved on the discrete level. The inconsistency between continuum and discrete levels was already identified as early as 1959 [4]. It was realized that a proper spatial discretization should preserve important physical properties [5, 6].

To understand what is happening the conservation laws the geometry and topology of the underlying problem needs to be considered. Take for example a plane in $\mathbb{R}^3$, which can be described by a pair of basis directions $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, or by the normal direction of the plain $\boldsymbol{n}$, related by the cross product $\boldsymbol{e}_1 \times \boldsymbol{e}_2 = \boldsymbol{n}$. Each of these terms is an equally valid description of the plane. This can be further generalized into the statement that any k-dimensional object in n-dimensional space can be described by k directions or by its complementary $n-k$ directions. From a physical point, this means that any variable has two complementary, but equally valid, descriptions. The velocity in $\mathbb{R}^3$ can be associated with a line $k=1$ through the velocity potential, or by a surface $n-k=2$ by means of the mass flux. These are both valid descriptions of the velocity, and while they are connected to each other, they are obviously not equal. Nevertheless, a lot of numerical schemes do not take this into consideration and will use them as if they are equal. This introduces errors into the simulations which results in the lack of conservation in the solutions.

Vector calculus is found to be lacking in differentiating between this dual representation of the physical variables. While variables in vector calculus contain geometric and topological information, this information is not explicitly conveyed by the mathematical description, making it difficult to use the geometry and topology correctly. The mathematical field of Exterior Calculus (EC) is able to convey the geometrical and topological information about a problem. This allows for the k-forms and $n-k$-forms of a variable to be taken into account when setting up the numerical simulation.

1.1 Literature study

The desire to create a numerical scheme that complies with the conservation laws is not new. A lot of research has been done on the subject and some promising schemes have been found. Yet most of them do not consider the geometry and topology fundamental to the problem. Cai et al. [7] conducted numerical simulations on the sine-Gordon equation with homogeneous Neumann boundary conditions using second-order central difference with a cell-centered grid, and summation by parts operators on a regular grid. Both methods were found to preserve the energy on a discrete level within machine precision. Cai and Wang [8] found one method that preserves the energy, and another method which preserved the momentum for a version of the Boussinesq equation with periodic and homogeneous boundary conditions.

To ensure the desired physical properties are preserved, a numerical method is required to properly apply the geometrical and topological information made available by exterior calculus. The beginnings of such methods date back surprisingly far. In the 1950's the Whitney forms were developed [9], which are finite elements capable of preserving certain physical properties. However, while the Whitney forms were found to have better results [10], the mathematical understanding of why did not come about until the 1990's [11]. The work done in the 90's led to the development of Finite Element Exterior Calculus (FEEC) by Arnold et al. in the early 2000's [12, 13, 14]. As the name suggest FEEC is a finite element method, but it uses exterior calculus instead of vector calculus. Most notable, the finite elements used in FEEC are based on exterior calculus [15], this allows the element to maintain the geometric and topological properties of the problem on a discrete level.

FEEC is therefore a great candidate for creating physical-compatible discretizations. By keeping track of the k-forms and $n-k$-forms of the variables, the problem can be solved in different ways. One formulation based on the k-forms and one formulation based on the $n-k$-forms. These schemes can subsequently be combined into a method that uses both forms. In literature such a method is often referred to as the dual field method. Zhang et al. [16] suggested a mimetic dual field method capable of conserving mass, kinetic energy and helicity for three dimensional incompressible Navier-Stokes equations with periodic boundary condition. Brugnoli et al. [17] used a dual field discretization to conduct simulations on the acoustic wave equation and the Maxwell equations in three dimension with mixed Dirichlet and Neumann boundary conditions. For both formulations using the k-forms and $n-k$-forms separately, one boundary condition was naturally imposed, while the other boundary condition was essentially enforced. They found that energy is not preserved for either formulation. However, when the two systems are combined to create a dual field method using both forms the energy was found to be preserved. This is undesired as the energy balance in the dual field method is merely reconstructed from the individual formulations which are not preserving the energy, preferably all systems conserve the energy on a discrete level. It is the use of an essential boundary condition that introduces problems in the discretizations. An essential boundary condition applies algebraic constrains, which makes them more difficult to deal with in numerical simulations.

1.2 Proposed method & Research questions

A scheme that preserves the physics on a discrete level without the use of use any essential boundary conditions would be desired, as such a scheme should be able to preserve the energy balance. The method suggested here is to introduce an internal interface boundary which decomposes the domain such that each subdomain has homogeneous boundary conditions. The proposed numerical method is to apply the dual field formulation to the decomposed domain. Each subdomain will solve for one of the formulations

of the variables, i.e. one subdomain will use the k-forms while the other uses the $n-k$-forms. The two subdomains are then coupled on the interface to create a dual field formulation on the entire domain.

The use of subdomains provides another opportunity for the proposed numerical scheme. Since each form of a variable is only solved on part of the domain instead of on the entire domain, a method is proposed to reduce the computational cost of the problem. A time staggering method is used such that for each (half) time step the solver only has to compute part of the domain. Depending on how the domain is decomposed, this can significantly reduce the computational domain for each (half) time step.

The proposed numerical scheme is to be a physics-compatible method. To ensure the proposed scheme does indeed work as intended, it needs to satisfy certain conditions. This leads to the main question to be answered in this work.

Can the physical properties be preserved discretely for problems with mixed boundary conditions by decomposing the domain?

The main research question can be divided into subquestions. Firstly, is to ask if the physics are preserved on each subdomain, since previous works have found that the energy can be preserved when the boundary conditions are homogeneous, the subdomains are expected to conserve the energy exactly. Which other physical conditions can be preserved will have to be determined. Secondly, to couple the subdomains while preserving the physics on the whole domain. When the subquestions have been answered, the main research questions can be answered too.

1. **Are the physics preserved on each subdomain?**
2. **Can the subdomain be coupled such that the physics are preserved on the entire domain?**

When the first two subquestions have been answered, the proposed time staggering scheme can be implemented. To ensure the energy balance is not effected by the implementation of a time staggering scheme, the energy balance is tested again. An estimate of the computational cost of the staggering and non-staggering methods is made to compare the different methods.

3. **Is the energy balance still preserved when using a time staggering scheme?**
4. **How much is the computational cost reduced by using a time staggering scheme when compared to non-staggering schemes?**

To investigate how the proposed scheme functions the Poisson problem $\partial_{tt}\phi - \nabla^2\phi = f(x,y)$ is discretized for a general domain with mixed boundary conditions, as seen in Figure 1. The time derivative and the Laplacian ∇^2 appear in a lot of problems, so the Poisson problem serves as a case study. The propsed method can be further generalized in future work.

$$\phi = u_0 \quad \text{on } \Gamma_{21}, \tag{1}$$

$$\nabla\phi \cdot n = u_1 \quad \text{on } \Gamma_{10}. \tag{2}$$

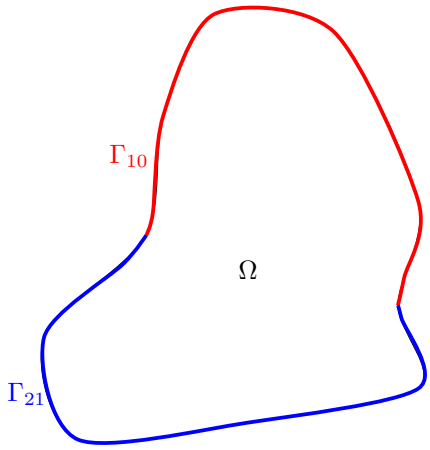


Figure 1: A general domain with mixed boundary conditions, with an Neumann boundary condition on Γ_{10} and a Dirichlet boundary condition on Γ_{21}.

1.3 Structure of the thesis

The thesis is structured as follows. In Section 2 the mathematical background necessary for understanding the rest of the work is given. The Poisson problem is then discretized for a general domain with mixed boundary conditions. First for the steady case, i.e. $\partial_t \phi = 0$, to show how the spatial discretization works for a standard finite element formulation in Section 3.1, and then for the proposed method in Section 3.2. The numerical simulations are performed on a unit square domain. Then for the time dependent case with a zero source function, $f(x, y) = 0$, using an implicit midpoint scheme for the time integration in Section 4. Finally in Section 5 a time staggering scheme is applied to the same problem as in Section 4, and the computational costs of the different methods are investigated. The appendix contains a short explanation the code used for the simulation, with some example codes to show the more uncommon sections of code.

2 Mathematical background

As mentioned in the introduction, the mathematical field of exterior calculus is used in this thesis instead of the more commonly known vector calculus. This section is meant to provide some of the concepts and definitions from exterior calculus required to understand the thesis. The use of exterior calculus in this work is mainly focused on its applications in FEEC, so the concepts and definitions found in this section are relevant to FEEC and not a general explanation of exterior calculus. For a more complete explanation the reader is encouraged to look at other resources, such as [18, 19, 20].

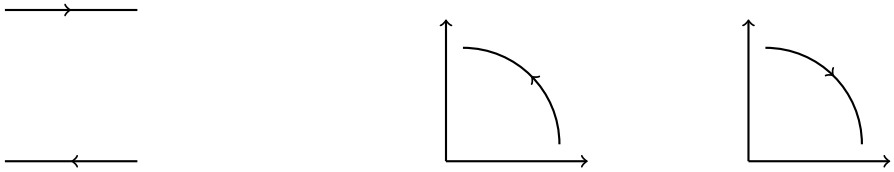
A note on vector calculus in this report.

To aid those unfamiliar with exterior calculus some parts of the thesis are also provided in an equivalent vector calculus notation, recognized by a blue box such as this one. These boxes provide the reader with equations in vector calculus form, with a short explanation about the equation. As the discretization in this thesis relies on concepts from exterior calculus, these vector calculus sections do not provide an explanation of why certain steps are taken. As a final note, these sections provide no new information so any reader familiar with exterior calculus can skip these sections.

2.1 Exterior Calculus

To create physics-compatible numerical schemes, the orientations need to be taken into account. While some readers might be unfamiliar with the concept of an orientation, it is likely they have used them often without knowing about it. A preferred orientation is often used for some conventions in physics and engineering, e.g. the outer unit normal instead of the inner unit normal, or the vorticity is positive in counterclockwise orientation. These conventions are an arbitrary choice, that is the physics are the same no matter the orientation.

The concept of orientations can also be applied to manifold, where any n-dimensional manifold can be said to have two orientations, a fact that becomes more evident in lower dimensions. For example, in 1-dimension it is possible to move in only two directions, which describe the two possible orientations of a 1-dimensional manifold, shown in Figure 2a. In 2-dimensions the directions are clockwise and anticlockwise that show the possible orientations of a manifold in two dimensions in Figure 2b. Similar to how the sign conventions in physics are merely a choice, one orientation is named the "True" or "Inner" orientation, while the other is called the "Pseudo" or "Outer" orientation. In exterior calculus one can change from one orientation to the other by means of the Hodge star operator $*$.



(a) The two orientations in $\mathbb{R}^1$.

(b) The two orientations in $\mathbb{R}^2$.

Figure 2: Orientations in $\mathbb{R}^1$ and $\mathbb{R}^2$.

Exterior calculus uses differential forms as its fundamental objects. In general, it can be said that for an n-dimensional manifold, it contains $n+1$ types of sub-manifolds, e.g. a surface contains points, lines and surfaces. Each differential form is related to one of these sub-manifolds, so a 0-form is related to a 0-dimensional sub-manifold, a 1-form to a 1-dimensional sub-manifold, etc., or in general a k-form is related to a k-dimensional sub-manifold. Much like in vector calculus scalar and vector fields are connected by operators like gradient or curl, the different k-forms are also connected by the exterior derivative d.

With the Hodge operator $*$ and the exterior derivative d, it is possible to move between different k-forms and orientations. To see how these operators function consider a differentiable Riemann manifold $M \subset \mathbb{R}$ with boundary ∂M, the space of smooth differential k-forms are denoted by $\Omega^k(M)$. The differential k-forms are connected through the exterior derivative d, which maps a k-form to a $k+1$-form, i.e. $d : \Omega^k(M) \rightarrow \Omega^{k+1}(M)$. The orientations of the manifold M results in the inner forms $\Omega^k(M)$ and the outer forms $\hat{\Omega}^k(M)$. The inner and outer forms are connected by means of the hodge star operator $* : \Omega^k(M) \rightarrow \hat{\Omega}^{n-k}(M)$, which maps a k-form to its complementary $n-k$-form. The different k-forms and orientations can then be connected by the use of the exterior derivative d and the hodge star operator $*$. As this work deals only with 2-dimensional problems, the complex for a 2-dimensional manifold in Figure 3 as an example. In general, such a complex exists for any n-dimensional problem.

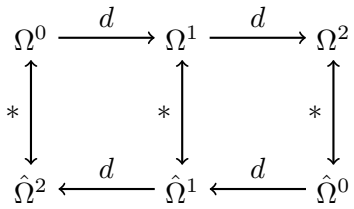
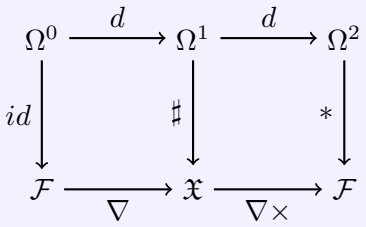


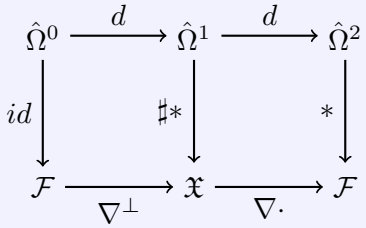
Figure 3: 2D complex in exterior calculus

Connecting forms to scalar and vector fields.

It is possible to translate forms to their respective vector calculus equivalent, shown here for the 2-dimensional case, for a more general and in depth explanation please see [21]. In the 2-dimensional case, the 0-form in either orientation is identically the scalar field $\mathcal{F}$. A 2-form can then be translated to the scalar field $\mathcal{F}$ by means of the hodge operator, as the hodge of a 2-form results in a 0-form (see Figure 3). To move between 1-forms and vector fields the canonical isomorphisms $\sharp$ and $\flat$ are used. The notation is inspired by musical notation and so they are referred to as sharp $\sharp$ and flat $\flat$ respectively. An inner 1-form Ω^1 is turned into a vector field by $\sharp$ and the inverse by $\flat = \sharp^{-1}$. For the outer 1-form $\hat{\Omega}^1$ a hodge has to be applied first before using the sharp $\sharp$. This results in two different isomorphism diagrams in 2-dimensions, for the inner and outer forms in Figures 4a and 4b respectively.



(a) Isomorphisms for the inner forms.



(b) Isomorphisms for the outer forms.

Figure 4: Isomorphisms for forms to scalar fields $\mathcal{F}$ and vector fields $\mathfrak{X}$ in 2-dimensions.

Four different vector calculus operations appear, the gradient ∇, the 2D-curl $\nabla\times$, the skew gradient $\nabla^\perp$ and the divergence $\nabla\cdot$. One may wonder why four different operations appear. This is due to considering the orientations. Take, for instance, the gradient $\nabla : \mathcal{F} \to \mathfrak{X}$ has the same mapping as $\sharp d : \mathcal{F} \to \mathfrak{X}$. Doing the same for the other vector calculus operations it is found that

$$\nabla = \sharp d, \tag{3}$$
$$\nabla\times = *d\flat, \tag{4}$$
$$\nabla^\perp = \sharp * d, \tag{5}$$
$$\nabla\cdot = *d*\flat. \tag{6}$$

The difference between the gradient and the skew gradient is that the hodge operator appears, which changes the orientation. This is also serves as an example of why it is important to consider the orientation.

With the basic definitions of the exterior derivative d and the hodge operator $*$, the use of exterior calculus in FEEC can be described. In finite element method various mathematical operations have to be performed, which have their own notations in exterior calculus. To ensure the exterior calculus notation of these operations is understood, the definitions for the operations encountered in this work are given in the section below.

Definition 1 (L^2 inner product)**.** *The L^2 inner product is defined for a smooth manifold of dimension*

$n := dim(M)$.

$$(\alpha^k, \gamma^k)_M := \int_M \alpha \wedge *\gamma^k \qquad \forall(\alpha, \gamma) \in \Omega^k(M), \tag{7}$$

where $\wedge$ is the wedge product which maps $\wedge : \Omega^k \times \Omega^l \to \Omega^{k+l}$ when $k, l \geq 0$ and $k + l \leq n$.

Definition 2 (Duality product)**.** *For a smooth manifold of dimension n the duality product is defined by*

$$\left\langle \alpha^k | \beta^{n-k} \right\rangle_M := \int_M \alpha^k \wedge *\beta^{n-k} \qquad \forall \alpha \in \Omega^k(M), \forall \beta^{n-k} \in \Omega^{n-k}(M) \tag{8}$$

Definition 3 (Duality product on the boundary)**.** *The duality product is also defined on the boundary, where the trace operator* tr *is defined as the pullback of the inclusion map $\iota : \partial M \to M$, such that*

$$\left\langle \alpha^k | \beta^{n-k-1} \right\rangle_{\partial M} := \int_{\partial M} \mathrm{tr}\alpha^k \wedge \mathrm{tr}\beta^{n-k-1} \tag{9}$$

Definition 4 (Co-differential)**.** *The codifferential is defined as :*

$$d^* := (-1)^{nk+n+1} * d * \tag{10}$$

Definition 5 (Integration by parts)**.** *Integration by parts for a codifferential in a weak formulation is*

$$(\alpha^k, d^*\beta^{k+1})_M := (d\alpha^k, \beta^{k+1})_M - < \alpha^k | *\beta^{k+1} >_{\partial M}, \qquad \forall \alpha \in \Omega^k(M), \forall \beta \in \Omega^{k+1}(M) \tag{11}$$

Definition 6 (Sobolev spaces)**.** *A form used in a weak formulation will then be an element of a Sobolev space such that $\alpha^k \in H\Omega^k(M)$, where H indicates a Sobolev space. Sobolev space is defined as*

$$H\Omega^k(M) := \{\omega^k \in L^2\Omega^k(M) | d\omega^k \in L^2\Omega^{k+1}(M)\}. \tag{12}$$

When the trace of a form is zero on the boundary, e.g. when an essential boundary condition is applied, the Sobolev space is limited such that

$$H_0\Omega^k(M, \Gamma) := \{\omega^k \in H\Omega^k(M) | \mathrm{tr}\omega|_\Gamma = 0\}. \tag{13}$$

2.2 Basis Functions

Any finite element has a canonical basis function φ which are used to determine the unknowns on the element. A more in depth explanation of the basis function can be found in the work of Kirby [22] or Brugnoli et al. [17]. When a form is discretized for use in a finite element method, the discrete from then becomes an element of a finite space, $\alpha_h^k \in V_h^k$, where V_h^k is a trimmed polynomial space for a triangular element $V_h^k \in \mathcal{P}^-\Omega^k(\Delta_n)$ [15]. The discrete form is then determined using the basis functions of the chosen element.

