Convergence Rate of the Dirichlet-Neumann Algorithm for Coupled Poisson Equations

Morgan Görtz

Master's thesis
2019:E21

LUND UNIVERSITY

Centre for Mathematical Sciences
Numerical Analysis

CENTRUM SCIENTIARUM MATHEMATICARUM

Abstract

This thesis presents and tests the convergence rate of the Dirichlet-Neumann algorithm for two Poisson equations coupled by transmission boundary conditions. Three second order discretisation methods are used when analyzing the convergence: standard equidistant finite difference, standard adaptive linear finite element, and standard adaptive finite volume discretisation of Poisson's equation. The convergence rate of the Dirichlet-Neumann algorithm, when using each of the discretisations for both sub problems, is presented and proved. Using elements of the proofs for the intermediate results leads to a theorem when combining the discretisations. The theorem states that the Dirichlet-Neumann algorithm's convergence rate is entirely independent of the grid used for any combination of the discretisations analyzed. Inspired by these results a general theorem of the convergence rate is presented. Using semi-discrete analysis it is possible to generalize the results to a large subset of discretisations in the asymptotic case. It is possible to remove the asymptotic argument if the discretisations approximate the homogenous solution exactly. All theoretical results were numerically confirmed. The numerical results aligned with the theoretical conclusions.

Acknowledgements

I would like to thank the staff at the institution of numerical analysis at Lund University for their enthusiasm. Thanks to them, I was hooked on numerical analysis from my very first course. I still remember when Claus Führer gave us the first methods to play with and tried to teach us some of the theory along the way. After that, I took all the numeric courses I could. I struggled with the theory of numerical approximations and drew nice pictures in numerical methods for CAGD with Carmen Arévalo. I became frustrated with the discretisations and stability calculus in numerical methods for differential equations with Gustaf Söderlind. I even took a challenging break from my engineering studies with a seminar course with Philipp Birken. This institution has formed a fascination and interest in a field I had no idea existed and for that I am grateful.

I would like to thank Philipp Birken, my supervisor, for his thorough and constructive criticism, directness, and openness which have been invaluable assets while writing this thesis. Philipp has always made me want to do my very best, which I appreciate greatly.

Lastly, I would like to thank Claus Führer. Claus is the person I could always talk to about my studies. He sparked my interest in numerics and he was the one who suggested combining a numerical degree with a computer science degree. Without him I would not have this passion for numerical analysis and I would not be pursuing a double master.

Contents

Chapter 1

Introduction

Differential equations coupled by transmission conditions are differential equations coupled by boundary conditions on common interfaces. These problems arise in aerodynamics [1], bio-mechanics [2] and rocket simulations [3] and are usually impossible to solve analytically. Finding properties of numerical methods that solve these transmission problems is therefore of interest.

Domain decomposition can be used to work with transmission problems. Instead of dealing with one large problem, we can analyze a set of sub problems. Using the Dirichlet-Neumann algorithm allows us to solve the sub problems several times in order to find a solution to the larger problem. Codes and knowledge could exist for the sub problems when none exist for the larger problem. In such cases the algorithm allows us to reuse code and therefore requires fewer resources.

The Dirichlet-Neumann algorithm allows us to mix discretisation techniques. We can use a finite volume discretisation on one domain and a finite element discretisation on the other domain because of the modularity of the algorithm. There are problems where mixed discretisations are of interest. There is research where a finite volume method is used on a fluid domain and a finite element method is used on a structure domain [4]. This thesis focuses on the convergence rate when mixing three different discretisations. The first discretisation is equidistant finite difference, which was chosen because it is familiar and the result of this thesis extends to it. The other discretisations are the second order standard adaptive finite element and finite volume methods. For these discretisations the convergence rate can be found and is found exactly.

This approach of dividing and mixing is only relevant if the algorithm converges and if it converges at a reasonable pace. The idea is interesting, but is it useful? If the algorithm does not converge a solution will not be found. If the convergence is slow, the execution times may be unreasonably long. To answer these questions, we need to analyze the algorithms convergence rate. In this thesis we focus on a specific set of transmission problems: two one-dimensional Poisson equations coupled at one point. Analyzing these convergence rates gives us information about effectiveness of the algorithm. The goal of this thesis is to give some insight into finding the convergence rates of higher dimensional time dependent problems. By analyzing an easier set of equations, one might find hints on how to handle the harder ones.

Chapter 2

Theory

2.1 Schur Complement

The Schur complement introduced in this section is inspired by Fuzhen [9, p.17-18].

Remark 2.1.1. *For a linear system of the form:*

$$\begin{bmatrix} A_{11} & 0 & A_{1\Gamma} \\ 0 & A_{22} & A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_\Gamma \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_\Gamma \end{bmatrix}$$

we can express u_Γ by the following equation:

$$\begin{aligned} Su_\Gamma &= f_\Gamma - A_{\Gamma 1} A_{11}^{-1} f_1 - A_{\Gamma 2} A_{22}^{-1} f_2 \\ S &= A_{\Gamma\Gamma} - A_{\Gamma 1} A_{11}^{-1} A_{1\Gamma} - A_{\Gamma 2} A_{22}^{-1} A_{2\Gamma}, \end{aligned} \tag{2.1}$$

where S is the Schur compliment.

Proof. We use block Gaussian elimination by multiplying the first system with $A_{11}^{-1} A_{\Gamma 1}$ and subtracting it from the last. We do the same thing with the second system with $A_{22}^{-1} A_{\Gamma 2}$. This becomes:

$$\begin{bmatrix} A_{11} & 0 & A_{1\Gamma} \\ 0 & A_{22} & A_{2\Gamma} \\ 0 & 0 & A_{\Gamma\Gamma} - A_{\Gamma 1} A_{11}^{-1} A_{1\Gamma} - A_{\Gamma 2} A_{22}^{-1} A_{2\Gamma} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_\Gamma \end{bmatrix} =$$

$$= \begin{bmatrix} A_{11} & 0 & A_{1\Gamma} \\ 0 & A_{22} & A_{2\Gamma} \\ 0 & 0 & S \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_\Gamma \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_\Gamma - A_{\Gamma 1} A_{11}^{-1} f_1 - A_{\Gamma 2} A_{22}^{-1} f_2 \end{bmatrix}.$$

Here we observe that the last equation in the linear system is (2.1). □

2.2 Convergence Rate

Definition 2.2.1. *Given a converging sequence $l_k,\ k \in \mathbb{Z}^+$, where $l_n \to l,\ n \to \infty$. Then the sequence l_k is said to converge linearly to $\mu \in (0,1)$, if:*

$$\frac{|e_{k+1}|}{|e_k|} \to \mu,\ k \to \infty. \tag{2.2}$$

where $e_k = l_k - l$. *We call* μ *the convergence rate.*

Definition 2.2.2. *A matrix* $\Sigma \in \mathbb{C}$ *is called an iteration matrix for the sequence* $l_k \in \mathbb{C}$ *if:*

$$l_{k+1} = \Sigma l_k + \alpha, \tag{2.3}$$

where α *is a constant independent of* k.

Theorem 2.2.1. *For the one dimensional case the convergence rate* μ *of a complex valued sequence* l_k *converging to* l *is the absolute value of the iteration matrix. Moreover:*

$$\frac{|e_{k+1}|}{|e_k|} = |\Sigma| = \mu,$$

where $e_k = l_k - l$.