Definition 7 (Discrete differential forms)**.** *The discrete version of a k-form can be written as the sum over the total number of degrees of freedom N_k of the form at degree i times the basis function at degree i.*

$$\alpha_h^k = \sum_i^{N_k} \alpha_h^i \varphi^i, \tag{14}$$

for $\alpha_h^i \in \mathbb{R}$.

Definition 8 (Discrete inner product)**.** *The inner product of the discrete test and trial functions can be written as*

$$\begin{aligned}\left(v_h^k, \alpha_h^k\right)_M &= \left(\sum_i^{N_k} v_{h,i}^k \varphi_i^k, \sum_j^{N_k} \alpha_{h,j}^k \varphi_j^k\right)_M \\ &= \sum_j^{N_k}\sum_i^{N_k} v_{h,i}^k M_{ij} \alpha_{h,j}^k = (\boldsymbol{v}^k)^T \boldsymbol{M}^k \boldsymbol{\alpha}^k,\end{aligned} \tag{15}$$

$$[\boldsymbol{M}^k]_i^j := \left(\varphi_i^k, \varphi_j^k\right)_M, \tag{16}$$

where $[\boldsymbol{M}^k]_i^j$ *indicates row* i *and column* j *and* $\boldsymbol{M}^k \in \mathbb{R}^{N_k \times N_k}$*.*

Definition 9 (Inner product with exterior derivative)**.** *For two discrete forms* $\xi_h^{k+1} \in V_h^{k+1}$ *and* $\alpha_h^k \in V_h^k$ *the inner product is expressed as*

$$\left(\xi_h^{k+1}, d\alpha_h^k\right)_M = (\boldsymbol{\xi}^{k+1})^T \boldsymbol{D}^k \boldsymbol{\alpha}^k, \tag{17}$$

$$[\boldsymbol{D}^k]_i^j = \left(\varphi_i^{k+1}, d\varphi_j^k\right)_M, \tag{18}$$

where the terms in (17) *can be transposed such that* $\boldsymbol{D}^k = (\boldsymbol{D}^k)^T$*, where* $\boldsymbol{D}^k \in \mathbb{R}^{N_{k+1} \times N_k}$*. Furthermore,* $\boldsymbol{D}^k$ *can be split into an inner product an exterior derivative operation, i.e.* $\boldsymbol{D}^k = \boldsymbol{M}^{k+1}\boldsymbol{E}^{k+1,k}$*, with* $\boldsymbol{E}^{k,k+1} = (\boldsymbol{E}^{k+1,k})^T$*.*

Definition 10 (Duality product over the boundary)**.** *The duality product over the boundary for* $\xi_h^{n-k-1} \in V_h^{n-k-1}$ *and* $\alpha_h^k \in V_h^k$ *is expressed as*

$$\left\langle \xi_h^{n-k-1} | \alpha_h^k \right\rangle_\Gamma = (\boldsymbol{\xi}^{n-k-1})^T \boldsymbol{L}_\Gamma^k \boldsymbol{\alpha}^k, \tag{19}$$

$$[\boldsymbol{L}_\Gamma^k]_i^j = \left\langle \varphi_i^{n-k-1} | \varphi_j^k \right\rangle_\Gamma, \tag{20}$$

where $\boldsymbol{L}_\Gamma^k \in \mathbb{R}^{N_{n-k-1} \times N_k}$ *and the transpose is* $\boldsymbol{L}_\Gamma^k = (-1)^{k(n-k-1)}(\boldsymbol{L}_\Gamma^{n-k-1})^T$*. The matrix* $\boldsymbol{L}_\Gamma^k$ *contains all the degrees of freedom of the entire domain* M*, but can only be non-zero on the boundary* ∂M*. The size of the matrix* $\boldsymbol{L}_\Gamma^k$ *can be reduced by taking the trace of the discrete forms.*

Definition 11 (Trace of a discrete k-form)**.** *The trace of a discrete* k*-form* $\alpha_h^k \in V_h^k$ *can be written as a sum of the trace of the basis functions multiplied by the* k*-form, which can then be shortened into the sum of the basis functions times the* k*-form over the degrees of freedom on on the boundary*

$$\mathrm{tr}\alpha_h^k = \sum_i^{N_k} \mathrm{tr}(\varphi_i^k)\alpha_i^k = \sum_l^{N_k^\Gamma} \varrho_l^k \alpha_{l,\Gamma}^k. \tag{21}$$

Definition 12 (Trace matrix)**.** *The vector of* k*-form* $\beta_h^k \in V_h^k$*, can be reduced to its components on the boundary by using the trace matrix such that*

$$\boldsymbol{\beta}_\Gamma^k = \boldsymbol{T}^k \boldsymbol{\beta}^k, \quad [\boldsymbol{T}^k]_l^i = \begin{cases} 1, \textit{if } \mathrm{tr}(\varphi_j^k) = \varrho_l^k & \forall l = 1, \cdots, N_k^\partial \\ 0, \textit{otherwise} & \forall i = 1, \cdots, N_k \end{cases} \tag{22}$$

The duality product on the boundary in (19) can then be expressed as

$$\left\langle \xi_h^{n-k-1} | \alpha_h^k \right\rangle_\Gamma = (\boldsymbol{T}^{n-k-1}\boldsymbol{\xi}^{n-k-1})^T \boldsymbol{\varrho}_\Gamma^k \boldsymbol{\alpha}_\Gamma^k, \tag{23}$$

$$\boldsymbol{\varrho}_\Gamma^k = \left\langle \varrho_i^{n-k-1} | \varrho_j^k \right\rangle_\Gamma, \tag{24}$$

where $\boldsymbol{\varrho}_\Gamma^k \in \mathbb{R}^{N_{n-k-1}^\Gamma \times N_k^\Gamma}$ and $\boldsymbol{L}_\Gamma^k$ is related to $\boldsymbol{\varrho}_\Gamma^k$ by

$$\boldsymbol{L}_\Gamma^k = (\boldsymbol{T}^{n-k-1})^T \boldsymbol{\varrho}_\Gamma^k \boldsymbol{T}^k. \tag{25}$$

From Equations (19) (23) the following matrix relation can be found

$$\boldsymbol{L}_\Gamma^k \boldsymbol{\alpha}^k = \boldsymbol{B}_\Gamma^k \boldsymbol{\alpha}_\Gamma^k, \tag{26}$$

$$\boldsymbol{B}_\Gamma^k = (\boldsymbol{T}^{n-k-1})^T \boldsymbol{\varrho}_\Gamma^k. \tag{27}$$

3 Time independent Poisson equation

In this section the steady Poisson problem, $-\nabla^2\phi = f(x,y)$ is discretized on a domain with mixed boundaries as shown in Figure 1. The problem is first redefined into exterior calculus instead of vector calculus for its application in FEEC. To obtain a physics-compatible scheme, a proper spatial discretization is required. The spatial integration for a standard finite element formulation is shown in Section 3.1 to be able to compare the proposed method to. The proposed method is discretized in Section 3.2. The simulation are performed on a unit square domain, the results of which are shown in Section 3.3.

The Poisson problem for the steady case, is first written in exterior calculus as follows

$$-*d*d\phi^0 = f^0, \tag{28}$$

where the boundary conditions in exterior calculus are now

$$\mathrm{tr}\phi^0|_{\Gamma_{21}} = u^0, \tag{29}$$
$$\mathrm{tr}\hat{\sigma}^1|_{\Gamma_{10}} = \hat{u}^1. \tag{30}$$

3.1 Standard finite element formulation

3.1.1 Strong formulation

The dual formulation solves for ϕ^0 and σ^1 on the entire domain. The Neumann boundary condition on Γ_{10} will be naturally imposed, while the Dirichlet boundary condition on Γ_{21} will be essential. The strong formulation for the dual system is given below as

$$d^*\sigma^1 = f^0, \tag{31}$$
$$d\phi^0 = \sigma^1, \tag{32}$$

where $\phi^0 \in \Omega^0(M)$ and $\sigma^1 \in \Omega^1(M)$. The definition for the co-differential from (10) has been used. Using the property $dd\alpha = 0$ for $\alpha \in \Omega^0$, which states that applying the exterior derivative on a 0-form twice will always result in zero, [17], the structure $d\sigma^1 = dd\phi^0 = 0$ from (32) is found. The strong formulation of the dual system is strongly enforcing that the exterior derivative of σ^1 is equal to zero everywhere in the domain.

Doing the same for the complementary primal formulation solving for $\hat{\phi}^2$ and $\hat{\sigma}^1$. The Poisson problem for the primal mixed formulation can be written as the strong formulation as

$$-d\hat{\sigma}^1 = \hat{f}^2, \tag{33}$$
$$\hat{\sigma}^1 = -d^*\hat{\phi}^2, \tag{34}$$

where now $\hat{\phi}^2 \in \hat{\Omega}^2(M)$ and $\hat{\sigma}^1 \in \hat{\Omega}^1(M)$.

The strong formulation in vector calculus

The strong formulation in vector calculus notation is given by

$$-\nabla\phi = \sigma, \tag{35}$$
$$\nabla \cdot \sigma = f(x, y). \tag{36}$$

Only one strong formulation is shown here instead of the two that appear in exterior calculus, as vector calculus notation does not convey any information about the orientation of the variable. One can say there are two strong formulations, similar to the two in exterior calculus, but writing them vector calculus notation will result in duplicate notation. The difference in orientation only becomes apparent in the weak formulation when using vector calculus, so only one strong formulation has been shown here.

3.1.2 Weak formulations

The weak formulation for the dual system is obtained by multiplying by the test functions v^0 and τ^1. Given the source function $f^0 \in \Omega^0(M)$ and boundary condition $\hat{u}^1 \in H^{-1/2}\hat{\Omega}^1(\Gamma_{10})$, find $\phi^0 \in H\Omega^0(M)$ and $\sigma^1 \in H\Omega^1(M)$ such that

$$\left(dv^0, \sigma^1\right)_M = \left(v^0, f^0\right)_M + \left\langle v^0 | *\sigma^1 \right\rangle_{\Gamma_{10}} \qquad \forall v^0 \in H_0\Omega^0(M, \Gamma_{21}), \tag{37}$$
$$\left(\tau^1, \sigma^1\right)_M - \left(\tau^1, d\phi^0\right)_M = 0 \qquad \forall \tau^1 \in H\Omega^1(M), \tag{38}$$

subject to $\mathrm{tr}\phi^0|_{\Gamma_{21}} = u^0$, where the term with the co-differential is integrated by parts using (11).

The basis functions from Section 2.2 can then be applied to the terms in the weak formulation to obtain the algebraic forms. The inner product with an exterior derivative on the test function v^0 can be written as

$$\left(dv^0, \sigma^1\right)_M = (\boldsymbol{\sigma}^1)^T \boldsymbol{D}^0 \boldsymbol{v}^0 = (\boldsymbol{v}^0)^T (\boldsymbol{D}^0)^T \boldsymbol{\sigma}^1, \tag{39}$$

where the transpose is taken such that the transpose of the test function is taken. This is important because later the transpose of the test function is eliminated from the equation. The inner product with the known source function f^0 is

$$\left(v^0, f^0\right)_M = (\boldsymbol{v}^0)^T \boldsymbol{M}^0 \boldsymbol{f}^0. \tag{40}$$

The duality product on the boundary is

$$\left\langle v^0 | *\sigma^1 \right\rangle_{\Gamma_{10}} = (\boldsymbol{v}^0)^T \boldsymbol{L}^1_{\Gamma_{10}} \hat{\boldsymbol{\sigma}}^1 = (\boldsymbol{v}^0)^T \boldsymbol{B}^0_{\Gamma_{10}} \hat{\boldsymbol{u}}^1, \tag{41}$$

where $\boldsymbol{L}^1_{\Gamma_{10}}$ is reduced to the boundary matrix $\boldsymbol{B}^0_{\Gamma_{10}}$ using the trace matrix from (22). The inner product with an exterior derivative on ϕ^0 becomes

$$\left(\tau^1, d\phi^0\right)_M = (\boldsymbol{\tau}^1)^T \boldsymbol{D}^0 \boldsymbol{\phi}^0, \tag{42}$$

where the transpose of the test function τ^1 appears immediately from (17). Lastly the inner product of two 1-forms is

$$\left(\tau^1, \sigma^1\right)_M = (\boldsymbol{\tau}^1)^T \boldsymbol{M}^1 \boldsymbol{\sigma}^1. \tag{43}$$

The problem can be rewritten to a linear problem of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$,

$$\begin{bmatrix} 0 & (\boldsymbol{D}^0)^T \\ -\boldsymbol{D}^0 & \boldsymbol{M}^1 \end{bmatrix} \begin{pmatrix} \boldsymbol{\phi}^0 \\ \boldsymbol{\sigma}^1 \end{pmatrix} = \begin{bmatrix} \boldsymbol{M}^0 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{f}^0 \\ \hat{\boldsymbol{f}}^2 \end{pmatrix} + \begin{bmatrix} 0 & \boldsymbol{B}^1_{\Gamma_{10}} \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^0 \\ \hat{\boldsymbol{u}}^1 \end{pmatrix}, \tag{44}$$

where the left hand side contains the unknowns $\boldsymbol{\phi}^0$ and $\boldsymbol{\sigma}^1$ and the right side the known source term $\boldsymbol{f}$ and the boundary input $\boldsymbol{u}$. Similarly for the primal system, given $\hat{f}^2 \in \hat{\Omega}^2(M)$ and $u^0 \in H^{-1/2}\Omega^0(\Gamma_{21})$, seek $\hat{\phi}^2 \in H\hat{\Omega}^2(M)$ and $\hat{\sigma}^1 \in H\hat{\Omega}^1(M)$ such that

$$-\left(v^2, d\hat{\sigma}^1\right)_M = \left(v^2, \hat{f}^2\right)_M \qquad \forall v^2 \in H\hat{\Omega}^2(M), \tag{45}$$

$$\left(\tau^1, \hat{\sigma}^1\right)_M + \left(d\tau^1, \hat{\phi}^2\right)_M = \left\langle \tau^1 | * \hat{\phi}^2 \right\rangle_{\Gamma_{21}} \qquad \forall \tau^1 \in H_0\hat{\Omega}^1(M, \Gamma_{10}), \tag{46}$$

subject to $\mathrm{tr}\hat{\sigma}^1|_{\Gamma_{10}} = \hat{u}^1$.

The terms of the weak formulation are rewritten with the basis functions as described in Section 2.2. The inner product with an exterior derivative of an outer form is

$$\left(v^2, d\hat{\sigma}^1\right)_M = (\boldsymbol{v}^2)^T \boldsymbol{D}^1 \hat{\boldsymbol{\sigma}}^1. \tag{47}$$

The inner product is

$$\left(v^2, \hat{f}^2\right)_M = (\boldsymbol{v}^2)^T \boldsymbol{M}^2 \hat{\boldsymbol{f}}^2, \tag{48}$$

the inner product of two 1-forms is again

$$\left(\tau^1, \hat{\sigma}^1\right)_M = (\boldsymbol{\tau}^1)^T \boldsymbol{M}^1 \hat{\boldsymbol{\sigma}}^1. \tag{49}$$

For the inner product with an exterior derivative of the test function, a transpose needs to be taken again such that

$$\left(d\tau^1, \hat{\phi}^2\right)_M = (\hat{\boldsymbol{\phi}}^2)^T \boldsymbol{D}^1 \boldsymbol{\tau}^1 = (\boldsymbol{\tau}^1)^T (\boldsymbol{D}^1)^T \hat{\boldsymbol{\phi}}^2. \tag{50}$$

The duality product on the boundary is

$$\left\langle \tau^1 | * \hat{\phi}^2 \right\rangle_{\Gamma_{21}} = (\boldsymbol{\tau}^1)^T \boldsymbol{L}^0_{\Gamma_{21}} \boldsymbol{\phi}^0 = (\boldsymbol{\tau}^1)^T \boldsymbol{B}^0_{\Gamma_{21}} \boldsymbol{u}^0, \tag{51}$$

where $\boldsymbol{L}^0_{\Gamma_{21}}$ is reduced to $\boldsymbol{B}^0_{\Gamma_{21}}$ using the trace matrix from (22). The problem can then be written as a linear problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$,

$$\begin{bmatrix} 0 & -\boldsymbol{D}^1 \\ (\boldsymbol{D}^1)^T & \boldsymbol{M}^1 \end{bmatrix} \begin{pmatrix} \hat{\boldsymbol{\phi}}^2 \\ \hat{\boldsymbol{\sigma}}^1 \end{pmatrix} = \begin{bmatrix} 0 & \boldsymbol{M}^2 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{f}^0 \\ \hat{\boldsymbol{f}}^2 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ \boldsymbol{B}^0_{\Gamma_{21}} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^0 \\ \hat{\boldsymbol{u}}^1 \end{pmatrix}, \tag{52}$$

where $\boldsymbol{x}$ are the unknowns $\hat{\boldsymbol{\phi}}^2$ and $\hat{\boldsymbol{\sigma}}^1$ and $\boldsymbol{b}$ the known inputs.

Weak formulations in vector calculus.

Doing the same in vector calculus for the dual system results in the weak formulation. Seek $(\phi, \sigma) \in H^1(M) \times H^{curl}(M)$ with $\phi|_{\Gamma_{21}} = u_0$ such that

$$\int_M \nabla v \cdot \sigma dx = \int_M v f dx + \int_{\Gamma_{10}} v\sigma \cdot n ds \qquad \forall v \in H_0^1(M, \Gamma_{21}), \tag{53}$$

$$\int_M \tau \cdot \nabla \phi dx = \int_M \tau \cdot \sigma dx \qquad \forall \tau \in H^{curl}(M). \tag{54}$$

The Neumann boundary condition can be seen in the integral, so it naturally imposed. The Dirichlet boundary in contrast, does not appear in the integral and thus has to be essential imposed.
For the primal weak formulation in vector calculus notation, seek $(\phi, \sigma) \in L^2 \times H^{div}$ with $\sigma|_{\Gamma_{10}} \cdot n = u_1$ such that

$$\int_M v \nabla \cdot \sigma dx = \int_M v f dx \qquad \forall v \in L^2(M), \tag{55}$$

$$\int_M \tau \cdot \sigma dx = -\int_M \phi \nabla \cdot \tau dx + \int_{\Gamma_{21}} \phi \tau \cdot n ds \qquad \forall \tau \in H_0^{div}(M, \Gamma_{10}). \tag{56}$$

One can see that the Dirichlet boundary condition is naturally implied in the integrals. The Neumann boundary is essential in this formulation.

3.2 Space discretization of mixed boundary conditions via a domain decomposition method

A domain M with Neumann and Dirchlet boundary conditions on Γ_{10} and Γ_{21} respectively, is split into two by introducing an arbitrary interface boundary Γ_{int}, as seen in Figure 5. This creates two subdomains such that $M = M_{10} \cup M_{21}$, where M_{10} is bounded by $\partial M_{10} = \Gamma_{10} \cup \Gamma_{int10}$ and M_{21} by $\partial M_{21} = \Gamma_{21} \cup \Gamma_{int21}$. The interface boundaries Γ_{int10} and Γ_{int21} are in the same location, but are in different orientations. Most notably the outer normal on the interfaces are in opposite directions, so $\Gamma_{int21} = -\Gamma_{int10}$. For ease of notation the interface of the M_{21} domain is redefined as $\Gamma_{int} := \Gamma_{int21}$, so that $\Gamma_{int10} = -\Gamma_{int}$.

The time independent Poisson problem is to be solved on this domain using the dual field discretization. The strong formulations for the dual and primal system are the same as for the standard method in Section 3.1.1, except that the unknowns are now contained to only part of the domain such that $\phi^0 \in \Omega^0(M_{10})$, $\sigma^1 \in \Omega^1(M_{10})$, $\hat{\phi}^2 \in \hat{\Omega}^2(M_{21})$ and $\hat{\sigma}^1 \in \hat{\Omega}^1(M_{21})$. The weak formulation set up in a similar way, but the domain decomposition in this problem requires coupling on the interface.

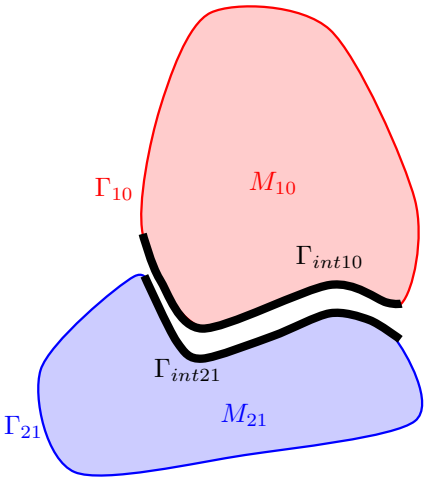


Figure 5: Domain for the domain decomposition method.

3.2.1 Weak formulation

The M_{10} domain will be discretized in the similar way as the dual formulation in Section 3.1. However, here there is an extra boundary in the interior boundary Γ_{int} which has to be taken into account. For the weak formulation for the decomposed domain, find $\phi^0 \in H\Omega^0(M_{10})$ and $\sigma^1 \in H\Omega^1(M_{10})$ such that

$$\left(dv^0, \sigma^1\right)_{M_{10}} - \left\langle v^0 | \hat{\sigma}^1 \right\rangle_{\Gamma_{int}} = \left(v^0, f^0\right)_{M_{10}} + \left\langle v^0 | * \sigma^1 \right\rangle_{\Gamma_{10}} \qquad \forall v^0 \in H\Omega^0(M_{10}), \tag{57}$$

$$\left(\tau^1, \sigma^1\right)_{M_{10}} - \left(\tau^1, d\phi^0\right)_{M_{10}} = 0 \qquad \forall \tau^1 \in H\Omega^1(M_{10}), \tag{58}$$

given $f^0 \in \Omega^0(M_{10})$ and $\hat{u}^1 \in H^{-1/2}\hat{\Omega}^1(\Gamma_{10})$. It can be noted here that the unknowns and the test functions are now in the same spaces, unlike in the standard formulation where they were in different spaces. This is because no essential boundary conditions have to be applied for the domain decomposition, and so the trace of the test function is not always zero on the boundary and thus does not have to be excluded from the space. Furthermore, there is no need to apply an algebraic constrained to the spaces of the unknowns.

The weak formulation in(57) and (58) can be rewritten using the basis functions from Section 2.2,

$$(\boldsymbol{D}^0)^T \boldsymbol{\sigma}^1 - \boldsymbol{L}^1_{\Gamma_{int}} \hat{\boldsymbol{\sigma}}^1 = \boldsymbol{M}^0 \boldsymbol{f}^0 + \boldsymbol{B}^1_{\Gamma_{10}} \hat{\boldsymbol{u}}^1, \tag{59}$$

$$\boldsymbol{M}^1 \boldsymbol{\sigma}^1 - \boldsymbol{D}^0 \boldsymbol{\phi}^0 = 0. \tag{60}$$

The weak formulation for the primal weak formulation is found in a similar way. The weak formulation is, given $\hat{f}^2 \in \hat{\Omega}^2(M_{21})$ and $u^0 \in H^{-1/2}\Omega^0(\Gamma_{21})$, seek $\hat{\phi}^2 \in H\hat{\Omega}^2(M_{21})$ and $\hat{\sigma^1} \in H\hat{\Omega}^1(M_{21})$ such that

$$-\left(v^2, d\hat{\sigma}^1\right)_{M_{21}} = \left(v^2, \hat{f}^2\right)_{M_{21}} \qquad \forall \hat{v}^2 \in H\hat{\Omega}^2(M_{21}), \tag{61}$$

$$\left(\tau^1, \hat{\sigma}^1\right)_{M_{21}} + \left(d\tau^1, \hat{\phi}^2\right)_{M_{21}} + \left\langle \tau^1 | \phi^0 \right\rangle_{\Gamma_{int}} = \left\langle \tau^1 | * \hat{\phi}^2 \right\rangle_{\Gamma_{21}} \qquad \forall \tau^1 \in H\hat{\Omega}^1(M_{21}). \tag{62}$$

The weak formulations for the primal system in Equations (61) and (62) can be rewritten using the basis

functions from Section 2.2,

$$-\boldsymbol{D}^1\hat{\boldsymbol{\sigma}}^1 = \boldsymbol{M}^2\hat{\boldsymbol{f}}^2, \tag{63}$$

$$\boldsymbol{M}^1\hat{\boldsymbol{\sigma}}^1 + (\boldsymbol{D}^1)^T\hat{\boldsymbol{\phi}}^2 + (\boldsymbol{L}^1_{\Gamma_{int}})^T\boldsymbol{\phi}^0 = \boldsymbol{B}^0_{\Gamma_{21}}\boldsymbol{u}^0. \tag{64}$$

The subdomains are coupled into one monolithic problem. Unlike the standard method, the four weak formulations are combined into one algebraic form $\boldsymbol{Ax} = \boldsymbol{b}$ instead of two separate ones. The coupling terms on the interface boundary can be found in the $\boldsymbol{A}$ matrix,

$$\begin{bmatrix} 0 & (\boldsymbol{D}^0)^T & 0 & -\boldsymbol{L}^1_{\Gamma_{int}} \\ -\boldsymbol{D}^0 & \boldsymbol{M}^1 & 0 & 0 \\ 0 & 0 & 0 & -\boldsymbol{D}^1 \\ (\boldsymbol{L}^1_{\Gamma_{int}})^T & 0 & (\boldsymbol{D}^1)^T & \boldsymbol{M}^1 \end{bmatrix} \begin{pmatrix} \boldsymbol{\phi}^0 \\ \boldsymbol{\sigma}^1 \\ \hat{\boldsymbol{\phi}}^2 \\ \hat{\boldsymbol{\sigma}}^1 \end{pmatrix} = \begin{bmatrix} \boldsymbol{M}^0 & 0 \\ 0 & 0 \\ 0 & \boldsymbol{M}^2 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{f}^0 \\ \hat{\boldsymbol{f}}^2 \end{pmatrix} + \begin{bmatrix} 0 & \boldsymbol{B}^1_{\Gamma_{10}} \\ 0 & 0 \\ 0 & 0 \\ \boldsymbol{B}^0_{\Gamma_{21}} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^0 \\ \hat{\boldsymbol{u}}^1 \end{pmatrix}. \tag{65}$$

The Poisson problem in vector calculus

The weak formulations for the M_{10} domain are obtained by seeking $(\phi_{10}, \sigma_{10}) \in H^1(M_{10}) \times H^{curl}(M_{10})$ such that

$$\int_{M_{10}} \sigma_{10} \cdot \nabla v_{10} dx + \int_{\Gamma_{int}} v_{10}\sigma_{10} \cdot n_{10} ds = \int_{M_{10}} f_{10} v_{10} dx + \int_{\Gamma_{10}} v_{10}\sigma_{10} \cdot n_{10} ds \quad \forall v_{10} \in H^1(M_{10}), \tag{66}$$

$$\int_{M_{10}} \sigma_{10} \cdot \tau_{10} dsx - \int_{M_{10}} \nabla\phi_{10} \cdot \tau_{10} = 0 \qquad \forall \tau_{10} \in H^{curl}(M_{10}). \tag{67}$$

The weak formulation in vector calculus on the M_{21} domain is to find $(\phi_{21}, \sigma_{21}) \in L^2(M_{21}) \times H^{div}(M_{21})$ such that

$$-\int_{M_{21}} v_{21} \nabla \cdot \sigma_{21} dx = \int_{M_{21}} f_{21} v_{21} dx \qquad \forall v_{21} \in L^2(M_{21}), \tag{68}$$

$$\int_{M_{21}} \tau_{21} \cdot \sigma_{21} dx + \int_{\Gamma_{21}} \phi_{21}\tau_{21} \cdot n_{21} ds - \int_{\Gamma_{int}} \phi_{21}\tau_{21} \cdot n_{21} ds = \int_{M_{21}} \phi_{21} \nabla \cdot \tau_{21} dx \quad \forall \tau_{21} \in H^{div}(M_{21}). \tag{69}$$

The two subdomains can be coupled in the interface boundary, where information can travel between the two systems. By using the fact that on a continuum level $\sigma|_{\Gamma_{int10}} = \sigma|_{\Gamma_{int21}}$ and $n_{10} = -n_{21}$ on Γ_{int}, the stress in normal direction on the interface can be written as $\sigma_{10} \cdot n_{10} = -\sigma_{21} \cdot n_{21}$. This means the integral term in (66) can be written as

$$\int_{\Gamma_{int}} v_{10}\sigma_{10} \cdot n_{10} dS \longrightarrow -\int_{\Gamma_{int}} v_{10}\sigma_{21} \cdot n_{21} ds. \tag{70}$$

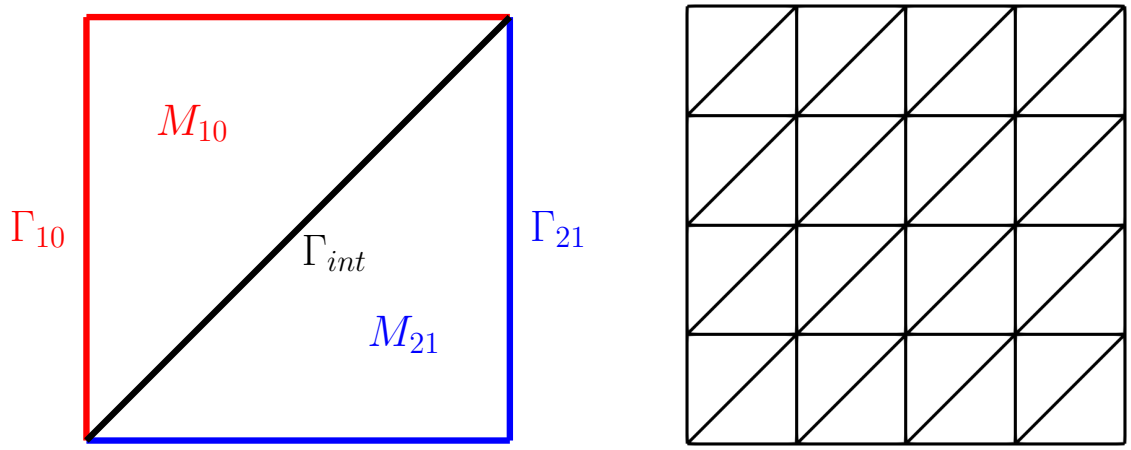
The integral term on the interface on the M_{21} domain (see (69)) can be coupled to the M_{21} domain by observing that $\phi_{21} = \phi_{10}$ on Γ_{int}. So the integral term in the weak formulation becomes

$$\int_{\Gamma_{int}} \phi_{21}\tau_{21} \cdot n_{21} ds \longrightarrow \int_{\Gamma_{int}} \phi_{10}\tau_{21} \cdot n_{21} ds. \tag{71}$$

3.3 Numerical simulations

In this section numerical simulations are conducted to determine the validity of the proposed scheme. An analytical solution is used, to be able to calculate the L^2-errors for each simulation, so the standard methods can be compared to the proposed method. The L^2-errors are further used to determine the convergence rates of the variables of the different schemes. These convergence rates can be compared to the expected theoretical convergence rates to ensure the simulations were performed correctly. A comparison of the L^2-error and the convergence rates of the different schemes is then made to see if the proposed method differs from the standard methods in terms. The curl of σ is then plotted for the standard formulations to see if the curl free condition of the Poisson problem is maintained on the discrete level.

The numerical simulations are performed on a unit square. For the domain decomposition method the interface is a diagonal line from the lower left to the upper right corner as shown in Figure 6a. The mesh used for solving the problem is a structured triangular mesh as shown in Figure 6b. The finite elements that are used are the Continuous Galerkin element for the 0-form ϕ^0, the Nédélec first kind H(curl) for the inner 1-form σ^1, the Discontinuous Galerkin element for the 2-form $\hat{\phi}^2$ and a Raviart-Thomas element for the outer 1-form $\hat{\sigma}^1$. The open source finite element solver Firedrake [23] has been used for the simulations. A more in depth explanation of how the simulation is set up in Firedrake can be found in Appendix A.1.



(a) Domain for the domain decomposition method.

(b) The structured triangular mesh for a spatial step of $h = 4$.

Figure 6: The decomposed unit square domain and an example mesh used for the simulations.

3.3.1 Analytical solution

The analytical solution used to analyze the different static methods is

$$g(x, y) = \cos(\omega_x x) \sin(\omega_y y), \tag{72}$$

the source term is then

$$q(x, y) = (\omega_x^2 + \omega_y^2) \cos(\omega_x x) \sin(\omega_y y). \tag{73}$$

The exact solutions are related by the hodge operator as following

$$\begin{aligned} \phi^0_{ex} &= g, \qquad \hat{\phi}^2_{ex} = *g, \qquad f^0 = q, \\ \sigma^1_{ex} &= dg, \qquad \hat{\sigma}^1_{ex} = *dg, \qquad \hat{f}^2 = *q. \end{aligned} \tag{74}$$

For the analytical solution $\omega_x = \omega_y = 1$ has been used. The exact solution is taken as the boundary condition so $u_0 = \phi^0_{ex}$ and $\hat{u}^1 = \hat{\sigma}^1_{ex}$.

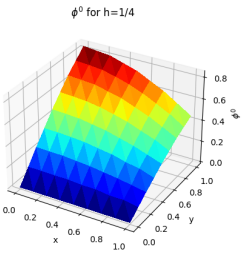
Analytical solution in vector calculus

The exact solutions in vector calculus notation are given by

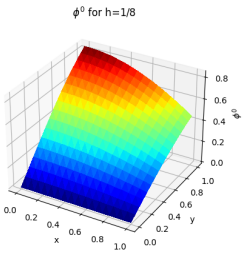
$$\phi = g, \tag{75}$$
$$\sigma = \nabla g, \tag{76}$$
$$f = q. \tag{77}$$

3.3.2 Results

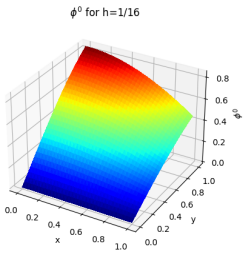
The results for ϕ^0 and σ^1 from the dual formulation have been plotted in Figure 7. The spatial step is calculated as $h = \frac{L}{N_{el}}$, where L is the length of each side, i.e. $L = 1$ for a unit square, and N_{el} the number of elements on each side. The results for ϕ^0 seem to include more elements than the spatial step h would suggest. This is because the Firedrake function used to plot the results uses Gaussian interpolation to smooth out the result. Nevertheless, as the spatial step h gets smaller the resulting figures can be seen to get smoother. The results for ϕ^0 are continuous over the domain, because the 0-form defined on the Continuous Galerkin element. A CG element always has some degrees of freedom on the vertices of a finite element [15], so bordering elements share the degrees of freedom on their shared vertices. This results in continuous results over the domain.



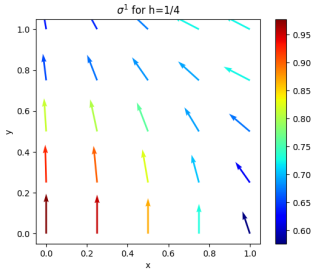
(a) ϕ^0 for a spatial step of $h = \frac{1}{4}$.



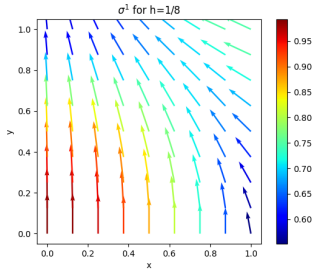
(b) ϕ^0 for a spatial step of $h = \frac{1}{8}$.



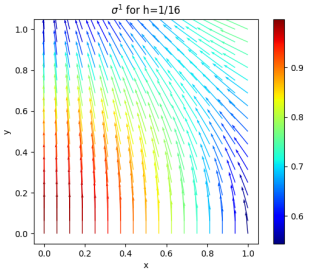
(c) ϕ^0 for a spatial step of $h = \frac{1}{16}$.



(d) σ^1 for a spatial step of $h = \frac{1}{4}$.



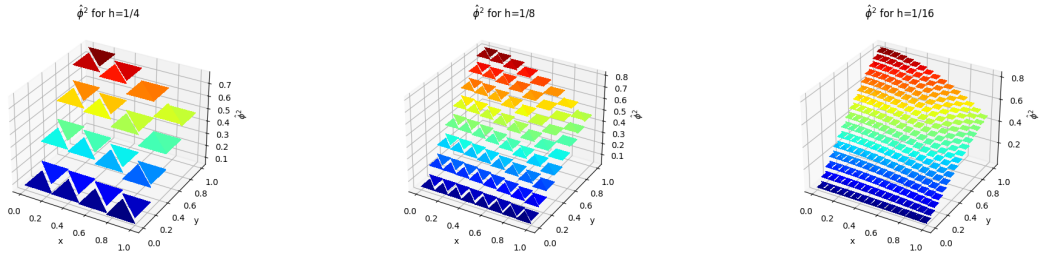
(e) σ^1 for a spatial step of $h = \frac{1}{8}$.