Proof. The first part of this proof is inspired by Fox [10, p.190-191].

$$l_{k+1} = \Sigma l_k + \alpha \Leftrightarrow l_{k+1} - \Sigma l_k = \alpha$$

$$\Rightarrow \lim_{k\to\infty} l_{k+1} - \Sigma l_k = l - \Sigma l = \alpha \Leftrightarrow \alpha = l - \Sigma l.$$

Inserting this into (2.3) we get:

$$l_{k+1} = \Sigma l_k + l - \Sigma l$$

$$\Leftrightarrow l_{k+1} = \Sigma(l_k - l) + l \Leftrightarrow l_{k+1} = \Sigma e_k + l \Leftrightarrow l_{k+1} - l = \Sigma e_k$$

$$\Leftrightarrow e_{k+1} = \Sigma e_k \Rightarrow e_{k+1} = \Sigma^{k+1} e_0$$

$$\Rightarrow \frac{|e_{k+1}|}{|e_k|} = \frac{|\Sigma^{k+1} e_0|}{|\Sigma^k e_0|} = \frac{(|\Sigma|)^{k+1}|e_0|}{(|\Sigma|)^k|e_0|} = |\Sigma|$$

$$\Leftrightarrow |e_{k+1}| = |\Sigma||e_k|,$$

which is what we wanted to prove. □

2.3 Coupled Poisson Equations

Let's start by introducing Poisson's equation with two sets of boundary conditions.

$$\begin{aligned} u'' = f(x),\ x \in [a,b] \\ u(a) \text{ and } u(b) \text{ given} \end{aligned} \tag{2.4}$$

and

$$\begin{aligned} u'' = f(x),\ x \in [a,b] \\ u'(a) \text{ and } u(b) \text{ given,} \end{aligned} \tag{2.5}$$

for some $x_\Gamma \in [a,b]$. The first differential equation has two Dirichlet boundary conditions and the second has one Dirichlet and one Neumann boundary condition. The coupled problem we wish to analyze in this thesis has the following form:

$$\begin{aligned} \lambda_1 v''(x) = f_1(x),\ x \in [a, x_\Gamma] \\ \lambda_2 w''(x) = f_2(x),\ x \in [x_\Gamma, b] \\ \lambda_1 v'(x_\Gamma) = \lambda_2 w'(x_\Gamma),\ v(x_\Gamma) = w(x_\Gamma) \\ v(a) \text{ and } w(b) \text{ given.} \end{aligned} \tag{2.6}$$

This equation is two coupled Poisson equations, coupled by two transmission conditions. The transmission interface is x_Γ and the first condition forces continuity between the two functions and the second forces a relation on the derivative over the interface. Examples and analytic solutions to all three of these problems can be found in Appendix A1. In Chapter 2.4 we will show a way to decouple (2.6). We will create a sequence of functions v_i and w_i, that are defined by solving equations (2.4) and (2.5) alternatingly. If these sequences converge to limits v and w they will be a solution to (2.6).

2.4 The Dirichlet-Neumann Algorithm in 1D

The Dirichlet-Neumann method and algorithm is presented in this section. The method and algorithm we use is presented in Toselli [11, p.8]. The section on convergence rate is presented similarly in Monge [5, p.20-26]. The main differences are that Monge presents the discretisation for the multidimensional problem, focuses on the finite element discretisation as discretisation, and is discretizing a slightly different differential equation.

2.4.1 Method and Algorithm

To find an approximation of u and v in the transmission problem (2.6) we will use the Dirichlet-Neumann method and algorithm. The Dirichlet-Neumann method starts by rearranging the initial problem as two separate problems coupled by a common interface, x_Γ. The two problems are: a Dirichlet boundary problem and a Neumann boundary problem.

$$\begin{aligned} \lambda_1 v''(x) &= f_1(x),\ x \in [a, x_\Gamma] \\ v(a) \text{ given and } v(x_\Gamma) &= w(x_\Gamma) \\ \\ \lambda_2 w''(x) &= f_2(x),\ x \in [x_\Gamma, b] \\ w'(x_\Gamma) &= \frac{\lambda_1}{\lambda_2} v'(x_\Gamma) \text{ and } w(b) \text{ given.} \end{aligned} \tag{2.7}$$

The algorithm start with an initial guess of $w(x_\Gamma)$. With this approximation we solve the Dirichlet part of (2.7). This will give us a function we will call v^1. We then solve the Dirichlet part of (2.7) with the Neumann condition $\frac{\lambda_1}{\lambda_2}(v^1)'(x_\Gamma)$. The solution to this problem we call w^1. Using w^1, we solve the Dirichlet part of (2.7) with Dirichlet condition $w^1(x_\Gamma)$. This new solution we call v^2. This process continues with v^k and w^k being the functions for the kth iteration. If w^k and v^k converge to some functions w and v, then by construction they must be a solution to (2.7) and in extension (2.6). The recursion is:

$$\begin{aligned}
&w^0(x_\Gamma) \text{ given}\\
&\lambda_1(v^{k+1})''(x) = f_1(x),\ x \in [a, x_\Gamma]\\
v^{k+1}(a) = v(a) \text{ and } &v^{k+1}(x_\Gamma) = w^k(x_\Gamma)\\
\\
&\lambda_2(w^{k+1})''(x) = f_2(x),\ x \in [x_\Gamma, b]\\
&(w^{k+1})'(x_\Gamma) = \frac{\lambda_1}{\lambda_2}(v^k)'(x_\Gamma) \text{ and } w^{k+1}(b) = w(b).
\end{aligned} \tag{2.8}$$

This recursive algorithm is called a continuous Dirichlet-Neumann algorithm. Now we discretize this algorithm. Each iteration requires us to solve two Poisson equations, one with Dirichlet boundary conditions and one with a Dirichlet and a Neumann boundary condition. So we discretize each of these individual problems. The discretisations of the Dirichlet problem we analyze can be written as the following linear system:

$$\lambda_1 A^{(1)}\bar{v}^{k+1} = b^{(1)}v(a) + \bar{f}^{(1)} - \lambda_1 A_\Gamma^{(1)} w^k(x_\Gamma), \tag{2.9}$$

where $\bar{v}^{k+1}$ is some discrete representation of v^{k+1}. The discretisation of the Neumann problem we analyze can be written as:

$$\lambda_2 \begin{bmatrix} A^{(2)} & A_\Gamma^{(2)} \\ d^{(2)} & d_\Gamma^{(2)} \end{bmatrix} \begin{bmatrix} \bar{w}^{k+1} \\ w^{k+1}(x_\Gamma) \end{bmatrix} = \begin{bmatrix} b^{(2)}u(b) + \bar{f}^{(2)} \\ f_\Gamma^{(2)} - f_\Gamma^{(1)} + \lambda_1 d^{(1)}\bar{v}^{k+1} + \lambda_1 d_\Gamma^{(1)} w^k(x_\Gamma), \end{bmatrix} \tag{2.10}$$

where $\bar{w}^{k+1}$ is some discrete representation of w^{k+1}, $d^{(1)}w^k(x_\Gamma) + d_\Gamma^{(1)}\bar{v}^{k+1}$ is an approximation of $(v^{k+1})'(x_\Gamma)$ and $d^{(2)}w^{k+1}(x_\Gamma) + d_\Gamma^{(2)}\bar{w}^{k+1}$ is an approximation of $(w^{k+1})'(x_\Gamma)$. Combining these discretisation into one linear system results in:

$$\begin{bmatrix} \lambda_1 A^{(1)} & 0 & 0 \\ 0 & \lambda_2 A^{(2)} & \lambda_2 A_\Gamma^{(2)} \\ -\lambda_1 d^{(1)} & \lambda_2 d^{(2)} & \lambda_2 d_\Gamma^{(2)} \end{bmatrix} \begin{bmatrix} \bar{v}^{k+1} \\ \bar{w}^{k+1} \\ w^{k+1}(x_\Gamma) \end{bmatrix} = \begin{bmatrix} b^{(1)}v(a) - \lambda_1 A_\Gamma^{(1)} w^k(x_\Gamma) + \bar{f}^{(1)} \\ b^{(2)}w(b) + \bar{f}^{(2)} \\ f_\Gamma^{(2)} - f_\Gamma^{(1)} + \lambda_1 d_\Gamma^{(1)} w^k(x_\Gamma) \end{bmatrix}. \tag{2.11}$$