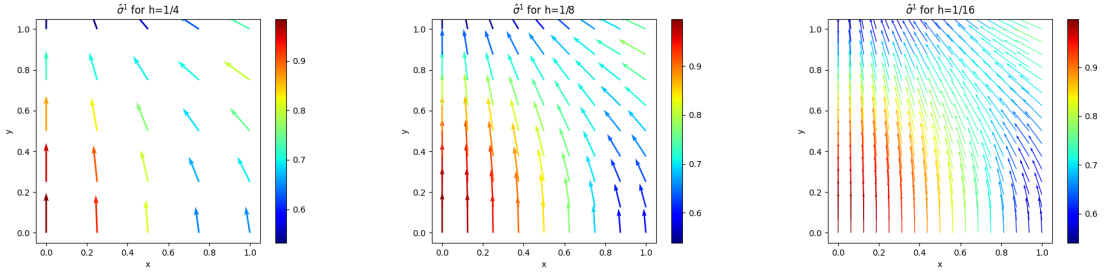
(f) σ^1 for a spatial step of $h = \frac{1}{16}$.

Figure 7: The results from the dual formulation over the domain M.

The results for $\hat{\phi}^2$ and $\hat{\sigma}^1$ from the primal formulation have been plotted in Figure 8. The results for $\hat{\phi}^2$ are observed to be discontinuous over the domain. This is because a 2-form is defined on a Discontinuous Galerkin (DG) element. The degrees of freedom for a DG element are found on the faces of the element [15], this means that when two elements border each other there are no shared degrees of freedom. This allows for discontinuities between the different elements as seen in Figures 14b.



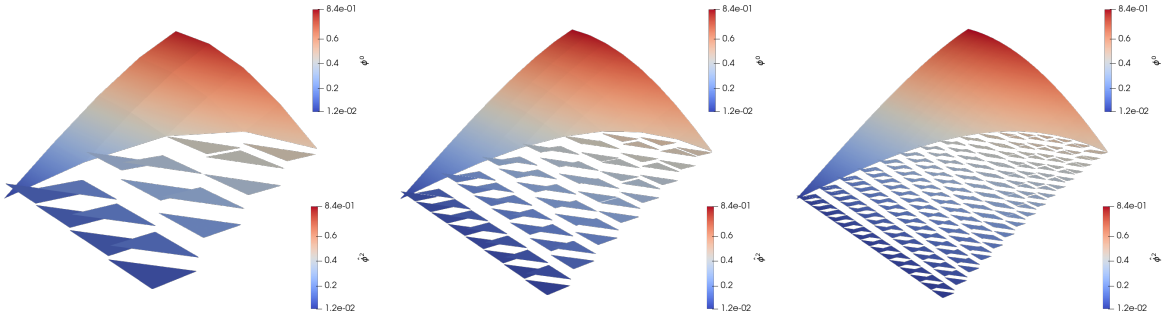
(a) $\hat{\phi}^2$ for a spatial step of $h = \frac{1}{4}$. (b) $\hat{\phi}^2$ for a spatial step of $h = \frac{1}{8}$. (c) $\hat{\phi}^2$ for a spatial step of $h = \frac{1}{16}$.



(d) $\hat{\sigma}^1$ for a spatial step of $h = \frac{1}{4}$. (e) $\hat{\sigma}^1$ for a spatial step of $h = \frac{1}{8}$. (f) $\hat{\sigma}^1$ for a spatial step of $h = \frac{1}{16}$.

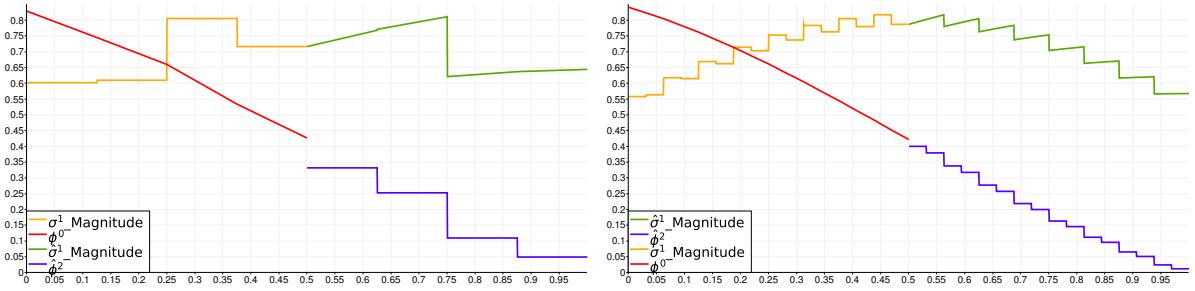
Figure 8: The results from the primal formulation over the domain M.

The visualization software Paraview [24] has been used to plot the results for the domain decomposition method in Figure 9. The results are continuous on the M_{10} domain and discontinuous on the M_{21} domain, due to the use of CG and DG elements respectively. The two domains in Figure 9 can be easily distinguished, but it is not immediately clear what is happening on the interface. Therefore the values of ϕ^0, σ^1, $\hat{\phi}^2$ and $\hat{\sigma}^1$ have been plotted on a line from the top left corner the bottom right corner in Figure 10. For ϕ^0 and $\hat{\phi}^2$ a discontinuity can be seen on the interface. For the coarser mesh (see Figure 10a) the gap is larger than for the finer mesh in Figure 10b. The discontinuity is getting smaller with decreasing mesh size.



(a) ϕ^0 on M_{10} and $\hat{\phi}^2$ on M_{21} for a spatial step of $h = \frac{1}{4}$. (b) ϕ^0 on M_{10} and $\hat{\phi}^2$ on M_{21} for a spatial step of $h = \frac{1}{8}$. (c) ϕ^0 on M_{10} and $\hat{\phi}^2$ on M_{21} for a spatial step of $h = \frac{1}{16}$.

Figure 9: The results for ϕ^0 and $\hat{\phi}^2$ over the domain M for different spatial steps.



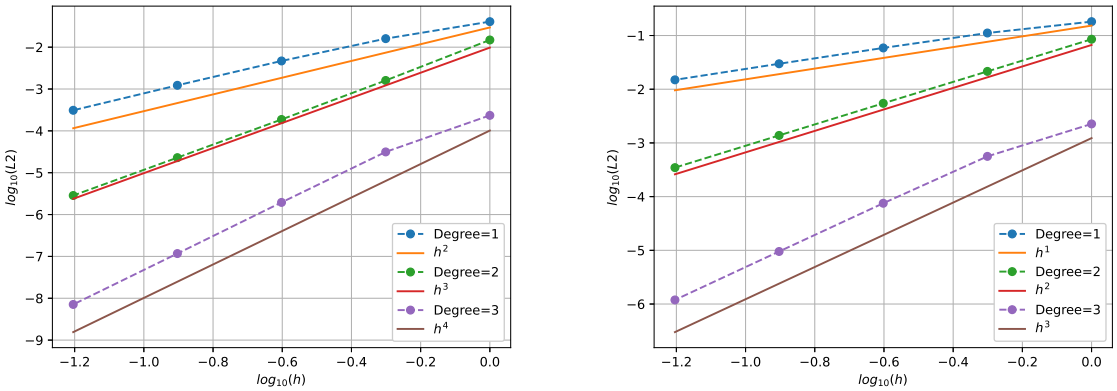
(a) ϕ^0, σ^1, $\hat{\phi}^2$ and $\hat{\sigma}^1$ for a spatial step of $h = \frac{1}{4}$.

(b) ϕ^0, σ^1, $\hat{\phi}^2$ and $\hat{\sigma}^1$ for a spatial step of $h = \frac{1}{16}$.

Figure 10: Results from the domain decomposition method on a line from $(x = 0, y = 1)$ to $(x = 1, y = 0)$.

The convergence rates for ϕ^0 and σ^1 from the dual system using the standard mixed element method are shown in Figures 11 for polynomial degrees $s = 1$, $s = 2$ and $s = 3$. To compare the convergence rates the theoretical convergence rates have also been plotted as h^s, where h is the spatial step and s the degree. The convergence rate of h^s indicates the theoretical convergence rate of an order s system, e.g. h^1 is a first order system.

The L^2-errors for ϕ^0 can be seen to exhibit superconvergence, i.e. the convergence rate is faster then the polynomial degree spatial step h^s, the convergence rates for ϕ^0 is instead h^{s+1}. This is in accordance with findings from Yonke and Yanhong [25], which found that for and element in $H\Omega^0$ the optimal convergence rate is rate is h^{s+1}. However, it should be noted that this result was found for problems with homogenous boundary conditions, and not specifically proven for mixed boundary conditions. Nevertheless, finding a convergence rate of h^{s+1} for $\phi^0 \in H\Omega^0$ for a mixed boundary problem is not unusual.

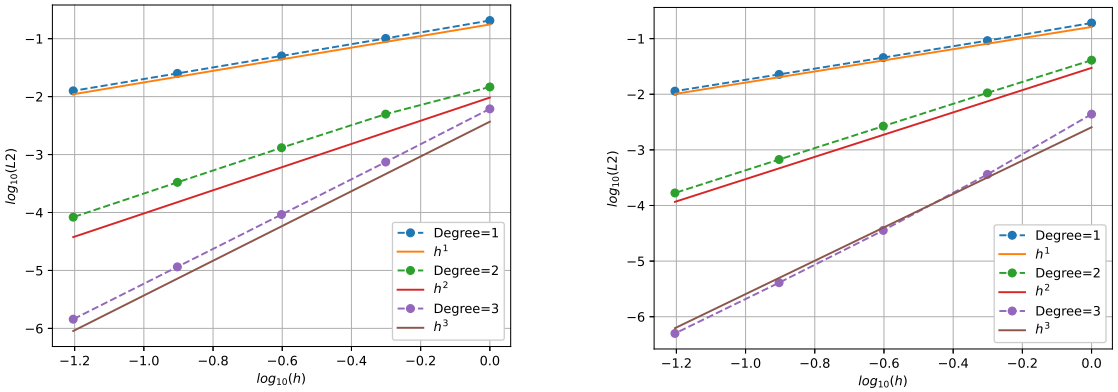


(a) Convergence rate for ϕ^0.

(b) Convergence rate for σ^1.

Figure 11: Convergence rates for ϕ^0 and σ^1 using the standard mixed element method.

The convergence rates for $\hat{\phi}^2$ and $\hat{\sigma}^1$ using the standard mixed element formulation on the primal system have been plotted in Figure 12, where h^s is the convergence rate of an order s system. The convergence rates for $\hat{\phi}^2$ and $\hat{\sigma}^1$ are found to be the same as h^s for a degree of s.

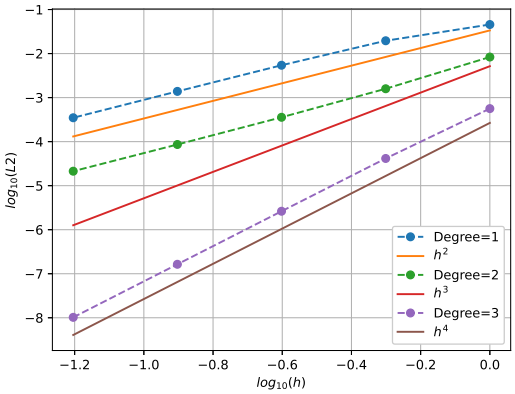


(a) Convergence rates for $\hat{\phi}^2$.

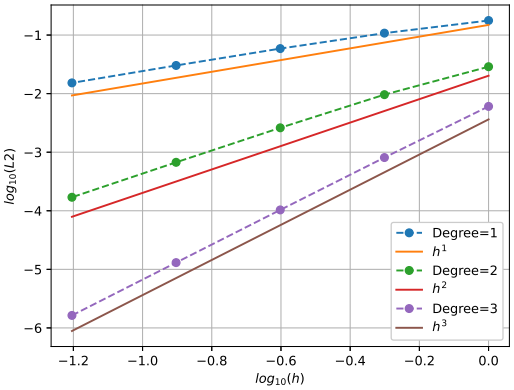
(b) Convergence rates for $\hat{\sigma}^1$.

Figure 12: Convergence rates for $\hat{\phi}^2$ and $\hat{\sigma}^1$ using the standard mixed element method.

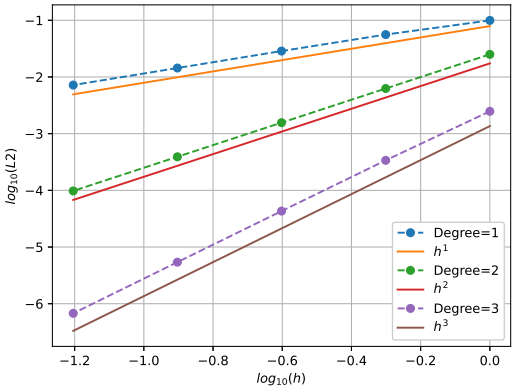
The convergence rates for ϕ^0, σ^1, $\hat{\phi}^2$ and $\hat{\sigma}^1$, on M_{10} and M_{21} respectively, have been plotted in Figure 13. The results can be compared to the convergence rates from the standard mixed element formulation from Figures 11 and 12. The convergence rates are found to be the same for almost all cases. Indicating that the introduction of an interface in the domain has no significant effect on the accuracy of the simulation. The convergence rate of ϕ^0 for a degree of $s = 2$ is found to be h^s instead of h^{s+1}. This is a bit unexpected and there is no obvious explanation. It is not a problem with higher polynomial degrees as $s = 3$ is converging with h^{s+1} again, so it seems it only occurs for the second degree.



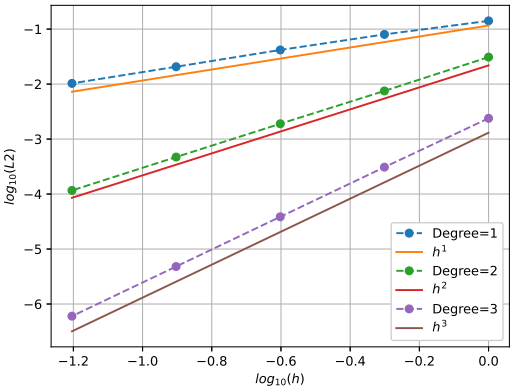
(a) Convergence rates for ϕ^0.



(b) Convergence rates for σ^1.



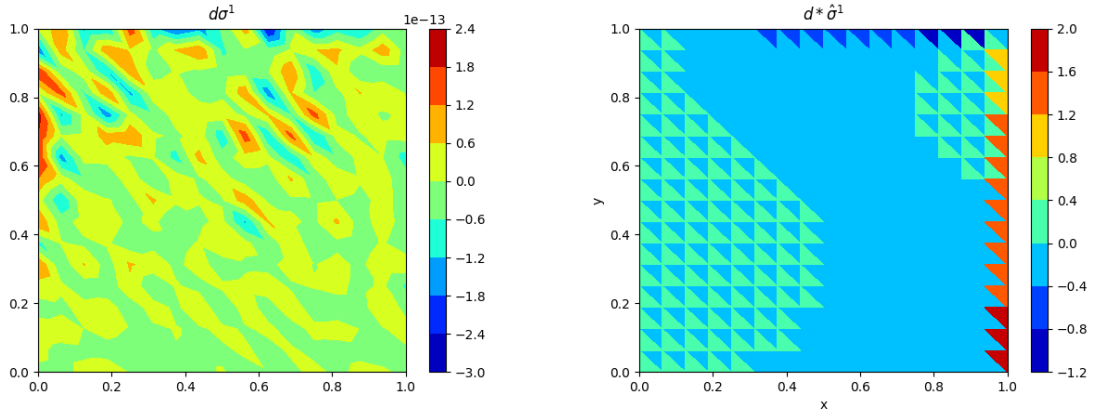
(c) Convergence rates for $\hat{\phi}^2$.



(d) Convergence rates for $\hat{\sigma}^1$.

Figure 13: Convergence rates for ϕ^0, σ^1, $\hat{\phi}^2$ and $\hat{\sigma}^1$ using the domain decomposition method.

In Section 3.1.1 the relation $d\sigma^1 = 0$ was found to be strongly enforced. To check that $d\sigma^1$ is indeed zero everywhere in the domain the value of $d\sigma^1$ has been plotted in Figure 14a. The value of $d\sigma^1$ is found to be in the order of 10^{-13} everywhere in the domain. This means the exterior derivative of σ^1 is zero within machine precision in the entire domain. The opposite is found when doing the same for the primal formulation in Figure 14b, where the $d{*}\hat{\sigma}^1$ has been plotted over the domain. The value is non-zero in large parts of the domain, so the structure is not being preserved when the primal formulation is used.



(a) The curl of σ^1 is within machine precision everywhere on the domain.

(b) The curl of $\hat{\sigma}^1$ is not within machine precision across the domain.

Figure 14: The curl free condition of the Poisson problem is not found for every discretization.

3.3.3 Conclusion

The proposed domain decomposition method using dual field discretization was compared with the standard finite element formulations based on the inner and outer forms respectively. The L^2-errors and convergence rates of the different variables were found the be the same for different polynomial degrees, so the numerical performance of the proposed method is equal to that of the standard finite element formulations. The curl free condition of the Poisson problem was observed within machine precision on a discrete level for the dual formulation. However, for the primal formulation the curl of $\hat{\sigma}^1$ was not found to be zero everywhere in the domain, so the primal formulation cannot be used if the curl free condition is to be preserved.

4 Time dependent Poisson equation

The time dependent Poisson equation on a domain with a Neumann boundary condition on Γ_{10} and a Dirichlet boundary condition on Γ_{21}, as seen in Figure 1. The time dependent Poisson equation in exterior calculus notation is given as

$$\partial_{tt}\phi^0 - *d*d\phi^0 = 0. \tag{78}$$

The second order differential equation is split into two first order differential equations by using the definitions of p^0 and ψ^1

$$p^0 := \partial_t \phi^0, \tag{79}$$
$$\psi^1 := d\phi^0, \tag{80}$$

where $p^0 \in \Omega^0(M)$, $\psi^1 \in \Omega^1(M)$. The boundary conditions are then given as

$$\mathrm{tr} p^0|_{\Gamma_{21}} = u^0 \text{ on } \Gamma_{21}, \tag{81}$$
$$\mathrm{tr}\hat{\psi}^1|_{\Gamma_{10}} = \hat{u}^1 \text{ on } \Gamma_{10}. \tag{82}$$

4.1 Standard finite element formulation

4.1.1 Strong formulations

The strong formulation for the dual system can then be obtained by substituting Equations (79) and (80) in (78), which results in

$$\partial_t p^0 = -d^*\psi^1, \tag{83}$$
$$\partial_t \psi^1 = dp^0, \tag{84}$$

where d^* is the definition of the co-differential, see (10). From (84) the structure $d\partial_t\psi^1 = ddp^0 = 0$ can be found by using the property $dd\alpha = 0$ for $\alpha \in \Omega^0(M)$ [17]. The exterior derivative of ψ^1 remains constant over time, so $d\psi^1 = 0$ at any time instant if the initial conditions are such that $d\psi^1 = 0$ at $t = 0$. This relation is also true for the weak formulation and should be preserved on the discrete level, as is discussed in Section 4.4.3.

Doing the same for the outer forms $\hat{p}^2$ and $\hat{\psi}^1$ to find the primal strong formulation,

$$\partial_t \hat{p}^2 = d\hat{\psi}^1, \tag{85}$$
$$\partial_t \hat{\psi}^1 = -d^*\hat{p}^2. \tag{86}$$

The exterior calculus identity $d\alpha = 0$ for $\alpha \in \Omega^n(M), dim(M) = n$, which states that the exterior derivative of a top form is equal to zero. From that statement it is found that $d\partial_t\hat{p}^2 = 0$. While taking the exterior derivative of a top form is equal to zero, it has no practical meaning. This means that there is no structure preserved weakly in this formulation. The structure that is being strongly enforced is that $\partial_t\hat{p}^2 = d\hat{\psi}^1$.

Strong formulation in vector calculus

The strong formulation in vector calculus can be obtained from $p = \partial_t \phi$ and $\psi = \nabla \phi$, which results in

$$\partial_t p = \nabla \cdot \psi, \tag{87}$$
$$\partial_t \psi = \nabla p. \tag{88}$$

As with the steady case in Section 3.1.1, the strong formulation in vector calculus notation gives no indication of the orientation of the variables.

4.1.2 Weak formulations

The weak formulation for the inner forms is found by taking the inner product of the test functions v^0 and τ^1 with p^0 and ψ^1 respectively. Given $\hat{u}^1 \in H^{-1/2}\hat{\Omega}^1(\Gamma_{10})$, seek $p^0 \in H\Omega^0(M)$ and $\psi^1 \in H\Omega^1(M)$ such that

$$(v^0, \partial_t p^0)_M = -(dv^0, \psi^1)_M + < v^0 | * \psi^1 >_{\Gamma_{10}} \qquad \forall v \in H_0\Omega^0(M, \Gamma_{21}), \tag{89}$$
$$(\tau^1, \partial_t \psi^1)_M = (\tau^1, dp^0)_M \qquad \forall \tau \in H\Omega^1(M), \tag{90}$$

subject to $\mathrm{tr} p^0|_{\Gamma_{21}} = u^0$. The weak formulation can be put into its algebraic form by using the basis function from Section 2.2,

$$\begin{bmatrix} \boldsymbol{M}^0 & 0 \\ 0 & \boldsymbol{M}^1 \end{bmatrix} \begin{pmatrix} \partial_t \boldsymbol{p}^0 \\ \partial_t \boldsymbol{\psi}^1 \end{pmatrix} = \begin{bmatrix} 0 & -(\boldsymbol{D}^0)^T \\ \boldsymbol{D}^0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{p}^0 \\ \boldsymbol{\psi}^1 \end{pmatrix} + \begin{bmatrix} 0 & \boldsymbol{B}^1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^0 \\ \hat{\boldsymbol{u}}^1 \end{pmatrix}. \tag{91}$$

The weak formulation for the primal system is found in a similar way. Given $u^0 \in H^{-1/2}\Omega^0(\Gamma_{21})$, find $\hat{p}^2 \in H\hat{\Omega}^2(M)$ and $\hat{\psi}^1 \in H\hat{\Omega}^1(M)$ such that

$$\left(v^2, \partial_t \hat{p}^2\right)_M = \left(v^2, d\hat{\psi}^1\right)_M \qquad \forall v^2 \in H\Omega^2(M), \tag{92}$$
$$\left(\tau^1, \partial_t \hat{\psi}^1\right)_M = -\left(d\tau^1, \hat{p}^2\right)_M + \left\langle \tau^1 | * \hat{p}^2 \right\rangle_{\Gamma_{21}} \qquad \forall \tau^1 \in H_0\Omega^1(M, \Gamma_{10}), \tag{93}$$

subject to $\mathrm{tr}\hat{\psi}^1|_{\Gamma_{10}} = \hat{u}^1$.

The primal weak formulation can be written its algebraic form using the expressions from Section 2.2,

$$\begin{bmatrix} \boldsymbol{M}^2 & 0 \\ 0 & \boldsymbol{M}^1 \end{bmatrix} \begin{pmatrix} \partial_t \hat{\boldsymbol{p}}^2 \\ \partial_t \hat{\boldsymbol{\psi}}^1 \end{pmatrix} = \begin{bmatrix} 0 & \boldsymbol{D}^1 \\ (-\boldsymbol{D}^1)^T & 0 \end{bmatrix} \begin{pmatrix} \hat{\boldsymbol{p}}^2 \\ \hat{\boldsymbol{\psi}}^1 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ \boldsymbol{B}^0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^0 \\ \hat{\boldsymbol{u}}^1 \end{pmatrix}. \tag{94}$$

Weak formulation in vector calculus

To obtain the weak formulation in vector calculus notation the variables need to be transformed to scalar and vector spaces such that $p \in C^\infty(M)$ and $\psi \in \mathfrak{X}(M)$. For the dual system the equivalent weak formulation in vector calculus, seek $(\phi, \sigma) \in H^1(M) \times H^{curl}(M)$, with $\phi|_{\Gamma_{21}} = u_0$, such that

$$\int_M v \partial_t p dx = -\int_M \nabla v \cdot \psi dx + \int_{\Gamma_{10}} v\psi \cdot n ds \qquad \forall v \in H_0^1(M, \Gamma_{21}), \tag{95}$$

$$\int_M \tau \cdot \partial_t \psi dx = \int_M \tau \cdot \nabla p dx \qquad \forall \tau \in H^{curl}(M). \tag{96}$$

The structure preserving identity $d\partial_t \psi^1 = 0$ can be expressed in vector calculus form as the 2D-curl of $\partial_t \psi$. The vector calculus identities $\nabla \times (A + B) = \nabla \times A + \nabla \times B$ and $\nabla \times (\lambda A) = \lambda(\nabla \times A) + (\nabla \lambda) \times A$ are used to rewrite the expression. The curl of ψ is found to remain constant over time, so for an irrotational flow the initial conditions have to be chosen such that $\nabla \times \psi = 0$,

$$\nabla \times \psi(t + \Delta t) = \nabla \times \psi(t). \tag{97}$$

The primal weak formulation for in vector calculus is as follows. Seek $(\phi, \sigma) \in L^2(M) \times H^{div}(M)$ such that

$$\int_M v \partial_t p dx = \int_M v \nabla \cdot \psi dx \qquad \forall v \in L^2(M), \tag{98}$$

$$\int_M \tau \cdot \partial_t \psi dx = -\int_M p \nabla \cdot \tau dx + \int_{\Gamma_{21}} p\tau \cdot n ds \qquad \forall \tau \in H_0^{div}(M, \Gamma_{10}), \tag{99}$$

subject to $\sigma|_{\Gamma_{10}} \cdot n = u_1$.

4.2 Space discretization of mixed boundary conditions via a domain decomposition method

In this section the time dependent Poisson problem is solved using the dual field discretization. A domain M is decomposed with an internal interface in the same way as described in Section 3.2, resulting in two subdomain M_{10} and M_{21}, as shown in Figure 5. The strong formulations are the same as for the standard formulations, except that the variables are now limited to their respective subdomain, so $\phi^0 \in \Omega^0(M_{10})$, $\sigma^1 \in \Omega^1(M_{10})$, $\hat{\phi}^2 \in \hat{\Omega}^2(M_{21})$ and $\hat{\sigma}^1 \in \hat{\Omega}^1(M_{21})$.

4.2.1 The weak formulations for a decomposed domain

The weak formulations are derived in a similar way to the weak formulations in Section 4.1.2. The difference is that the weak formulations are limited to their respective subdomains and there is now a term on the interface boundary. For the dual system seek $p^0 \in H\Omega^0(M_{10})$ and $\psi^1 \in H\Omega^1(M_{10})$, given

$\hat{u}^1 \in H^{-1/2}(\Gamma_{10})$, such that

$$\left(v^0, \partial_t p^0\right)_{M_{10}} = -\left(dv^0, \psi^1\right)_{M_{10}} + \left\langle v^0 | \hat{\psi}^1 \right\rangle_{\Gamma_{int}} + \left\langle v^0 | * \psi^1 \right\rangle_{\Gamma_{10}} \qquad \forall v^0 \in H\Omega^0(M_{10}), \tag{100}$$

$$\left(\tau^1, \partial_t \psi^1\right)_{M_{10}} = \left(\tau^1, dp^0\right)_{M_{10}} \qquad \forall \tau^1 \in H\Omega^1(M_{10}). \tag{101}$$

The dual weak formulation can be written as sums of the basis functions, as described in Section 2.2, which gives

$$\boldsymbol{M}^0 \partial_t \boldsymbol{p}^0 = (-\boldsymbol{D}^0)^T \boldsymbol{\psi}^1 + \boldsymbol{L}^1_{\Gamma_{int}} \hat{\boldsymbol{\psi}}^1 + \boldsymbol{B}^1 \hat{\boldsymbol{u}}^1, \tag{102}$$

$$\boldsymbol{M}^1 \partial_t \boldsymbol{\psi}^1 = \boldsymbol{D}^0 \boldsymbol{p}^0. \tag{103}$$

For the primal system on the M_{21} subdomain, given $u^0 \in H^{-1/2}\Omega^0(\Gamma_{21})$, find $\hat{p}^2 \in H\hat{\Omega}^2(M_{21})$ and $\hat{\psi}^1 \in H\hat{\Omega}^1(M_{21})$ such that

$$\left(v^2, \partial_t \hat{p}^2\right)_{M_{21}} = \left(v^2, d\hat{\psi}^1\right)_{M_{21}} \qquad \forall v^2 \in H\Omega^2(M_{21}), \tag{104}$$

$$\left(\tau^1, \partial_t \hat{\psi}^1\right)_{M_{21}} = -\left\langle \tau^1 | p^0 \right\rangle_{\Gamma_{int}} - \left(d\tau^1, \hat{p}^2\right)_{M_{21}} + \left\langle \tau^1 | * \hat{p}^2 \right\rangle_{\Gamma_{21}} \qquad \forall \tau^1 \in H\Omega^1(M_{21}). \tag{105}$$

The primal weak formulation can be written in algebraic form using the expressions from Section 2.2,

$$\boldsymbol{M}^2 \partial_t \hat{\boldsymbol{p}}^2 = \boldsymbol{D}^1 \hat{\boldsymbol{\psi}}^1, \tag{106}$$

$$\boldsymbol{M}^1 \partial_t \hat{\boldsymbol{\psi}}^1 = -(\boldsymbol{L}^1_{\Gamma_{int}})^T \boldsymbol{p}^0 - (\boldsymbol{D}^1)^T \hat{\boldsymbol{p}}^2 + \boldsymbol{B}^0 \boldsymbol{u}^0. \tag{107}$$

From the subdomains M_{10} and M_{21} the system of equations can be created by combining Equations (102), (103), (106) and (107). The system has the form of $\boldsymbol{M}\dot{\boldsymbol{x}} = \boldsymbol{J}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$, where $\boldsymbol{M}$ is a diagonal matrix and $\boldsymbol{J}$ is skew-symmetric. The coupling on the interface is done in the $\boldsymbol{J}$ matrix by the $\boldsymbol{L}^1_{\Gamma_{int}}$ and $(\boldsymbol{L}^1_{\Gamma_{int}})^T$ terms,

$$\begin{bmatrix} \boldsymbol{M}^0 & 0 & 0 & 0 \\ 0 & \boldsymbol{M}^1 & 0 & 0 \\ 0 & 0 & \boldsymbol{M}^2 & 0 \\ 0 & 0 & 0 & \boldsymbol{M}^1 \end{bmatrix} \begin{pmatrix} \partial_t \boldsymbol{p}^0 \\ \partial_t \boldsymbol{\psi}^1 \\ \partial_t \hat{\boldsymbol{p}}^2 \\ \partial_t \hat{\boldsymbol{\psi}}^1 \end{pmatrix} = \begin{bmatrix} 0 & -(\boldsymbol{D}^0)^T & 0 & \boldsymbol{L}^1_{\Gamma_{int}} \\ \boldsymbol{D}^0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{D}^1 \\ -(\boldsymbol{L}^1_{\Gamma_{int}})^T & 0 & -(\boldsymbol{D}^1)^T & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{p}^0 \\ \boldsymbol{\psi}^1 \\ \hat{\boldsymbol{p}}^2 \\ \hat{\boldsymbol{\psi}}^1 \end{pmatrix} + \begin{bmatrix} 0 & \boldsymbol{B}^1 \\ 0 & 0 \\ 0 & 0 \\ \boldsymbol{B}^0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^0 \\ \hat{\boldsymbol{u}}^1 \end{pmatrix}. \tag{108}$$

Weak formulation in vector calculus

The weak formulation on the M_{10} subdomain to seek $(\phi_{10}, \sigma_{10}) \in H^1(M_{10}) \times H^{curl}(M_{10})$ such that

$$\int_{M_{10}} v_{10}\partial_t p_{10} dx = -\int_{M_{10}} \psi_{10} \cdot \nabla v_{10} dx + \int_{\Gamma_{int}} v_{10}\psi_{10} \cdot n_{10} ds + \int_{\Gamma_{10}} v_{10}\psi_{10} \cdot n_{10} ds \qquad \forall v_{10} \in H^1(M_{10}), \tag{109}$$

$$\int_{M_{10}} \tau_{10} \cdot \partial_t \psi_{10} dx = \int_{M_{10}} \tau_{10} \cdot \nabla p_{10} dx \qquad \forall \tau_{10} \in H^{curl} * M_{10}). \tag{110}$$

For the weak formulation for the M_{21} subdomain in vector calculus, seek $(\phi_{21}, \sigma_{21}) \in L^2(M_{21}) \times H^{div}(M_{21})$ such that

$$\int_{M_{21}} v_{21}\partial_t p_{21} dx = \int_{M_{21}} v_{21} \nabla \cdot \psi_{21} dx \qquad \forall v_{21} \in L^2(M_{21}), \tag{111}$$

$$\int_{M_{21}} \tau_{21} \cdot \partial_t \psi_{21} dx = -\int_{\Gamma_{int}} p_{21}\tau_{21} \cdot n_{21} ds - \int_{M_{21}} p_{21} \nabla \cdot \tau_{21} dx + \int_{\Gamma_{21}} p_{21}\tau_{21} \cdot n_{21} ds \qquad \forall \tau \in H^{div}(M_{21}). \tag{112}$$

The weak formulation as given in Equations (109), (110), (111) and (112) are not yet coupled on the interface Γ_{int}. The coupling is done in the same way as for the steady case in Section 3.2, specifically Equations (70) and (71).

4.3 Time integration using the implicit midpoint scheme

For the temporal integration scheme some special care has to be taken in choosing a method. The structures preserved by the spatial discretization should also be preserved for the temporal discretization. Most importantly the power balance needs to be preserved. The method used here is the implicit midpoint rule. The implicit midpoint rule is a so called symplectic method, which are well suited for preserving the physical properties of a system [26]. Aoues et al. [27] showed that the implicit midpoint rule is able to preserve the energy exactly on the discrete level for some linear systems. The implicit version has been chosen, instead of the explicit version, because any explicit Runge-Kutta method cannot be symplectic [28]. Furthermore, the implicit midpoint scheme is also unconditionally stable. The midpoint rule used here is a second order system. Although a fourth order midpoint method has been developed [29] and shown to work when implemented [30]. The extra computational demands are not worth it for this work, but the fourth order method should be considered when applying structure preserving schemes to real world applications.

The general form of the temporal problem is given as

$$\boldsymbol{M}\dot{\boldsymbol{x}} = \boldsymbol{J}\boldsymbol{x} + \boldsymbol{G}\boldsymbol{u}, \tag{113}$$

where $\mathbf{x}$ are the unknowns to be solved and $\mathbf{u}$ the inputs. The time derivative and midpoint value of $\mathbf{x}$

are approximated by

$$\dot{\boldsymbol{x}} \approx \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\Delta t}, \tag{114}$$

$$\boldsymbol{x}_{k+\frac{1}{2}} \approx \frac{\boldsymbol{x}_{k+1} + \boldsymbol{x}_k}{2}, \tag{115}$$

where the subscript k indicates the time step.

Substituting Equations (114) and (115) into (113) and rewriting so all the $\mathbf{x}_{k+1}$ terms are on the left and all the $\mathbf{x}_k$ terms on the right, gives the midpoint integration scheme,

$$(\boldsymbol{M} - \frac{\Delta t}{2}\boldsymbol{J})\boldsymbol{x}_{k+1} = (\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{J})\boldsymbol{x}_k + \Delta t \boldsymbol{G}\boldsymbol{u}_{k+\frac{1}{2}}. \tag{116}$$

The expressions for $\boldsymbol{M}$, $\boldsymbol{J}$ and $\boldsymbol{G}$ are taken from the system in (108). Substituting these expressions into (116) gives a linear problem of the form $a_{k+1} = L_k$.

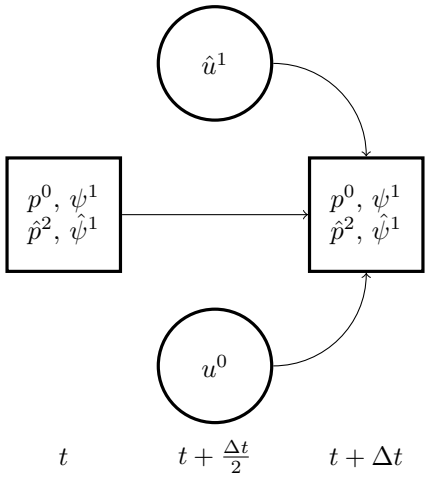


Figure 15: Flow chart midpoint integration method.

4.4 Numerical simulations

Now that the spatial and temporal discretizations for the time dependent Poisson problem have been explained, numerical simulations have to be conducted to test the performance of the proposed scheme. First an analytical solution is provided to analyze the L^2-error and subsequently the convergence rates of the different schemes. The numerical performance of the proposed scheme can then be compared to the standard finite element formulations. Furthermore, the curl free condition of the Poisson equation is investigated for each scheme by plotting the curl of σ over time. The power balance is then plotted for each simulation to determine if the energy is preserved on a discrete level.

The simulation are again performed on a unit square as shown for the decomposed domain in Figure 6a. The mesh is a structured triangular mesh as shown in Figure 6b. The finite elements are the same as for the steady case, the Continuous Galerkin element for the 0-form ϕ^0, the Nédélec first kind H(curl) element for the inner 1-form σ^1, the Discontinuous Galerkin element for the 2-form $\hat{\phi}^2$ and a Raviart-Thomas

element for the outer 1-form $\hat{\sigma}^1$. A time step of $\Delta t = \frac{1}{1000}$ has been used for all simulation. The time step was chosen such that the spatial error is dominant over the temporal error. The open source finite element solver Firedrake [23] has been used to simulate the time dependent Poisson problem. A more detailed explanation of the simulation set up can be found in Appendix A.1.