We call this the discrete Dirichlet-Neumann Algorithm. This algorithm allows us to solve the transmission problem (2.6) by using any discretisation methods to approximate solutions to Poisson equations without modifications. This is incredibly useful. That is, if it converges and if its convergence is reasonable. In the next section we will discuss this in detail.

2.4.2 Convergence Rate

We will analyze the convergence rate of $w^k(x_\Gamma)$. If this value converges, so will the algorithm. To find the convergence rate of $w^k(x_\Gamma)$ of the Dirichlet-Neumann algorithm we first find the iteration matrix Σ.

$$w^{k+1}(x_\Gamma) = \Sigma w^k(x_\Gamma) + \alpha,\ \alpha \text{ independent on } k. \tag{2.12}$$

To get an expression for $w^{k+1}(x_\Gamma)$, we start by taking the Schur compliment, Section 2.1, of (2.11).

$A_1 = \lambda_1 A^{(1)},\ A_2 = \lambda_2 A^{(2)},\ A_{1\Gamma} = 0,\ A_{2\Gamma} = \lambda_2 A_\Gamma^{(2)},\ A_{\Gamma 1} = -\lambda_1 d^{(1)},\ A_{\Gamma 2} = \lambda_2 d^{(2)}$

$$A_{\Gamma\Gamma} = \lambda_2 d_\Gamma^{(2)},\ f_1 = b^{(1)}v(a) + \bar{f}^{(1)} - \lambda_1 A_\Gamma^{(1)} w^k(x_\Gamma),\ f_2 = b^{(2)}w(b) + \bar{f}^{(2)},$$

$$f_\Gamma = f_\Gamma^{(2)} - f_\Gamma^{(1)} + \lambda_1 d_\Gamma^{(1)} w^k(x_\Gamma).$$

With this we can get an equation for $w^{k+1}(x_\Gamma)$:

$$Sw^{k+1}(x_\Gamma) = f_\Gamma^{(2)} - f_\Gamma^{(1)} + \lambda_1 d_\Gamma^{(1)} w^k(x_\Gamma) - (-\lambda_1 d^{(1)})(\lambda_1 A^{(1)})^{-1}(b^{(1)}v(a) + \bar{f}^{(1)} - \lambda_1 A_\Gamma^{(1)} w^k(x_\Gamma))$$

$$-(\lambda_2 d^{(2)})(\lambda_2 A^{(2)})^{-1}(b^{(2)}w(b) + \bar{f}^{(2)})$$

$$S = \lambda_2 d_\Gamma^{(2)} - (\lambda_2 d^{(2)})(\lambda_2 A^{(2)})^{-1}(\lambda_2 A_\Gamma^{(2)}).$$

Next we put everything that isn't dependent on k into a new constant α_1.

$$Sw^{k+1}(x_\Gamma) = \lambda_1(d_\Gamma^{(1)} - (d^{(1)})(A^{(1)})^{-1}(A_\Gamma^{(1)}))w^k(x_\Gamma) + \alpha_1$$

Because we are in the one dimensional case, we know that $S \in \mathbb{R}$.

$$\Rightarrow w^{k+1}(x_\Gamma) = \frac{\lambda_1}{\lambda_2} \frac{d_\Gamma^{(1)} - (d^{(1)})(A^{(1)})^{-1}(A_\Gamma^{(1)})}{d_\Gamma^{(2)} - (d^{(2)})(A^{(2)})^{-1}(A_\Gamma^{(2)})} w^k(x_\Gamma) + \alpha,$$

where α is another variable that is independent of k. From this we can get Σ:

$$\Sigma = \frac{\lambda_1}{\lambda_2} \frac{d_\Gamma^{(1)} - d^{(1)}(A^{(1)})^{-1}A_\Gamma^{(1)}}{d_\Gamma^{(2)} - d^{(2)}(A^{(2)})^{-1}A_\Gamma^{(2)}}.$$

We know from Theorem 2.2.1 that the convergence rate is the absolute value of the iteration matrix Σ.

$$\mu = |\Sigma| = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{d_\Gamma^{(1)} - d^{(1)}(A^{(1)})^{-1}A_\Gamma^{(1)}}{d_\Gamma^{(2)} - d^{(2)}(A^{(2)})^{-1}A_\Gamma^{(2)}}\right|. \tag{2.13}$$

2.5 Finite Difference Method for Poisson's Equation

A second order finite difference discretisation of Poisson's equation is presented in this section. The discretisations will be used to approximate the one dimensional Poisson's equation. This section is heavily inspired by Ortega [7, p.74-82]. In Ortega [7, p.74-82], the approximations used and the method to construct the finite difference discretisation in this section are presented.

Before we introduce our approximations and discretisation we need to define a grid. We will use equidistant grids where Δx is the distance between the grid points, x_i, on the interval $[a, b]$. With the grid defined, we start by providing the two second order finite difference approximations we use for our numerical method.

$$u''(x) = \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{\Delta x^2} + \mathcal{O}(\Delta x^2). \tag{2.14}$$

$$u'(x) = \frac{-3u(x) + 4u(x+\Delta x) - u(x+2\Delta x)}{2\Delta x} + \mathcal{O}(\Delta x^2). \tag{2.15}$$

2.5.1 Discretizing Poisson's Equation

In this section we wish to to find an approximation of u in the differential equations (2.4) and (2.5). We start by approximating (2.4):

$$u''(x) = f(x),\ x \in [a,b],\ u(a) \text{ and } u(b) \text{ given.}$$

This method will produce approximations $u(x_i) \approx u_i$ for each grid point, so we focus on:

$$u''(x_i) = f(x_i),\ \forall i = 1 \dots n.$$

Next we use the discrete approximation of the second derivative (2.14) to get the following approximated relation,

$$\left(\frac{-u_{i-1} + 2u_i - u_{i+1}}{\Delta x}\right) = -\Delta x f(x_i),\ \forall i = 1 \dots n,$$

where $u_0 = u(a)$ and $u_{n+1} = u(b)$. The discrete approximation, $\bar{u}$, of u is then given by solving the following linear system:

$$\begin{aligned} A\bar{u} &= \bar{f} + \frac{u(a)}{\Delta x}e_1 + \frac{u(b)}{\Delta x}e_n \\ \bar{f} \in \mathbb{R}^n :\ [\bar{f}]_i &= -\Delta x f(x_i) \\ A \in \mathbb{R}^{n\times n} :\ [A]_{i,j} &= \begin{cases} \frac{2}{\Delta x},\ i = j \\ -\frac{1}{\Delta x},\ i = j+1 \\ -\frac{1}{\Delta x},\ j = i+1 \\ 0 \text{ else} \end{cases}. \end{aligned} \tag{2.16}$$