4.4.1 Analytical solution

The analytical solution can be constructed from the temporal and spatial parts of the possible solutions,

$$f(t) = 2\sin(\omega_t t) + 3\cos(\omega_t t), \tag{117}$$
$$g(x,y) = \cos(\omega_x x)\sin(\omega_y y) \tag{118}$$

where $\omega_t = \sqrt{\omega_x^2 + \omega_y^2}$. The exact solutions will then have to form of

$$\begin{aligned} p^0_{ex} &= g\frac{df}{dt}, & \hat{p}^2_{ex} &= *g\frac{df}{dt} \\ \psi^1_{ex} &= fdg, & \hat{\psi}^1_{ex} &= *fdg \end{aligned} \tag{119}$$

The analytical solution used here is obtained by taking $\omega_x = \omega_y = 1$. The boundary conditions are taken to be the exact solution on the boundary, such that $u^0 = \mathrm{tr} p^0_{ex}|_{\Gamma_{21}}$ and $\hat{u}^1 = \mathrm{tr}\hat{\psi}^1_{ex}|_{\Gamma_{10}}$.

Analytical solution in vector calculus

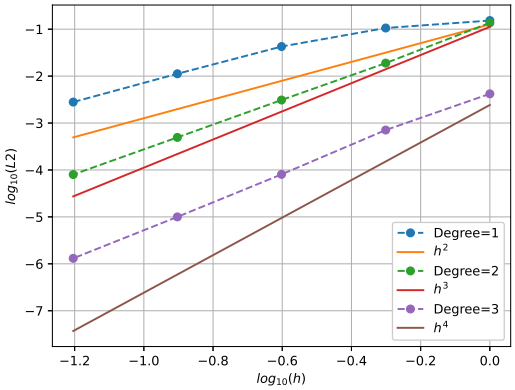
The analytical solution can be obtained in vector calculus by translating the exterior calculus forms into scalars and vector functions. $p^0_{ex} \to \mathcal{F}$ is identically the scalar, while $*\hat{p}^2_{ex} \to \mathcal{F}$. The inner 1-from translates to a vector function by $\sharp\psi^1_{ex} \to \mathfrak{X}$ and the outer form by $\sharp*\hat{\psi}^1_{ex} \to \mathfrak{X}$, see Figures 4a and 4b. This means the values of p^0 and ψ^1 are the same as $\hat{p}^2$ and $\hat{\psi}^1$ in vector calculus, so the expression for the analytical solutions are

$$p_{ex} = g\frac{df}{dt}, \tag{120}$$
$$\psi_{ex} = f\nabla g. \tag{121}$$

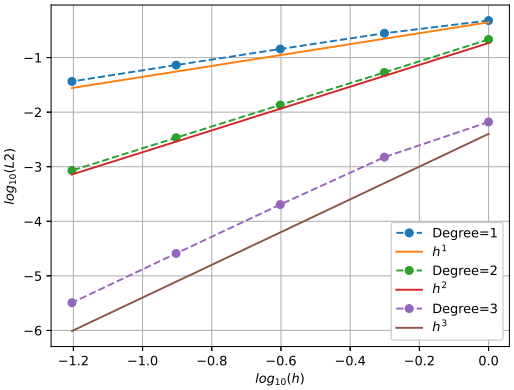
4.4.2 Results

To ensure the simulations are behaving as expected the convergence rates for the conventional method and the domain decomposition method are analyzed. Since the same problem is solved on the same mesh, the effect of the interface on the L^2-error can also be found by comparing the errors of the two methods. The convergence rates of p^0 and ψ^1 obtained from the conventional method have been plotted in Figure 16. Where h^s are the theoretical convergence rates for a polynomial degree of s. A polynomial degree of $s = 1$ is a first order spatial integration method, meaning that if the value of h is halved so is the L^2-error. The convergence rates for ψ^1 in Figure 16b are as expected. The convergence rate for p^0 is seen to be h^{s+1} for a polynomial degree of $s = 1$. The convergence rate of 0-forms is h^{s+1} for homogeneous boundary conditions [25], so finding a convergence rate of h^{s+1} for mixed boundary conditions is not unexpected. The convergence rates for p^0 does seem to slow down for higher polynomial degrees. For a

polynomial degree of $s = 2$ the convergence rate is between h^s and h^{s+1}, and for $s = 3$ the convergence rate is observed to be h^s.



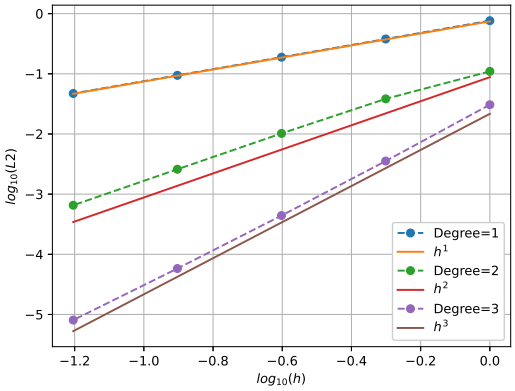
(a) Convergence rate for p^0.



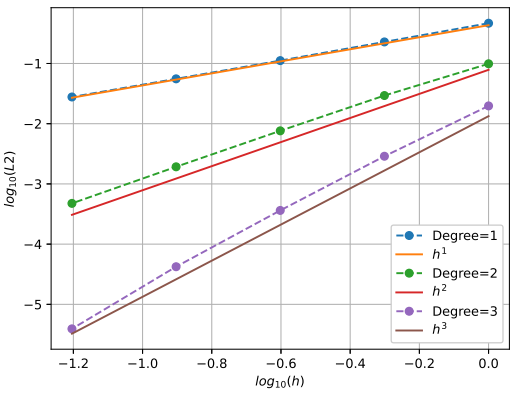
(b) Convergence rate for ψ^1.

Figure 16: Convergence rates for p^0 and ψ^1 for the conventional method.

The convergence rates for the primal formulations are shown in Figure 17. The convergence rates for $\hat{p}^2$ and $\hat{\psi}^1$ agree with the theoretical convergence rates of h^s.



(a) Convergence rate for $\hat{p}^2$.

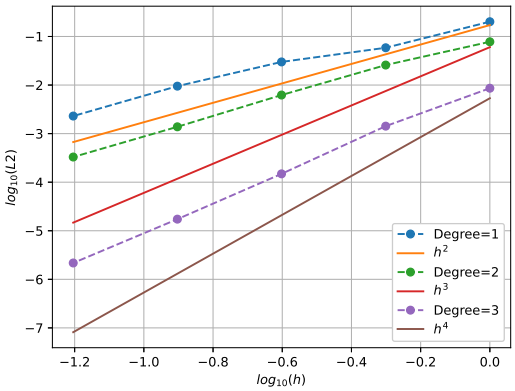


(b) Convergence rate for $\hat{\psi}^1$.

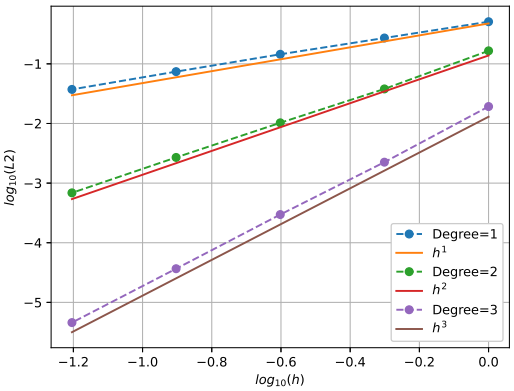
Figure 17: Convergence rates for p^0 and ψ^1 for the conventional method.

The convergence rates for p^0, ψ^1, $\hat{p}^2$ and $\hat{\psi}^1$ from the domain decomposition method have been plotted in Figure 18. The convergence rate of p^0 in Figure 18a is again h^{s+1} for a polynomial degree of $s = 1$. For the higher degrees $s = 2$ and $s = 3$ the convergence rates are found to be h^s. The convergence rate for $s = 2$ is again slower when compared to the standard finite element formulation in Figure 16a. The same result was found for the steady case in Section 3.3. The convergence rate of ψ^1 on M_{21} is shown in

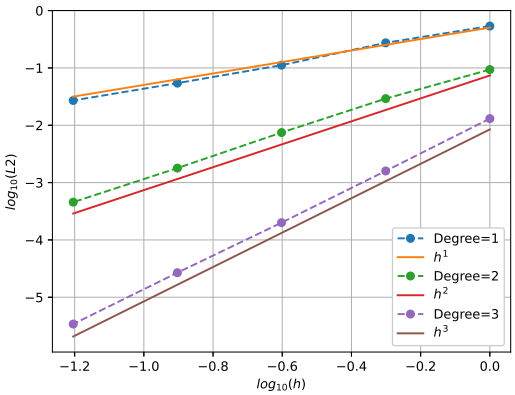
Figure 18b, and $\hat{p}^2$ and $\hat{\psi}^1$ on M_{21} can be seen in Figures 18c and 18d respectively. All three variables show a convergence rate equal to their polynomial degree, e.g. $s = 1$ gives h^1.



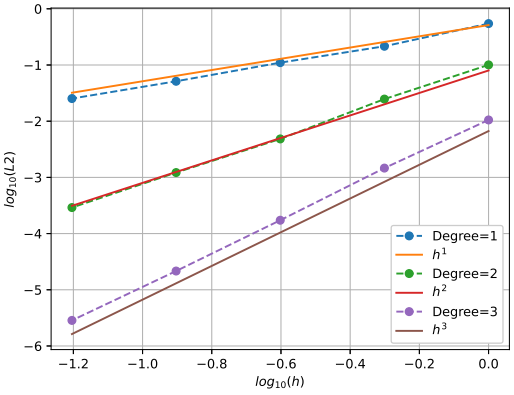
(a) Convergence rate of p^0.



(b) Convergence rate of ψ^1.



(c) Convergence rate of $\hat{p}^2$.

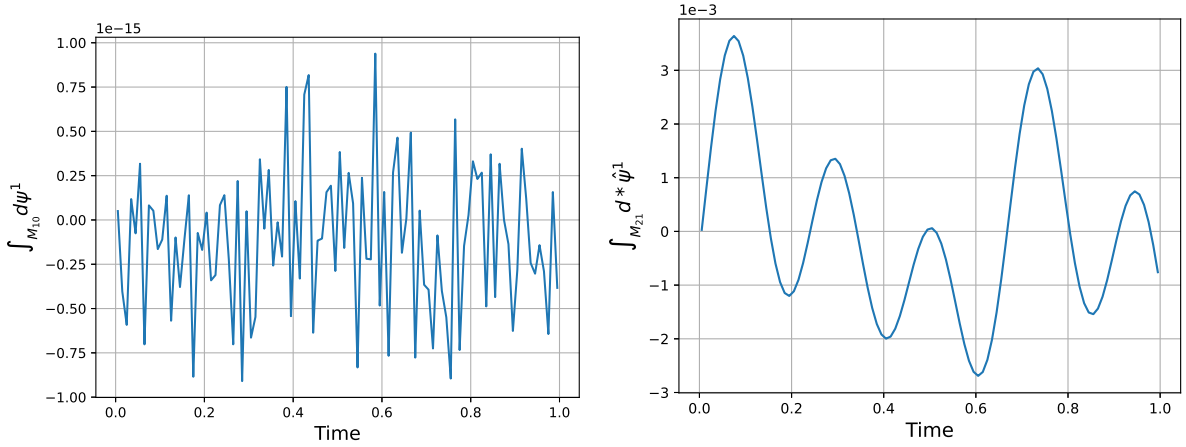


(d) Convergence rate for $\hat{\psi}^1$.

Figure 18: Convergence rates for p^0, ψ^1, $\hat{p}^2$ and $\hat{\psi}^1$ for the domain decomposition method.

4.4.3 Curl free flow

From the strong formulation of the dual formulation in (84), the relation $d\partial_t\psi^1 = 0$ can be found. The vector calculus equivalent for taking an exterior derivative of a 1-form in 2D, is to take the 2D-curl of ψ. The 2d-curl over time on the subdomain M_{10} can be seen to be within machine precision in Figure 19a. To make sure that the 2d-curl is non-zero on the M_{21} domain, $d*\hat{\psi}^1$ is calculated on M_{21}. The hodge operator is necessary to transfer to the inner orientation, as the 2d-curl is only present in the inner orientation in two dimensions. In the Firedrake code, this means the values of $\hat{\psi}^1$ need to be interpolated from a Raviart-Thomas element to a Nédélec first kind H(curl) element before calculating the 2d-curl. The 2d-curl of ψ over time is shown in Figure 19b. The curl is observed to be a smooth function with magnitude of 10^{-3}, so it is not within machine precision.



(a) $d\psi^1$ over time on the subdomain M_{10}.

(b) $d*\hat{\psi}^1$ on the subdomain M_{21}.

Figure 19: $d\psi^1$ and $d*\psi^1$ from the domain decomposition method.

4.4.4 Power Balance

The energy contribution of a k-form in the domain is given by

$$H_k = \frac{1}{2}\left(\alpha^k, \alpha^k\right)_M . \tag{122}$$

The change of energy over time is then

$$\dot{H}_k = \left(\alpha^k, \partial_t \alpha^k\right)_M . \tag{123}$$

The weak formulations in (89) and (90) are valid for any v^0 and v^1 , so v^0 and v^1 are set to p^0 and ψ^1 respectively. This results in

$$\left(p^0, \partial_t p^0\right)_M + \left(u^1, \partial_t u^1\right)_M = \left\langle p^0 | \hat{\psi}^1 \right\rangle_{\partial M} \tag{124}$$

Using (123) and $\dot{H}_{10} = \dot{H}_1 + \dot{H}_0$, it can be rewritten as

$$\dot{H}_{10}^M = \left\langle p^0 | \hat{\psi}^1 \right\rangle_{\partial M} , \tag{125}$$

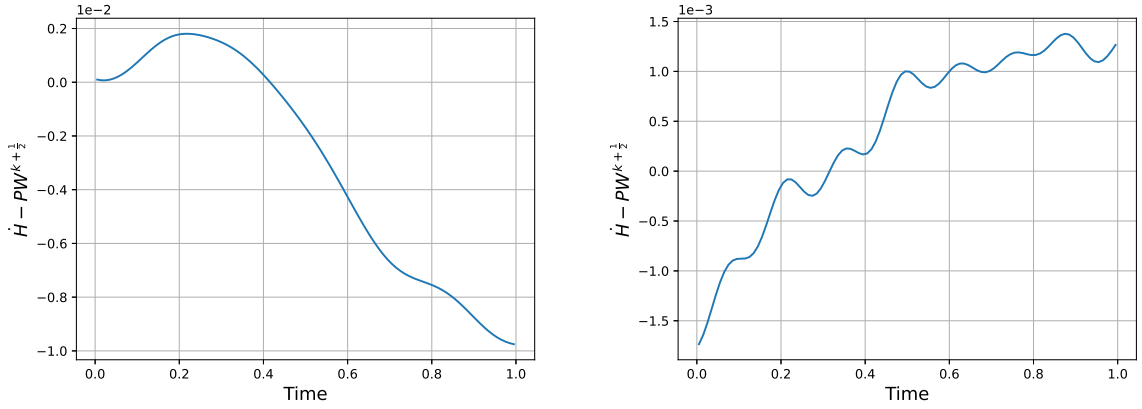
where the superscript M indicates the domain. So the energy balance is preserved on the continuum level, for both weak formulations. On the discrete level the energy conservation can be checked by setting the test functions to the midpoint value, e.g. v^0 to $\frac{p_0^{k+1}+p_0^k}{2}$, where the subscript now indicates the form. The discrete energy balance is

$$\left(\frac{p_0^{k+1}+p_0^k}{2}, \frac{p_0^{k+1}-p_0^k}{\Delta t}\right)_M + \left(\frac{\psi_1^{k+1}+\psi_1^k}{2}, \frac{\psi_1^{k+1}-\psi_1^k}{\Delta t}\right)_M = \left\langle p_0^{k+\frac{1}{2}} | \hat{\psi}_1^{k+\frac{1}{2}} \right\rangle_{\partial M} . \tag{126}$$

Doing the same derivation for the primal formulation on the entire domain M, the discrete power balance is found to be

$$\left(\frac{\hat{p}_2^{k+1}+\hat{p}_2^k}{2}, \frac{\hat{p}_2^{k+1}-\hat{p}_2^k}{\Delta t}\right)_M + \left(\frac{\hat{\psi}_1^{k+1}+\hat{\psi}_1^k}{2}, \frac{\hat{\psi}_1^{k+1}-\hat{\psi}_1^k}{\Delta t}\right)_M = \left\langle p_0^{k+\frac{1}{2}} | \hat{\psi}_1^{k+\frac{1}{2}} \right\rangle_{\partial M} . \tag{127}$$

The power balance can be preserved on a discrete level when using the correct method. For the conventional problem the essential boundary condition causes issues. The right hand sides of (127) shows that the inputs at the boundary need to be evaluated at the midpoint. However, when applying an essential boundary condition the value is set to the integer time step $k+1$. This causes the energy conservation to break down on a discrete level, as can be seen in Figure 20. The application of an essential boundary condition actually solves for a different power balance [17].



(a) Power balance for dual formulation

(b) Power balance for primal formulation

Figure 20: The power balance for the standard method.