To extend this approximation to Neumann boundary conditions, (2.5), we treat $u(a) \approx u_a$ as an unknown.

$$\begin{bmatrix} A & -\frac{1}{\Delta x} \end{bmatrix} \begin{bmatrix} \bar{u} \\ u_a \end{bmatrix} = \bar{f} + \frac{u(b)}{\Delta x}e_n. \tag{2.17}$$

We need one more equation to make this system square. The equation will be chosen such that the Neumann condition holds. By using the one-sided approximation of $u'(a)$, (2.15), we can get such an equation.

$$u'(a) \approx \frac{-3u(x_0) + 4u(x_1) - u(x_2)}{2\Delta x} \approx \frac{-3u_a + 4u_1 - u_2}{2\Delta x}.$$

Adding this equation to (2.17) gives:

$$\begin{bmatrix} A & -\frac{1}{\Delta x} \\ \frac{4}{2\Delta x}e_1^T - \frac{e_2^T}{2\Delta x} & -\frac{3}{2\Delta x} \end{bmatrix} \begin{bmatrix} \bar{u} \\ u_a \end{bmatrix} = \begin{bmatrix} \bar{f} + \frac{u(b)}{\Delta x}e_n \\ u'(a) \end{bmatrix} \tag{2.18}$$

This system is square and solving it gives the discrete approximation $\bar{u}$ and u_a of $u(x)$ and $u(a)$.

2.5.2 Numerical confirmation

The approximations' convergence and order were analyzed. To analyze the methods convergence we used the test problems (5.2) and (5.1). These test problems were approximated by the finite difference approximations (2.16) and (2.18) and made continuous by linear interpolation. These continuous approximations we call u^h. These approximations' error with respect the the exact solution was then evaluated in the L_2-norm, $\|u - u^h\|_{L_2}$ for different Δx. These errors were then plotted in a logarithmic graph. This graph is presented in Figure 2.1. In Figure 2.1 the error seems to converge as Δx tends to zero. It seems to be a second order convergence. We see this because when Δx is 10^{-3} the error is 10^{-7} and when Δx is 10^{-4} the error is almost 10^{-9}. The method we constructed was supposed to be a second order method and the numerical results shows a second order convergence.

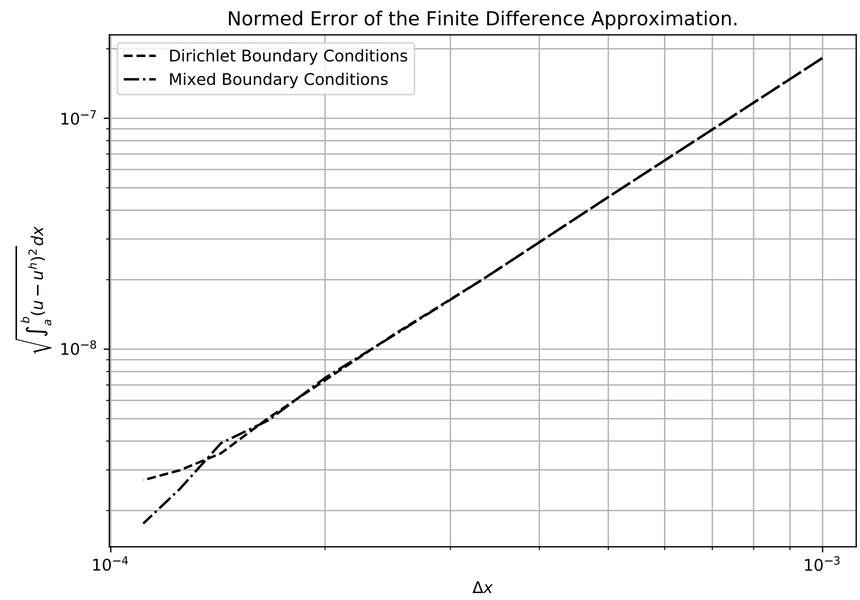

Figure 2.1: The error of the approximations with respect to Δx. The discrete approximations were found using the strategies presented in Section 2.5.1. Then they were made continuous by linear interpolation. The continuous approximation, u^h, was then compared to the real solution, u, in the L_2-norm. Both methods seem to be second order in this norm.

2.6 Finite Element Method For Poisson's Equation

In this section a finite element discretisation for the Poisson's equation in one dimension is presented. First the grid and the basis functions used for the method are presented. Then we use Galerkin's method to get a general discretisation template for Poisson's equation. With a template we create the discretisations for the Poisson's equation with both Dirichlet and Neumann boundary conditions. Lastly we confirm that the method converges and that the method is a second order method numerically. We omit the proof that discusses the order of the method; a detailed explanation can be found in Hughes [6, p.27-31].

2.6.1 Basis functions

The goal is to find a discrete approximation, u^h, of u for Poisson's equation. The finite element method approximates u using a linear combination of basis functions ϕ.

$$u \approx u^h = \sum_{k=1}^{n} u_k \phi_k.$$

We partition these functions into two sets: ϕ^I and ϕ^Γ. A function in ϕ^I is 0 on the boundary points $\{a, b\}$ and a function in ϕ^Γ is not. With this new partition we reformulate our approximation as:

$$u \approx u^h = \sum_{k=1}^{n^I} u_k^I \phi_k^I + \sum_{k=1}^{n^\Gamma} u_k^\Gamma \phi_k^\Gamma,$$

where u_k^I and u_k^Γ are unknowns we wish to find. We are going to focus on approximations were only two boundary functions are used, so we re-write u^h as:

$$u^h = u_1^\Gamma \phi_1^\Gamma + \sum_{k=1}^{n} u_k^I \phi_k^I + u_2^\Gamma \phi_2^\Gamma. \tag{2.19}$$

We define the basis functions with respect to a grid. The grid points we call x_i, where

$$a \leq x_1^I \leq x_2^I \leq \cdots \leq x_n^I \leq b.$$

With these grid points, we define Δx_i as the distance between them.

$$\Delta x_1 = x_1^I - a,\ \Delta x_i = x_i^I - x_{i-1}^I,\ \Delta x_{n+1} = b - x_n,\ i = 2, \ldots, n.$$

An illustration of the different variables is presented in Figure 2.2.

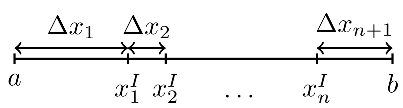

Figure 2.2: The grid used for the finite element method in one dimension.

The basis functions we use are the standard linear finite element basis functions presented in Hughes [6, p.20]:

$$\phi_i^I(x) = \begin{cases} \frac{x - x_{i-1}^I}{\Delta x_i},\ x \in (x_{i-1}^I, x_i^I] \\ \frac{x_{i+1}^I - x}{\Delta x_{i+1}},\ x \in (x_i^I, x_{i+1}^I) \\ 0,\ \text{else} \end{cases}, \tag{2.20}$$

using the convention $x_0^I = a$ and $x_{n+1}^I = b$. ϕ_1^Γ and ϕ_2^Γ are defined as cut off saw-tooth functions.

$$\phi_1^\Gamma(x) = \begin{cases} \frac{x-x_1^I}{\Delta x_1}, \ x \in [a, x_1^I) \\ 0, \ \text{else} \end{cases} \quad \phi_2^\Gamma(x) = \begin{cases} \frac{x_n^I - x}{\Delta x_{n+1}}, \ x \in (x_n^I, b] \\ 0, \ \text{else} \end{cases} \quad . \tag{2.21}$$

An illustration of the basis function are presented in Figure 2.3

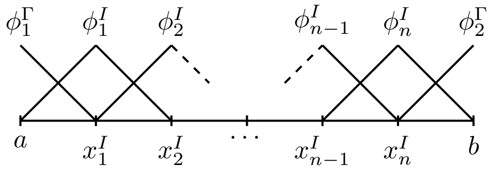

Figure 2.3: The basis functions ϕ^I and ϕ^Γ.

2.6.2 Galerkin's method

Instead of working with the Poisson equations (2.4) and (2.5) directly, we are going to work with their weak form. A motivation to why we can work with these equations instead of the original Poisson equations can be found in Hughes [6, p.4-6]. In this thesis we will use Galerkin's method for finding the basis coefficients u_k^I in (2.19). We are following the same steps as Hughes [6, p.7-11]. We simplify the process by making it less general.

The method evaluates the solution u when:

$$\int_a^b u'' \phi_i^I dx = \int_a^b f \phi_i^I dx, \ i = 1 \dots n.$$

Using integration by parts on the left hand side gives the following equivalent expression.

$$\int_a^b u'' \phi_i^I dx = \left[u' \phi_i^I \right]_a^b - \int_a^b u' (\phi_i^I)' dx = \int_a^b f \phi_i^I dx, \ i = 1 \dots n.$$

Now we use the fact that $\phi_i^I(a) = \phi_i^I(b) = 0$, which leads to the result:

$$-\int_a^b u' (\phi_i^I)' dx = \int_a^b f \phi_i^I dx, \ i = 1 \dots n.$$

To get an expression with the coefficients u_k^I, we replace u with our approximation u^h. This will produce the following linear system.

$$-\int_a^b (u^h)' (\phi_i^I)' dx = \int_a^b f \phi_i^I dx, \ i = 1 \dots n$$

$$\Leftrightarrow -\int_a^b \left(\sum_{k=1}^n u_k^I (\phi_k^I)' (\phi_i^I)' \right) dx - \int_a^b u_1^\Gamma (\phi_1^\Gamma)' (\phi_i^I)' + u_2^\Gamma (\phi_2^\Gamma)' (\phi_i^I)' dx = \int_a^b f \phi_i^I dx, \ i = 1 \dots n$$

$$\Leftrightarrow -\sum_{k=1}^{n} u_k^I \int_a^b (\phi_k^I)'(\phi_i^I)'dx - \int_a^b u_1^\Gamma(\phi_1^\Gamma)'(\phi_i^I)' + u_2^\Gamma(\phi_2^\Gamma)'(\phi_i^I)'dx = \int_a^b f\phi_i^I dx,\ i = 1\dots n$$

$$\begin{aligned} Au^I &= b - \bar{f}. \\ [A]_{i,j} &= \int_a^b (\phi_i^I)'(\phi_j^I)'dx \\ b_i &= -u_1^\Gamma \int_a^b (\phi_1^\Gamma)'(\phi_i^I)'dx - u_2^\Gamma \int_a^b (\phi_2^\Gamma)'(\phi_i^I)'dx \\ \bar{f}_i &= \int_a^b f\phi_i^I dx. \end{aligned} \tag{2.22}$$

In this thesis we only use one set of basis functions. We therefore compute the linear system with respect to the basis functions chosen. We do this by making and proving three remarks. Then we use the following three remarks to simplify the linear system (2.22).

Remark 2.6.1. *Any entry of A that is outside of the first sub-diagonal or diagonal is 0.*

Proof. The function $(\phi_i)'(\phi_j)'$ is the zero function if if $|i - j| > 1$. This is because they have no overlapping non-zeros function values. □

Remark 2.6.2.

$$[A]_{i,i} = \frac{1}{\Delta x_i} + \frac{1}{\Delta x_{i+1}},\ i = 1\dots n.$$

Proof. We will be using the convention $x_1^I = a,\ x_{n+1}^I = b$. With this we can show:

$$[A]_{i,i} = \int_{x_{i-1}^I}^{x_{i+1}^I} ((\phi_i^I)')^2\,dx = \int_{x_{i-1}^I}^{x_i^I} (\frac{1}{\Delta x_i})^2\,dx + \int_{x_i^I}^{x_{i+1}^I} \left(-\frac{1}{\Delta x_{i+1}}\right)^2 dx =$$

$$\Delta x_i \left(\frac{1}{\Delta x_i}\right)^2 + \Delta x_{i+1}\left(\frac{1}{\Delta x_{i+1}}\right)^2 = \frac{1}{\Delta x_i} + \frac{1}{\Delta x_{i+1}}.$$

□

Remark 2.6.3.

$$[A]_{i,i+1} = [A]_{i+1,i} = -\frac{1}{\Delta x_i},\ i = 1,\dots,n-1.$$

Proof. We will be using the convention $x_0^I = a,\ x_{n+1}^I = b$. With this we can show:

$$[A]_{i,i+1} = \int_{x_{i-1}^I}^{x_i^I} ((\phi_i^I)'(\phi_{i+1}^I)')\,dx = \int_{x_{i-1}^I}^{x_i^I} \left(-\frac{1}{\Delta x_i}\right)\left(\frac{1}{\Delta x_i}\right) dx$$

$$= -\Delta x_i \frac{1}{\Delta x_i^2} = -\frac{1}{\Delta x_i}.$$

Due to symmetry $[A]_{i+1,i}$ is the same. □

Remark 2.6.4.

$$b = \left(\frac{u_1^\Gamma}{\Delta x_1} e_1^T + \frac{u_2^\Gamma}{\Delta x_{n+1}} e_n^T \right). \tag{2.23}$$

Proof. $i = 1$ and $i = n$ are the only cases when it is not zero. For those two cases we have:

$$[b]_1 = -u_1^\Gamma \int_a^b (\phi_1^\Gamma)'(\phi_1^I)' dx = -u_1^\Gamma \int_a^{a+\Delta x_1} (\Delta x_1)^{-1}(-\Delta x_1)^{-1} dx = \frac{u_1^\Gamma}{\Delta x_1},$$

$$[b]_n = -u_2^\Gamma \int_a^b (\phi_2^\Gamma)'(\phi_n^I)' = -u_2^\Gamma \int_{b-\Delta x_{n+1}}^b (-\Delta x_{n+1})^{-1}(\Delta x_{n+1})^{-1} dx = \frac{u_2^\Gamma}{\Delta x_{n+1}}.$$

Writing this in vector form finishes the proof. □

$\bar{f}$ can not be simplified as it depends on what function f we use. Using Remark 2.6.2, 2.6.3 and 2.6.4 in (2.22) gives:

$$\begin{aligned} Au^I &= b - \bar{f}. \\ [A]_{i,j} &= \begin{cases} \frac{1}{\Delta x_i} + \frac{1}{\Delta x_{i+1}}, \; i = j \\ -\frac{1}{\Delta x_{i+1}}, \; i = j+1 \\ -\frac{1}{\Delta x_{i+1}}, \; j = i+1 \\ 0, \; \text{else} \end{cases} \\ b &= \lambda \left(\frac{u_a}{\Delta x_1} e_1 + \frac{u_b}{\Delta x_{n+1}} e_n \right) \\ \bar{f}_i &= \int_a^b f \phi_i^I dx. \end{aligned} \tag{2.24}$$

We let $u_1^\Gamma = u_a \approx u(a)$ and $u_2^\Gamma = u_b \approx u(b)$. This is motivated by $u^h(a) = u_1^\Gamma$ and $u^h(b) = u_2^\Gamma$.