The domain decomposition method does not contain any essential boundary conditions, so the problems caused by them do not occur. The introduction of an interface does create two subdomains that need to be checked for energy conservation. Due to the interface, the terms on the boundary for each subdomain now contain a duality product on the interface,

$$PW_{M_{10}}^{k+\frac{1}{2}} = \left\langle p_0^{k+\frac{1}{2}} | \hat{\psi}_1^{k+\frac{1}{2}} \right\rangle_{\Gamma_{10}} + \left\langle p_0^{k+\frac{1}{2}} | \hat{\psi}_1^{k+\frac{1}{2}} \right\rangle_{\Gamma_{int}}, \tag{128}$$

$$PW_{M_{21}}^{k+\frac{1}{2}} = \left\langle p_0^{k+\frac{1}{2}} | \hat{\psi}_1^{k+\frac{1}{2}} \right\rangle_{\Gamma_{21}} + \left\langle p_0^{k+\frac{1}{2}} | \hat{\psi}_1^{k+\frac{1}{2}} \right\rangle_{\Gamma_{int}}. \tag{129}$$

The energy change in the M_{10} domain $\dot{H}_{10}^{M_{10}}$, can be determined in similar way as $\dot{H}_{10}^{M}$, by doing the same for the weak formulations in Equations (100) and (101), now limited to the subdomain M_{10}. The discrete power balances are then

$$\left(\frac{p_0^{k+1} + p_0^k}{2}, \frac{p_0^{k+1} - p_0^k}{\Delta t} \right)_{M_{10}} + \left(\frac{\psi_1^{k+1} + \psi_1^k}{2}, \frac{\psi_1^{k+1} - \psi_1^k}{\Delta t} \right)_{M_{10}} = \left\langle p_0^{k+\frac{1}{2}} | \hat{\psi}_1^{k+\frac{1}{2}} \right\rangle_{\partial M_{10}}, \tag{130}$$

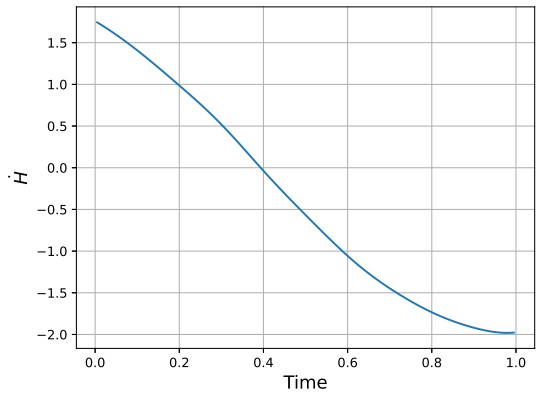
$$\left(\frac{\hat{p}_2^{k+1} + \hat{p}_2^k}{2}, \frac{\hat{p}_2^{k+1} - \hat{p}_2^k}{\Delta t} \right)_{M_{21}} + \left(\frac{\hat{\psi}_1^{k+1} + \hat{\psi}_1^k}{2}, \frac{\hat{\psi}_1^{k+1} - \hat{\psi}_1^k}{\Delta t} \right)_{M_{21}} = \left\langle p_0^{k+\frac{1}{2}} | \hat{\psi}_1^{k+\frac{1}{2}} \right\rangle_{\partial M_{21}}, \tag{131}$$

for the M_{10} and M_{21} subdomains respectively. The discrete power balance for the M_{10} and M_{21} domains are then preserved on a discrete level if Equations (130) and (131) hold true. The discrete power balance

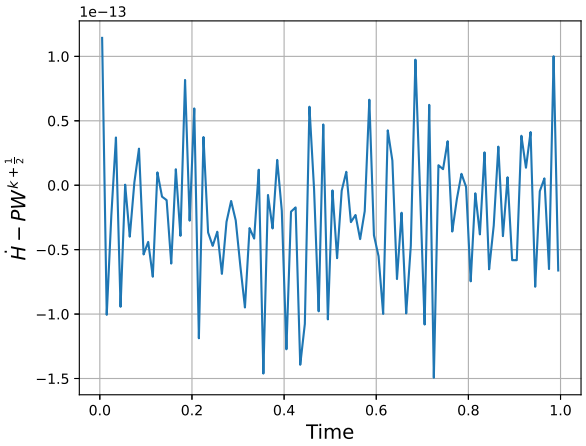
for the entire system can then be found by adding Equations (130) and (131) together, which results in

$$\left(\frac{p_0^{k+1}+p_0^k}{2},\frac{p_0^{k+1}-p_0^k}{\Delta t}\right)_{M_{10}}+\left(\frac{\psi_1^{k+1}+\psi_1^k}{2},\frac{\psi_1^{k+1}-\psi_1^k}{\Delta t}\right)_{M_{10}}$$
$$+\left(\frac{\hat{p}_2^{k+1}+\hat{p}_2^k}{2},\frac{\hat{p}_2^{k+1}-\hat{p}_2^k}{\Delta t}\right)_{M_{21}}+\left(\frac{\hat{\psi}_1^{k+1}+\hat{\psi}_1^k}{2},\frac{\hat{\psi}_1^{k+1}-\hat{\psi}_1^k}{\Delta t}\right)_{M_{21}}=\left\langle p_0^{k+\frac{1}{2}}|\hat{\psi}_1^{k+\frac{1}{2}}\right\rangle_{\Gamma_{10}}+\left\langle p_0^{k+\frac{1}{2}}|\hat{\psi}_1^{k+\frac{1}{2}}\right\rangle_{\Gamma_{21}}, \tag{132}$$

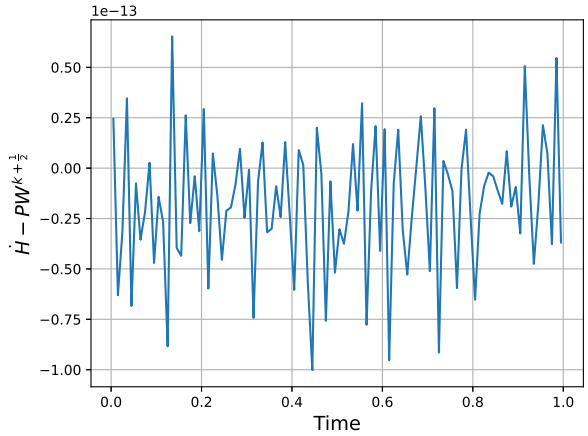
where the interface terms on the right hand sides have disappear, due to the minus introduced by $\Gamma_{int10}=-\Gamma_{int21}$.



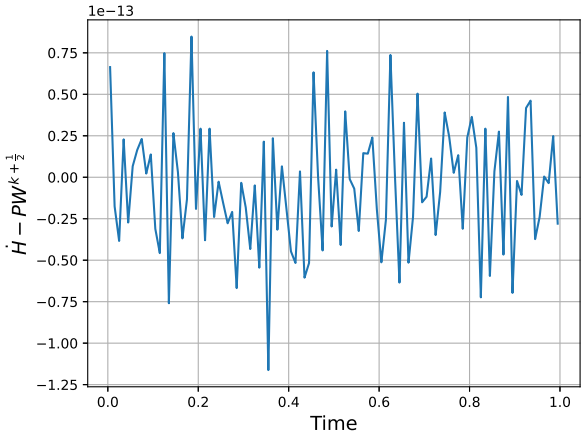
(a) The $\dot{H}$ for the total domain M.



(b) The power balance for the whole domain M.



(c) Power balance for M_{10}.



(d) Power balance for M_{21}.

Figure 21: The energy conservation for the domain decomposition method.

The energy change over time $\dot{H}$, and the power balances for M_{10}, M_{21} and the total domain M have been plotted in Figures 21. Figure 21a shows how the energy changes over time in the entire domain

M. The difference between $\dot{H}$ and $PW^{k+\frac{1}{2}}$ over the entire domain M can be seen in Figure 21b. The results is random noise in the order of 10^{-13}, which means the error is within machine precision, so the power balance is preserved exactly. The results for the power balances of the subdomains M_{10} and M_{21}, have been plotted in Figures 21c and 21d respectively. The errors are again random values in the order of 10^{-13} for both subdomains, so the power balance is being preserved exactly. The results from Figures 21 show that use of domain decomposition makes it possible to preserve the power balance on a discrete level.

4.4.5 Conclusion

The time dependent Poisson equation was discretized for the dual and primal systems using a standard finite element formulation. For the proposed scheme the time dependent Poisson equation was discretized using the dual field method on a decomposed domain. The L^2-errors and convergence rates for the four different variables were determined. The proposed method was found to have no significant difference compared to the standard finite element methods when it comes to the numerical performance of the schemes. The inner 0-form showed a convergence rate of h^{s+1}, while the other forms had a convergence rate of h^s.

The integral of the curl of σ^1 and $\hat{\sigma}^1$ were calculated on the subdomain M_{10} and M_{21} respectively. The curl on the subdomain M_{10}, using the using the dual formulation, was found to be within the order of 10^{-15} over time. The curl on the M_{21} domain, which used the dual formulation, was not zero over time. The curl free condition was thus valid on a discrete level on the subdomain which used the dual formulation, but not on the subdomain which used the primal formulation. The result of this is that part of the domain will never meet the curl free condition if the domain decomposition method is used.

The power balance was determined for both the standard finite element formulations and the proposed dual field formulation. The power balance was not maintained for either of the standard finite element formulations. The power balance for the proposed method was found to be within machine precision on the subdomains M_{10} and M_{21}. On the coupled domain M the energy balance remained within the order of 10^{-13} over time, so it can be said that the energy is preserved on a discrete level for the proposed method.

5 Time Staggering

5.1 Time integration using a staggering implicit midpoint scheme

The spatial integration for the time staggering method is similar as in Section 4. The weak formulations are again Equations (100) and (101) for the M_{10} domain and (104) and (105) for the M_{21} domain. The subdomains are solved separately so the algebraic forms are set up for each subdomain. The difference with the monolithic method in (108), is that the coupling on the interface no longer occurs in the **J** matrix. The coupling terms on the interface become an input in a separate matrix instead,

$$\begin{bmatrix} \boldsymbol{M}^0 & 0 \\ 0 & \boldsymbol{M}^1 \end{bmatrix} \begin{pmatrix} \partial_t \boldsymbol{p}^0 \\ \partial_t \boldsymbol{\psi}^1 \end{pmatrix} = \begin{bmatrix} 0 & -(\boldsymbol{D}^0)^T \\ \boldsymbol{D}^0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{p}^0 \\ \boldsymbol{\psi}^1 \end{pmatrix} + \begin{bmatrix} 0 & \boldsymbol{L}^1_{\Gamma_{int}} \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \hat{\boldsymbol{p}}^2 \\ \hat{\boldsymbol{\psi}}^1 \end{pmatrix} + \begin{bmatrix} 0 & \boldsymbol{B}^0_{\Gamma_{10}} \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^0 \\ \hat{\boldsymbol{u}}^1 \end{pmatrix}, \tag{133}$$

$$\begin{bmatrix} \boldsymbol{M}^2 & 0 \\ 0 & \boldsymbol{M}^1 \end{bmatrix} \begin{pmatrix} \partial_t \hat{\boldsymbol{p}}^2 \\ \partial_t \hat{\boldsymbol{\psi}}^1 \end{pmatrix} = \begin{bmatrix} 0 & \boldsymbol{D}^1 \\ -(\boldsymbol{D}^1)^T & 0 \end{bmatrix} \begin{pmatrix} \hat{\boldsymbol{p}}^2 \\ \hat{\boldsymbol{\psi}}^1 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ -(\boldsymbol{L}^1_{\Gamma_{int}})^T & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{p}^0 \\ \boldsymbol{\psi}^1 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ \boldsymbol{B}^0_{\Gamma_{21}} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^0 \\ \hat{\boldsymbol{u}}^1 \end{pmatrix}. \tag{134}$$

The M_{10} domain has been chosen to be determined at the integer time steps, e.g. k. The M_{21} domain is calculated at the half time steps, e.g. $k + \frac{1}{2}$. The initial conditions for p^0 and ψ^1 on M_{10} are set to p^0_{ex} and ψ^1_{ex} at time $t = 0$. The initial conditions for $\hat{p}^2$ and $\hat{\psi}^1$ on M_{21} are set to $\hat{p}^2_{ex}$ and $\hat{\psi}^1$ at time $t = \frac{\Delta t}{2}$. The results of the M_{10} domain are calculated first for the time step $k + 1$. They can then be used as the input on the interface to determine the results for M_{21} at time step $k + \frac{3}{2}$. Which can then be used in turn as an input for M_{10} for the time step $k + 2$. A flow chart of the time integration can be seen in Figure 22

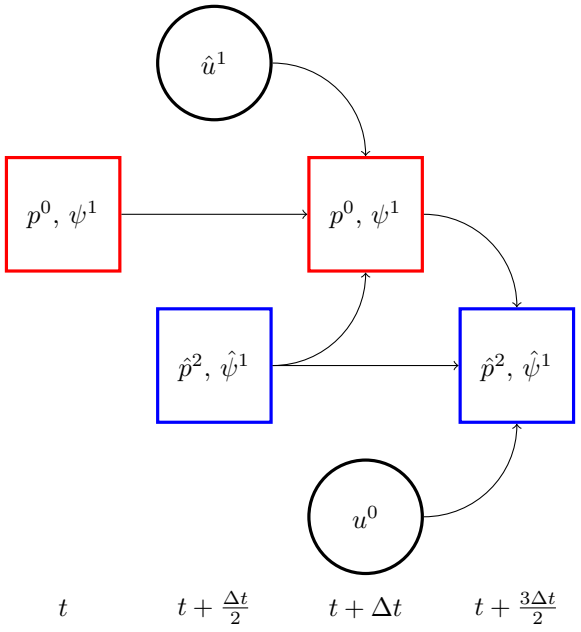
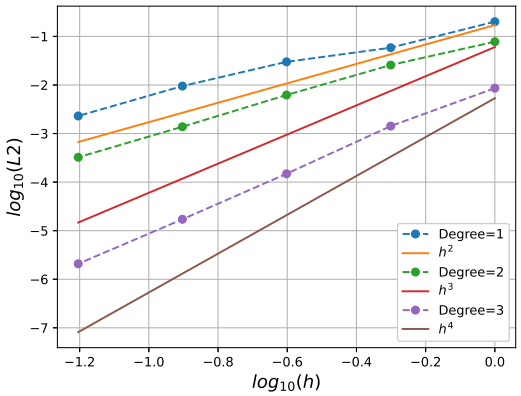


Figure 22: A flow chart of the staggered time integration.

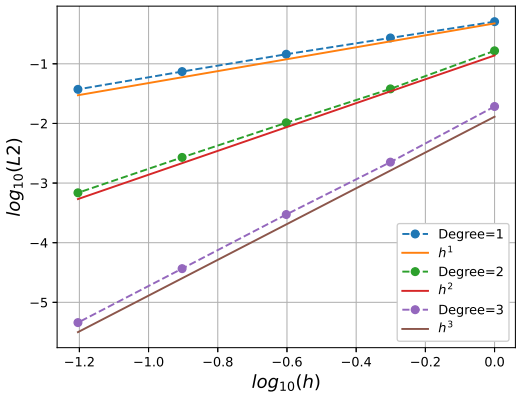
5.2 Numerical simulations

5.2.1 Convergence

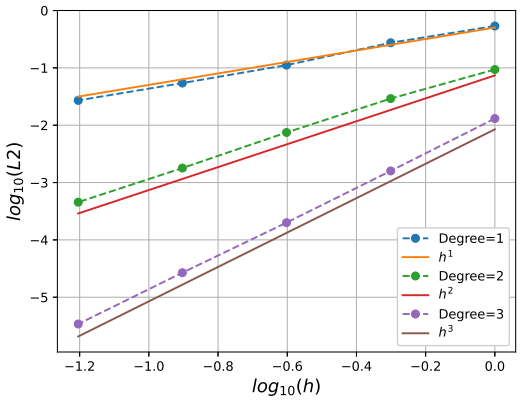
The convergence rates for p^0 and ψ^1 on M_{10} and $\hat{p}^2$ and $\hat{\psi}^1$ on M_{21} have been plotted in Figures 23. The convergence rates are as expected, with the 0-form p^0 as h^{s+1} for a degree of $s = 1$ and h^s for a degree of $s = 2$ and $s = 3$. The other convergence rates for ψ^1, $\hat{p}^2$ and $\hat{\psi}^1$ are h^s. The results are similar to the convergence rates from the non-staggering method in Figures 18, so the staggered method does not sacrifice on accuracy when compared to the monolithic method.



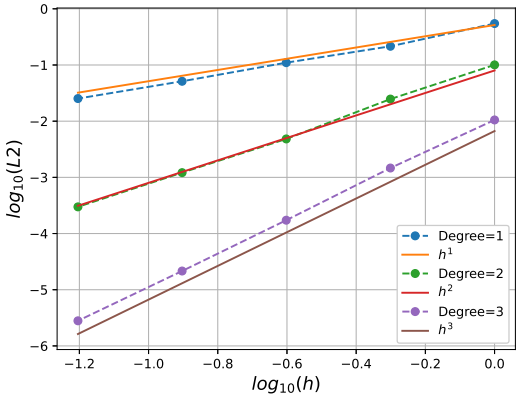
(a) Convergence rate for p^0.



(b) Convergence rate for ψ^1.



(c) Convergence rate for $\hat{p}^2$.

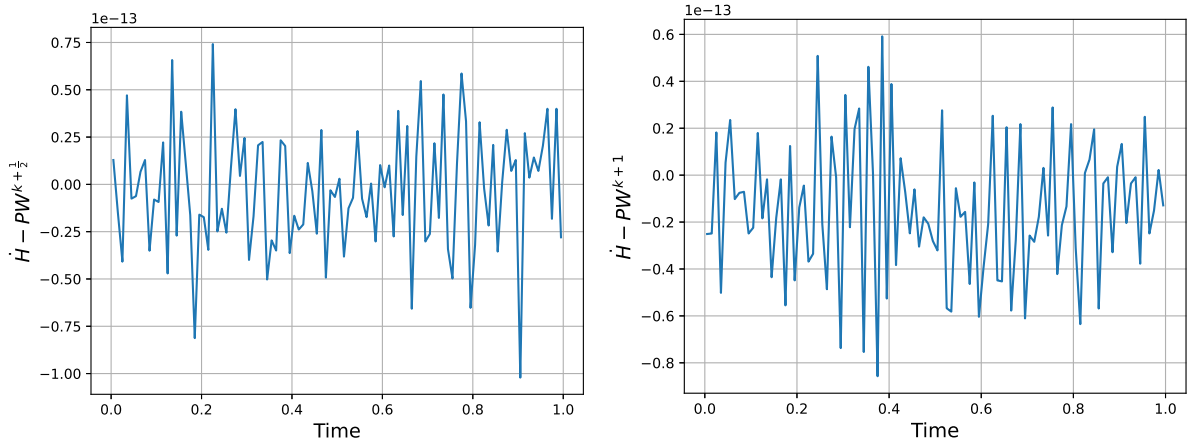


(d) Convergence rate for $\hat{\psi}^1$.

Figure 23: Convergence rates for the time staggering method.

5.2.2 Power balance

The power balance is determined in the same way as in Section 4.4.4, only (131) is shifted by half a time step due to the staggering. The left hand side of (131) is now determined at the half time step and the right hand side at the integer time step. The power balance for the subdomains M_{10} and M_{21} are shown in Figures 24a and 24b respectively. The power balance is conserved within machine precision for both subdomains.



(a) Power balance for the M_{10} domain.

(b) Power balance for the M_{21} domain.

Figure 24: The power balance for the time staggering scheme.

5.2.3 Computational cost

There are a few reason to implement a time staggering scheme instead of a monolithic scheme as in Section 4. The main reason is time staggering reduces the computational cost for each (half) time step. Comparing (108) with (133) and (134), the size of the matrix in (108) is much bigger than individual matrices from the time staggering method. When a program like Firedrake solves the problem, it needs to invert these matrices. A smaller matrix is less computationally demanding than a larger matrix. This makes it more efficient when compared with other time staggering methods which do not use domain decomposition. The methods without domain decomposition calculate the degrees of freedom over the entire domain each half time step, instead of only on part of the domain.

How much the computational cost is reduced by staggering is hard to predict, because it depends on multiple factors. Firstly, how the domain is decomposed. The interface is an arbitrary line that splits the domain into subdomains, so these subdomains do not have to be of the same size. Secondly, even if the subdomains are of the same size, they could be using different finite elements. For instance, p^0 uses Continuous Galerkin (CG) elements while $\hat{p}^2$ uses Discontinuous Galerkin (DG) elements. For a single element with the lowest polynomial degree, the CG elements will have three degrees of freedom, while the DG will have only one [15]. Lastly, programs like Firedrake are not simply inverting the matrices, the code is instead using highly optimized methods. How these methods work and how they effect the computational cost is beyond the scope if this work.