Dirichlet Boundary conditions

If we have Dirichlet boundary conditions on a and b, we let $u(a) = u_a$ and $u(b) = u_b$ in (2.24). With these variables set, (2.24) is a linear system we can solve to get the coefficients of the approximation u^h.

A Neumann Boundary condition

Here we wish to find an approximation for (2.5). Just as the Dirichlet case we set $u_b = u(b)$ in (2.24). Unlike the Dirichlet case, $u(a)$ is not known. Therefore we treat u_a as an unknown and move it to the left hand side:

$$\begin{bmatrix} A & -\frac{1}{\Delta x_1} e_1 \end{bmatrix} \begin{bmatrix} u^I \\ u_a \end{bmatrix} = \frac{u(b)}{\Delta x_{n+1}} e_n - \bar{f}. \tag{2.25}$$

Adding an unknown to the linear system means that we are missing one equation to make it square. To get this last equation we analyze:

$$\int_a^b u'' \phi_1^\Gamma dx = \int_a^b f \phi_1^\Gamma dx.$$

By using integration by parts we get:

$$\int_a^b u''\phi_1^\Gamma dx = \left[u'\phi_1^\Gamma\right]_a^b - \int_a^b u'(\phi_1^\Gamma)'dx = \int_a^b f\phi_1^\Gamma dx$$

$$\Leftrightarrow -u'(a) - \int_a^b u'(\phi_1^\Gamma)'dx = \int_a^b f\phi_1^\Gamma dx.$$

Here we replace u with our approximation u^h.

$$-u'(a) - \int_a^b (u^h)'(\phi_1^\Gamma)'dx = \int_a^b f\phi_1^\Gamma dx$$

$$\Leftrightarrow -u_a \int_a^{a+\Delta x_1} (\phi_1^\Gamma)'(\phi_1^\Gamma)'dx - u_1^I \int_a^{a+\Delta x_1} (\phi_1^I)'(\phi_1^\Gamma)'dx = u'(a) + \int_a^b f\phi_1^\Gamma dx$$

$$\Leftrightarrow -u_a \int_a^{a+\Delta x_1} \frac{1}{\Delta x_1^2}dx - u_1^I \int_a^{a+\Delta x_1} \left(-\frac{1}{\Delta x_1}\right)\frac{1}{\Delta x_1}dx = u'(a) + \int_a^b f\phi_1^\Gamma dx$$

$$\Leftrightarrow -u_a\frac{1}{\Delta x_1} + u_1^I\frac{1}{\Delta x_1} = u'(a) + \int_a^b f\phi_1^\Gamma dx$$

$$\Leftrightarrow \frac{u_1^I - u_a}{\Delta x_1} = u'(a) + \int_a^b f\phi_1^\Gamma dx.$$

Adding this equation to (2.25) gives a square system describing the coefficients of our approximation.

$$\begin{bmatrix} A & -\frac{1}{\Delta x_1}e_1 \\ \frac{1}{\Delta x_1}e_1^T & -\frac{1}{\Delta x_1} \end{bmatrix} \begin{bmatrix} u^I \\ u_a \end{bmatrix} = \begin{bmatrix} \frac{u(b)}{\Delta x_{n+1}}e_n - \bar{f} \\ u'(a) + \int_a^b f\phi_1^\Gamma dx \end{bmatrix}. \tag{2.26}$$

2.6.3 Numerical Confirmation

We confirmed that a set of approximations found using the method in Section 2.6.2 converged to their exact solutions and analyzed the order of the convergence numerically. We decided to quantify the error using the L_2 norm, $\|u - u^h\|_{L_2}$. This error was then computed for different approximations, u^h, for different equidistant grids. The Poisson equations used for testing were (5.2) and (5.1). The errors of these approximations were then put into a logarithmic graph. This plot is presented in Figure 2.4. In this graph, we notice convergence. Moreover we have second order convergence.

2.7 Finite Volume Method For Poisson's Equation

A finite volume discretisation of Poisson's equation is presented in this section. The methods are presented in Leveque [8]. Analysis of the order of the methods has been omitted in this section. Instead we will motivate the second order of the methods by performing numerical experiments. For order testing we used equidistant grids.

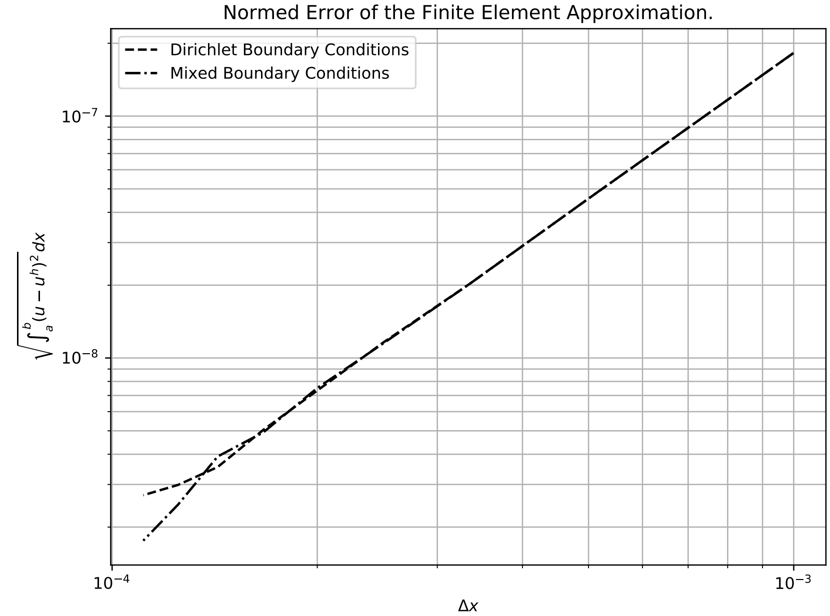

Figure 2.4: The errors $\|u - u^h\|_2$ for different approximations Δx. The approximations are found using the strategies presented in Section 2.6.2. At $\Delta x = 10^{-2}$ the error is 10^{-4} and at $\Delta x = 10^{-3}$ the error is 10^{-6}. This indicates that the method is second order.

2.7.1 Discretizing Poisson's Equation

In this chapter we follow Leveque [8, p.64-66]. The function, u, we wish to approximate is defined in $[a, b]$. With this in mind, we define the interior grid points, x_k $k = 1 \ldots n$, and the distances between these points Δx_i.

$$a < x_1 < x_2 < \cdots < x_n < b$$

$$\Delta x_i = x_i - x_{i-1},\ i = 1, \ldots, n+1$$

$$x_0 = a \text{ and } x_{n+1} = b.$$

We also define $x_{i+0.5}$ and $x_{i-0.5}$ as:

$$x_{i+0.5} = x_i + 0.5\Delta x_{i+1} \qquad x_{i-0.5} = x_i - 0.5\Delta x_i.$$

To illustrate these variables Figure 2.5 is provided.