The size of the matrix is dependent on the number of degrees of freedom that need to be solved for. To get an idea of the computational cost of each method the number of degrees of freedom solved at each (half) time step is investigated. The relation between geometry and degrees of freedom is used here. Namely that the degrees of freedom depend on the number of faces, edges and vertices. The mesh used is a structured triangular mesh, this allows for some general expressions for the number of faces, edges and vertices.

The number of faces on the entire domain is found to be $F_M = 2h^2$ and the number of vertices $V_M = (h+1)^2$. The Euler characteristic for a 2-dimensional orientable manifold with no holes and one border can then be used to relate the number of faces and vertices to the number of edges [31],

$$E = F + V - 1. \tag{135}$$

The number of edges, faces and vertices of a surface in $\mathbb{R}^2$ can then be related to each other, which can then used to determine the number of edges for the entire domain,

$$E_M = 3h^2 + 2h. \tag{136}$$

When the domain is decomposed, the number of faces in half the domain, e.g. M_{10}, is $F_{M_{10}} = h^2$ and the number of vertices is $V_{M_{10}} = \sum_{n=1}^{h+1} n = \frac{(h+1)(h+2)}{2}$. Using the Euler characteristic from (135) the number of edges can be determined,

$$E_{M_{10}} = \frac{3h^2 + 3h}{2}. \tag{137}$$

Now that the number of Faces, edges and vertices are known, the number of degrees of freedom can be determined. The Continuous Galerkin (CG) element of a degree of 1 has a number of degrees of freedom equal to the number of vertices. The Nédélec first kind H(curl) (N1curl) element and the Raviart-Thomas (RT) elements of degree 1 a number of degrees of freedom equal to the number of edges. The Discontinuous Galerkin (DG) elements of degree 0 have the number of faces as the number of degrees of freedom.

The dual formulation uses CG and N1curl, so on a square domain it has $N_{10} = 4h^2 + 4h + 1$ degrees of freedom, on half the domain it will have $N_{10}^{half} = 2h^2 + 3h + 1$. The primal formulation, which uses DG and RT elements, has $N_{21} = 5h^2 + 2h$ degrees of freedom on the entire domain and $N_{21}^{half} = \frac{5}{2}h^2 + \frac{3}{2}h$ on half the domain. From these equation we find $N_{10} < N_{21}$ for $h \geq 3$ and similarly $N_{10}^{half} < N_{21}^{half}$ for $h \geq 4$, so the primal formulation uses more degrees of freedom for finer meshes. This is because the number of faces, associated with DG elements, grows faster with decreasing mesh size.

One might expect that if the domain is cut in half, the number of degrees of freedom is also halved. However, it can be determined that $\frac{1}{2}N_{10} = N_{10}^{half} - h - \frac{1}{2}$ and $\frac{1}{2}N_{21} = N_{21}^{half} - \frac{1}{2}h$, thought for large values of h it does tends towards half the number of degrees of freedom. For the decomposed domain without time staggering we find $N_{10} < N_{10}^{half} + N_{21}^{half} < N_{21}$. In fact since the interface is an arbitrary line, it can be places closer and closer to Γ_{10} until $M_{10} \cap M = M$, and similarly for the other domain until $M_{21} \cap M = M$. This means that for the decomposed domain without time staggering, the number of degrees of freedom will always be between N_{10} and N_{21}.

To summarize, for a two dimensional structured triangular mesh, the primal formulations requires more degrees of freedom than the dual formulation. For the domain decomposition without time staggering, the number of degrees of freedom will be larger than using only a dual formulation, but smaller compared to only using a primal formulation. Lastly, for the time staggering method the number of degrees of freedom for each subdomain is nearly halved. The dual formulation also uses less degrees of freedom than the primal formulation when the domain is halved.

6 Conclusion

A numerical scheme is proposed where the domain was decomposed such that for each subdomain the boundary conditions are naturally incorporated. The power balance is found to be preserved discretely on each individual subdomain. The subdomains have been coupled using the dual field method such that the energy is also conserved on the entire domain. Unlike existing energy preserving schemes [17] where the energy is reconstructed from the dual and primal formulation that do not preserve the energy, the energy is conserved in all systems for the proposed method.

Furthermore, the introduction of an internal interface boundary does not lead to an increase in the L^2-error when compared to standard finite element discretizations. So proposed method does not sacrifice in accuracy. However, when the curl free condition was analyzed for the proposed method. It was found that the resulting integral of the curl is zero on the subdomain that uses the dual formulation, but on the subdomain that employs the primal formulation the curl free condition is not valid. That means by using the proposed method, part of the domain will never meet the curl free condition.

Finally, a time staggering integration scheme was applied to the time dependent Poisson equation. It was found to still preserve the energy, and the L^2-error was comparable to the non-staggering method. The computational cost was estimated by looking at the number of degrees of freedom in the domains of the different schemes. The time staggering scheme is found to be able to significantly reduce the number of degrees of freedom to be solved for each (half) time step when compared to non-staggering schemes.

7 Recommendation

The beginnings of a dual field method with domain decomposition have been created in this work. The proposed method shows promising results, but a lot of work is still to be done before it can be applied to real world applications. To advance the usefulness of the method, some recommendations for future work are suggested in this section.

Firstly, the only domain used in this work was a simple unit square domain. While suitable for academic works such as this, simulations in industry are almost never performed on such simple domains. Although in a mathematical sense the proposed method should work on any arbitrary domain, it is important to show this with actual numerical results.

Secondly, the simulations were performed for mixed boundary problems with one Neumann and one Dirichlet boundary conditions. However, even simple problems, such as a channel flow, can have more than two boundaries. For any added boundary an additional interface must be introduced for the proposed method to function as intended. The extra interfaces will introduce more coupling terms in the weak formulation which will make the problem harder to solve.

Thirdly, the interface used to divide the domain in the simulation was a straight diagonal line. For more complex domain shapes a straight line interface will not always be possible. Simulation with interfaces with an arbitrary shape should be performed. For those interfaces the mesh will have to be adapted to take the shape of the internal face into account. This can have an effect on the accuracy, convergence rates and the local mesh refinement.

Fourthly, a second order implicit midpoint scheme was used for the time integration part of the problem. As already mentioned in Section 4.3, a fourth order implicit midpoint scheme has been developed which should be suitable for the method used in this work. Using a fourth order scheme instead of a second order scheme could significantly effect the accuracy of the results.

Finally, the problems solved by the proposed method are of a port-Hamiltonian nature. The port-Hamiltonian systems have certain properties that can be beneficial to the computational time of the simulation. Every matrix $A \in \mathbb{C}^{n,n}$ can be split into $A = H + S$ with $H = \frac{1}{2}(A + A^*)$ and $S = \frac{1}{2}(A - A^*)$, where A^* indicates the Hermitian or conjugate transpose of A. For Hamiltonian systems this splitting is not merely a mathematical operation, but it occurs naturally and has a physical meaning [32]. The splitting of the matrix can be used in numerical schemes to improve performance by preconditioning the system. The Hermitian and skew-Hermitian splitting (HSS) can be applied to linear problems to reduce the computational times. Güdücü et al. [32] compared Widlund's method, Rapoport's method and L-GMRES to the standard GMRES method for linear problems with the form of $Ax = b$. They found a significant reduction in computational time for Widlund's, Rapoport's and the L-GMRES methods when compared with the GMRES method. The computational time of the problems presented in this work could be similarly improved by implementing these preconditioning methods.

References

[1] Sergey Charnyi, Timo Heister, Maxim A Olshanskii, and Leo G Rebholz, On conservation laws of navier–stokes galerkin discretizations, *Journal of Computational Physics*, (2017), 337:289–308.

[2] Donghyun You, Frank Ham, and Parviz Moin, Discrete conservation principles in large-eddy simulation with application to separation control over an airfoil, *Physics of Fluids*, (2008), 20(10):101515.

[3] Davide Modesti and Sergio Pirozzoli, A low-dissipative solver for turbulent compressible flows on unstructured meshes, with openfoam implementation, *Computers & Fluids*, (2017), 152:14–23.

[4] Norman A Phillips, An example of non-linear computational instability, *The atmosphere and the sea in motion*, (1959), 501:504.

[5] Gennaro Coppola, Francesco Capuano, and Luigi de Luca, Discrete energy-conservation properties in the numerical simulation of the navier–stokes equations, *Applied Mechanics Reviews*, (2019), 71(1).

[6] Akio Arakawa, Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. part i, *Journal of computational physics*, (1997), 135(2):103–114.

[7] Wenjun Cai, Chaolong Jiang, Yushun Wang, and Yongzhong Song, Structure-preserving algorithms for the two-dimensional sine-gordon equation with neumann boundary conditions, *Journal of Computational Physics*, (2019), 395:166–185.

[8] Jiaxiang Cai and Yushun Wang, Local structure-preserving algorithms for the "good" boussinesq equation, *Journal of Computational Physics*, (2013), 239:72–89.

[9] Hassler Whitney, *Princeton mathematical series: Geometric integration theory*, Princeton University Press, (1957).

[10] Jin-Fa Lee and Zachary Sacks, Whitney elements time domain (wetd) methods, *IEEE Transactions on Magnetics*, (1995), 31(3):1325–1329.

[11] Alain Bossavit, Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism, *IEE Proceedings A (Physical Science, Measurement and Instrumentation, Management and Education, Reviews)*, (1988), 135(8):493–500.

[12] Douglas N Arnold, Differential complexes and numerical stability, *arXiv preprint math/0212391*, (2002).

[13] Douglas N Arnold, Richard S Falk, and Ragnar Winther, Finite element exterior calculus, homological techniques, and applications, *Acta numerica*, (2006), 15:1–155.

[14] Douglas Arnold, Richard Falk, and Ragnar Winther, Finite element exterior calculus: from hodge theory to numerical stability, *Bulletin of the American mathematical society*, (2010), 47(2):281–354.

[15] Douglas N Arnold and Anders Logg, Periodic table of the finite elements, *Siam News*, (2014), 47(9):212.

[16] Yi Zhang, Artur Palha, Marc Gerritsma, and Leo G Rebholz, A mass-, kinetic energy-and helicity-conserving mimetic dual-field discretization for three-dimensional incompressible navier-stokes equations, part i: Periodic domains, *Journal of Computational Physics*, (2022), 451:110868.

[17] Andrea Brugnoli, Ramy Rashad, and Stefano Stramigioli, Dual field structure-preserving discretization of port-hamiltonian systems using finite element exterior calculus, *arXiv preprint arXiv:2202.04390*, (2022).

[18] Ralph Abraham, Jerrold E Marsden, and Tudor Ratiu, *Manifolds, tensor analysis, and applications*, volume 75, Springer Science & Business Media, (2012).

[19] Joseph J Rotman, *An introduction to algebraic topology*, volume 119, Springer Science & Business Media, (2013).

[20] Günter Schwarz, *Hodge Decomposition-A method for solving boundary value problems*, Springer, (2006).

[21] Marián Fecko, Vector calculus in two-dimensional space, *arXiv preprint arXiv:2201.05470*, (2022).

[22] Robert C Kirby, Algorithm 839: Fiat, a new paradigm for computing finite element basis functions, *ACM Transactions on Mathematical Software (TOMS)*, (2004), 30(4):502–516.

[23] Florian Rathgeber, David A. Ham, Lawrence Mitchell, Michael Lange, Fabio Luporini, Andrew T. T. McRae, Gheorghe-Teodor Bercea, Graham R. Markall, and Paul H. J. Kelly, Firedrake: automating the finite element method by composing abstractions, *ACM Trans. Math. Softw.*, (2016), 43(3):24:1–24:27.

[24] James Ahrens, Berk Geveci, and Charles Law, Paraview: An end-user tool for large data visualization, *The visualization handbook*, (2005), 717(8).

[25] Yongke Wu and Yanhong Bai, Error analysis of energy-preserving mixed finite element methods for the hodge wave equation, *SIAM Journal on Numerical Analysis*, (2021), 59(3):1433–1454.

[26] Denis Donnelly and Edwin Rogers, Symplectic integrators: An introduction, *American Journal of Physics*, (2005), 73(10):938–945.

[27] Said Aoues, Damien Eberard, and Wilfrid Marquis-Favre, Canonical interconnection of discrete linear port-hamiltonian systems, In: *Proc. , 52nd IEEE Conference on Decision and Control*. IEEE, (2013), 3166–3171.

[28] Jesus M Sanz-Serna, Symplectic integrators for hamiltonian problems: an overview, *Acta numerica*, (1992), 1:243–286.

[29] JM Sanz-Serna and L Abia, Order conditions for canonical runge–kutta schemes, *SIAM Journal on Numerical Analysis*, (1991), 28(4):1081–1096.

[30] J De Frutos and JM Sanz-Serna, An easily implementable fourth-order method for the time integration of wave problems, *Journal of Computational Physics*, (1992), 103(1):160–168.

[31] Moustafa M Salama, M Higazy, and Saleh Omran, Algebraic topological approach for grid generation of certain manifolds, *International Journal of Applied Engineering Research*, (2018), 13(8):6225–6233.

[32] Candan Güdücü, Jörg Liesen, Volker Mehrmann, and Daniel B Szyld, On non-hermitian positive (semi) definite linear algebraic systems arising from dissipative hamiltonian daes, *SIAM Journal on Scientific Computing*, (2022), 44(4):A2871–A2894.

[33] Tomasz Salwa and Onno Bokhove, Linear mixed fluid-structure interaction system, `https://www.firedrakeproject.org/demos/linear_fluid_structure_interaction.py.html`, [Online; accessed 25-April-2023].

A Appendix

A.1 Firedrake simulation

Firedrake is an open source program that calculates solutions to weak formulations for a given mesh. Firedrake was used with Pycharm and a custom Docker image. The simulation starts by defining the required finite elements for the simulations. One easily missed thing here is that the input on the natural boundary is defined on another finite element type. For example, $\hat{u}^1$ uses a Raviart-Thomas element while ϕ^0 and σ^1 use Continuous Galerkin and Nédélec first kind respectively. The Raviart-Thomas element will still have to be properly defined in the code to correctly apply the boundary input. The N1 and CG spaces are combined into mixed element space, the RT space is not in this mixed space as ϕ^0 and σ^1 do not use RT.

```
P_0 = FiniteElement("CG", triangle, deg)
P_1 = FiniteElement("N1curl", triangle, deg)

P_1_out = FiniteElement("RT", triangle, deg)

V_1 = FunctionSpace(mesh, P_1)

V_0 = FunctionSpace(mesh, P_0)
V_1_out = FunctionSpace(mesh, P_1_out)

V_10 = V_1 * V_0
```

The next big issue is the analytical solution. While the Firedrake program suggest that the elements are based on differential forms in exterior calculus. Underneath the code the elements are actually described in vector calculus. This becomes a problem for the 1-forms in two-dimensions, from Figure 4 we see that to go from a vector field requires a sharp operator $\sharp : \Omega^1 \to \mathfrak{X}$. The Firedrake documentation is suggesting that it is performing this transformation while it is in fact doing no such thing. The user will have to translate the 1-forms to a vector field on paper themselves before putting it in the code.

The nice thing about using Firedrake is that it able to transfer from weak formulations to algebraic form automatically. That means that the large matrixes, e.g. (65), do not have to be created by the user, so the weak formulation can be used directly.

```
a_form10 = inner(v_1, u_1) * dx - inner(v_1, grad(p_0)) * dx - inner(grad(v_0), u_1) * dx
L_form10 = -v_0 * dot(u_1_out, n_vec) * ds - inner(v_0, f_source) * dx
```

Those are the main things to mention when simulating standard finite element formulations in Firedrake. For the proposed method things can get a bit more complicated due to the domain decomposition. The coupling on the interface requires information of all the degrees of freedom and variables on the interface. To obtain this information is not straightforward in Firedrake, because the indexing of the different degrees of freedom is not consistent. That is, the degrees of freedom with index 1 can be nowhere near the dof with index 2. A method had to be found to ensure the calculation were performed in the correct place. The best method was found to be the use of an exclusion boundary condition, which works as follows. Firstly, all the finite element spaces were defined on the entire domain, not just the subdomain the are used in. Then an indicator was used such that it is 1 on the desired domain and 0 everywhere else, by using the Firedrake functions *par_loop*. It should be noted that the code segments form this point onwards have been based on an advanced Firedrake tutorial [33], where the code has been edited wherever necessary.

```
def Indicator(DGspace, sub_domain):
    # DGspace : A Discontinuous Galerkin FunctionSpace of degree 0
    # sub_domain : The domain the indicator function is applied to
    I = Function(DGspace)  # Function in DG
    par_loop(("{[i] : 0 <= i < f.dofs}", "f[i, 0] = 1.0"),
             dx(sub_domain),
             {"f": (I, WRITE)},
             is_loopy_kernel=True)
    return I
```

The stepfunction can be used to create a Firedrake function that assign a value to the degrees of freedom in the finite element space.

```
def StepFunction(I,H):
    # I : An indicator function
    # H : The heaviside step Function for the desired FunctionSpace
    par_loop(("{[i] : 0 <= i < A.dofs}", "A[i, 0] = fmax(A[i, 0], B[0, 0])"),
             dx,
             {"A": (H, RW), "B": (I, READ)},
             is_loopy_kernel=True)
    return H
```

The Heaviside step function I can then be used to create an exclusion boundary condition. The Firedrake function *DirichletBC* is redefined into a new type of boundary condition. What this boundary condition does is that it tells the solver not to use any data from part of the domain, thereby excluding that part of the domain from the solver. In this way the domain decomposition method is implemented in the code.

```
class MyBC(DirichletBC):
    def __init__(self, V, value, markers):  # (sub)space, value, function on space
        # Call superclass init
        # We provide a dummy subdomain id.
        super(MyBC, self).__init__(V, value, 0)
        # Override the "nodes" property which says where the boundary
        # condition is to be applied.
        self.nodes = np.unique(np.where(markers.dat.data_ro_with_halos == 0)[0])
```

There is one more problem on the interface, there are two versions of each variable on the interface, one from Γ_{int10} and on from Γ_{int21}. When coupling on the interface the solver has to use the correct one for the simulation to work. In fact, if it is not specified the program crashes. While this needs to be done for any variable used on the interface the easiest to explain is the normal vector. Firedrake defined the normal as the outward unit normal, but the interface is an internal interface so which direction is the outward normal? Firedrake uses signs "+" and "-" to indicate the different sides, but it is again not consistent with its use, i.e. not all "+" signs have to be on the left side of the interface. Luckily, we already have defined an indicator function when creating the exclusion boundary condition. This indicator is useful because it is 1 on one side of the interface and 0 on the other side. The direction of the normal on the interface can then be enforced by

```
def interface_val(I, variable):
    # I : Indicator Function
    # var : Variable
    return I("+") * variable("+") + I("-") * variable("-")
```