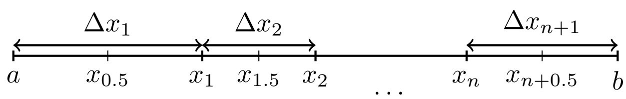

Figure 2.5: The grid used for the finite volume method.

We let $C_i = [x_{i-0.5}, x_{i+0.5}]$ be called the ith cell. These cells are used to define averages u_i of $u(C_i)$:

$$u(x_k) \approx u_k = \frac{1}{|C_i|}\int_{C_i} u\ dx.$$

We define our approximation as:

$$\hat{u}^h(x) = \begin{cases} u(a),\ x \in [a, x_{0.5}) \\ u_i,\ x \in C_i \\ u(b),\ x \in [x_{n-0.5}, b] \end{cases} . \tag{2.27}$$

To illustrate this approximation Figure 2.6 is provided.

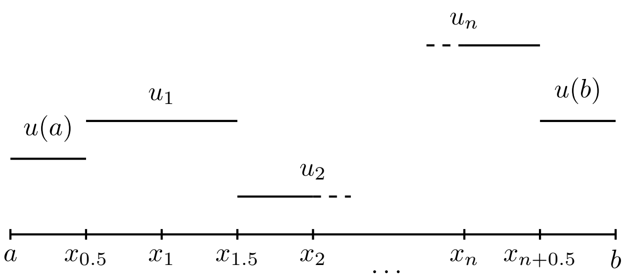

Figure 2.6: The approximation function $\hat{u}^h$.

To get u_i we use the following property of our problem:

$$\triangle u = f \Rightarrow \int_{C_i} u'' dx = \int_{C_i} f\ dx,\ i = 1, \dots, n.$$

Using the fundamental theorem of calculus we get:

$$u'(x_{i+0.5}) - u'(x_{i-0.5}) = \int_{C_i} f\ dx,\ i = 1, \dots, n. \tag{2.28}$$

Now $\hat{u}^h$ is discontinuous at $x_{k+0.5}$ $k = 0 \dots n$. To handle this we introduce the numerical flux, similar to Leveque [8, p.67 eq (4.10)], $F_{0.5}, F_{1.5}, \dots, F_{n+0.5}$.

$$u'(x_{k+0.5}) \approx F_{k+0.5} = \frac{u_{k+1} - u_k}{\Delta x_k},\ k = 0 \dots n.$$

An illustration of the numerical flux is presented in Figure 2.7.

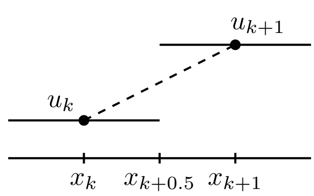

Figure 2.7: The numerical flux, $F_{k+0.5}$, is the slope of the dashed line.

Using the numerical flux as approximations of $u'(x_{k+0.5})$ in (2.28) gives:

$$F_{i+0.5} - F_{i-0.5} \approx u'(x_{i+0.5}) - u'(x_{i-0.5}) = \int_{C_i} f\ dx,\ i = 1, \dots, n.$$

The approximations coefficients, $\bar{u} = (u_1, \dots, u_n)^T$, are defined by the following linear system.

$$F_{i+0.5} - F_{i-0.5} = \int_{C_i} f\ dx,\ i = 1, \dots, n.$$

$$\Leftrightarrow \begin{cases} \frac{1}{\Delta x_2}(u_2 - u_1) - \frac{1}{\Delta x_1}(u_1 - u(a)) = \int_{C_1} f dx \\ \frac{1}{\Delta x_3}(u_3 - u_2) - \frac{1}{\Delta x_2}(u_2 - u_1) = \int_{C_2} f dx \\ \vdots \\ \frac{1}{\Delta x_{n+1}}(b - u_n) - \frac{1}{\Delta x_n}(u_n - u_{n-1}) = \int_{C_n} f dx \end{cases}$$

$$\Leftrightarrow \begin{cases} \frac{1}{\Delta x_1}u(a) - \left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2}\right)u_1 + \frac{1}{\Delta x_2}u_2 = \int_{C_1} f dx \\ \frac{1}{\Delta x_2}u_1 - \left(\frac{1}{\Delta x_2} + \frac{1}{\Delta x_3}\right)u_2 + \frac{1}{\Delta x_3}u_3 = \int_{C_2} f dx \\ \vdots \\ \frac{1}{\Delta x_n}u_{n-1} - \left(\frac{1}{\Delta x_n} + \frac{1}{\Delta x_{n+1}}\right)u_n + \frac{1}{\Delta x_{n+1}}u(b) = \int_{C_n} f dx \end{cases}.$$

$$\Leftrightarrow \quad \begin{aligned} A\bar{u} &= \left(\frac{e_1}{\Delta x_1}u(a) + \frac{e_n}{\Delta x_{n+1}}u(b)\right) - \bar{f} \\ [A]_{i,j} &= \begin{cases} \frac{1}{\Delta x_i} + \frac{1}{\Delta x_{i+1}},\ i = j \\ -\frac{1}{\Delta x_{i+1}},\ i = j+1 \\ -\frac{1}{\Delta x_{i+1}},\ j = i+1 \\ 0,\ \text{else} \end{cases} \\ \bar{f} &= \left(\int_{C_1} f\ dx, \dots, \int_{C_n} f\ dx\right). \end{aligned} \tag{2.29}$$

Dirichlet problem

To get $\bar{u} = (u_1, \dots, u_n)^T$, we solve (2.29) where $u(a)$ and $u(b)$ are given. If $f = 0$ then the linear system $Au^I = b$ is identical to that in the finite element case (2.24).

Neumann problem

For the one dimensional Neumann problem, (2.5), we have $u(b)$ given, but $u(a)$ is unknown. We use (2.29), but approximate $u(a)$ and move it to the left hand side to get:

$$\begin{bmatrix} A & -\frac{e_1}{\Delta x_1} \end{bmatrix} \begin{bmatrix} u^I \\ u_a \end{bmatrix} = \frac{e_n}{\Delta x_{n+1}}u(b) - \bar{f}, \tag{2.30}$$

where u_a is an approximation of $u(a)$. We have $n+1$ unknowns but only n equations, so we need to add one equation. Here we use the fact that we know $u'(a)$. To get the last equation we integrate our differential equation over the first segment:

$$u'' = f \Rightarrow \int_a^{x_{0.5}} u'' dx = \int_a^{x_{0.5}} f dx$$

$$\Leftrightarrow (u'(x_{0.5}) - u'(a)) = \int_a^{x_{0.5}} f dx.$$

u^h is discontinuous in 0.5, so we use the flux function $F_{0.5}$ to approximate the derivative. Using this approximation in the equation gives:

$$F_{0.5} = \left(\frac{u(x_1) - u(a)}{\Delta x_1}\right) = u'(a) + \int_a^{x_{0.5}} f dx$$

$$\Rightarrow \frac{u(x_1^I) - u(a)}{\Delta x_1} = u'(a) + \int_a^{x_{0.5}} f dx$$

$$\Rightarrow \frac{u(x_1^I) - u_a}{\Delta x_1} \approx u'(a) + \int_a^{x_{0.5}} f dx. \tag{2.31}$$

Adding equation (2.31) to the linear system in (2.30) gives us an expression for the coefficients of the approximation.

$$\begin{bmatrix} A & -\frac{e_1}{\Delta x_1} \\ \frac{e_1^T}{\Delta x_1} & -\frac{1}{\Delta x_1} \end{bmatrix} \begin{bmatrix} u^I \\ u(a) \end{bmatrix} = \begin{bmatrix} \frac{e_n}{\Delta x_{n+1}} u(b) - \bar{f} \\ u'(a) + \int_a^{x_{0.5}} f dx. \end{bmatrix} \tag{2.32}$$

2.7.2 Numerical Confirmation

To confirm that our finite volume discretisation is correct, we analyzed the errors of a set of approximations numerically. We used the L_2-norm when defining the error, $\|u(x) - u^h(x)\|_{L_2}$. Using the L_2 norm, our approximation $\hat{u}^h$ will never have second order. To give the method a chance in order testing we use linear interpolation of the averages at the center of each cell. This modified function is what we use as u^h. The test problems used in testing were the same as in Section 2.6.3, (5.2) and (5.1). The results, errors, of both tests are presented in Figure 2.8. In Figure 2.8 we can observe second order convergence for both discretisations. When $\Delta x = 10^{-3}$ the error is approximately 10^{-6} and when $\Delta x = 10^4$ the error is approximately 10^{-8}.

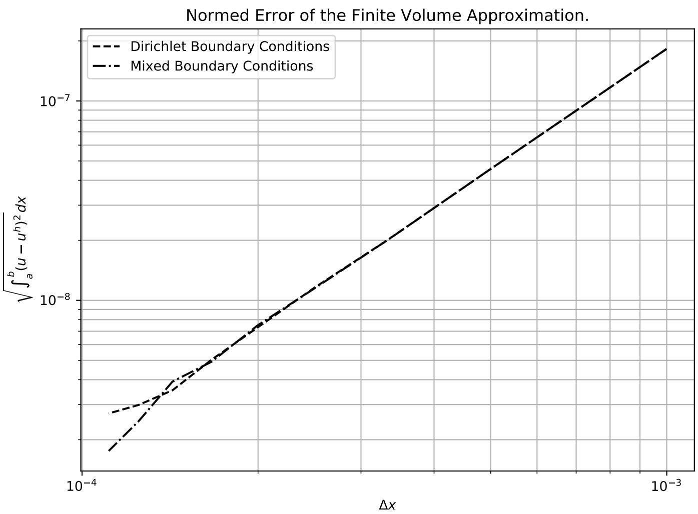

Figure 2.8: An error graph for the finite volume discretisation of the Poisson's equation in one dimension. Approximations were found using the finite volume method presented in Section 2.7.1. At $\Delta x = 10^{-2}$ the error was 10^{-4} and at $\Delta x = 10^{-3}$ the error was 10^{-6}. This indicates that the method is of second order.

Chapter 3

Convergence rate of the Algorithm

The convergence rate for the Dirichlet Neumann algorithm will be analyzed and numerically tested in this chapter. First, the convergence rate for non -mixed standard discretisations are analyzed and proved using linear algebra. The results from these three separate cases are combined into one theorem for the mixed case. This results is surprising. It sais that the convergence rate for mixed adaptive discretisations is indepedent of the grids chosen. It also shows that it is possible to find the convergence rate by analyzing the discretisations separately. This is made easier due to the clear structure of the discretisations. The same approach could be used for higher order methods and the multidimensional case. The linear systems would be harder to solve, but there would still be a similar structure to them. This chapter ends by analyzing the linear systems arising in the asymptotic case for a set of general discretisations. Using this approach might give more general information about the convergence rate. There is no clear way to extend this result to the multidimensional case, but a similar approach could be used to make statements of the convergence rate for higher order discretisations.

3.1 The Dirichlet-Neumann Algorithm FDM - FDM

In this section, we analyze the convergence rate of the Dirichlet-Neumann algorithm when using the finite difference discretisation to get an approximation of the solution to the transmission problem (2.6). We will use the finite difference discretisation introduced in Section 2.5.1.

3.1.1 The Method

The first step will introduce a new grid for our discrete solution. We use two equidistant methods, so we construct two equidistant grids. The spacing of the grids are defined by:

$$\Delta x = \frac{x_\Gamma - a}{n_1 + 1} \text{ and } \Delta y = \frac{b - x_\Gamma}{n_2 + 1},$$

where n_1 will be the resolution of the Dirichlet part and n_2 for the Neumann part. Following the steps of Section 2.4.1, using the finite difference method to approximate discrete solutions for both the Dirichlet and Neumann problem leads to the following terms in (2.11):

$$\begin{aligned} A^{(1)} = A_x,\ \bar{f}^{(1)} = \bar{f}_x,\ A_\Gamma^{(1)} = -\frac{e_{n_1}}{\Delta x},\ b^{(1)} = \lambda_1 \frac{e_1}{\Delta x}. \\ A^{(2)} = A_y,\ \bar{f}^{(2)} = \bar{f}_y,\ A_\Gamma^{(2)} = -\frac{e_1}{\Delta y},\ f_\Gamma^{(1)} = f_\Gamma^{(2)} = 0,\ b^{(2)} = \lambda \frac{e_{n_2}}{\Delta x} \\ d_\Gamma^{(2)} = -\frac{3}{2\Delta y},\ d^{(2)} = \frac{4e_1^T - e_2^T}{2\Delta y},\ d_\Gamma^{(1)} = \frac{3}{2\Delta x},\ d^{(1)} = -\frac{4e_{n_1}^T - e_{n_1-1}^T}{2\Delta x}, \end{aligned} \tag{3.1}$$

where A_x and A_y are the two A matrices and $\bar{f}_x$ and $\bar{f}_y$ are the $\bar{f}$s in the Dirichlet and Neumann problem respectfully. We get $d^{(1)}$ and $d^{(2)}$ by using a mirrored version of (2.15). The Dirichlet-Neumann algorithm follows by using the recursion defined in (2.11) with these terms and an initial guess w_{Γ_0}.

Numerical confirmation

The discretisation was set up and tested. To analyze the convergence of the discretisations of v and w, we used the test problem (5.3). Knowing the exact solutions, we quantify the error of the interpolated discrete approximations v^h and w^h with a modified L_2 norm:

$$\sqrt{\int_a^{x_\Gamma} (v - v^h)^2 dx + \int_{x_\Gamma}^b (w - w^h)^2 dx}. \tag{3.2}$$

With the error quantified, the error for different approximations of the test equation with different Δx and Δy were evaluated. For simplicity we let $\Delta x = \Delta y$. We then analyze the error for different Δx and collect them in Figure 3.1.1. Figure 3.1.1 shows that the approximations converges to the correct solutions when Δx goes to zero. Moreover, the convergence is second order.

3.1.2 Convergence Rate

Before we go into Theorem 3.1.1, we provide some Lemmas.

Lemma 3.1.1. *$A^{(1)}$ and $A^{(2)}$ have the form of A in Lemma 5.0.1 and 5.0.2.*

Proof.

$$\frac{2}{\Delta x} = \frac{1}{\Delta x} + \frac{1}{\Delta x} \text{ and } \frac{2}{\Delta y} = \frac{1}{\Delta y} + \frac{1}{\Delta y}.$$

□

Lemma 3.1.2.

$$\frac{1}{2\Delta x}\left(3 - \left(4e_{n_1}^T - e_{n_1-1}^T\right)\frac{(A^{(1)})^{-1}}{\Delta x} e_{n_1}\right) = \frac{1}{l_1} \tag{3.3}$$

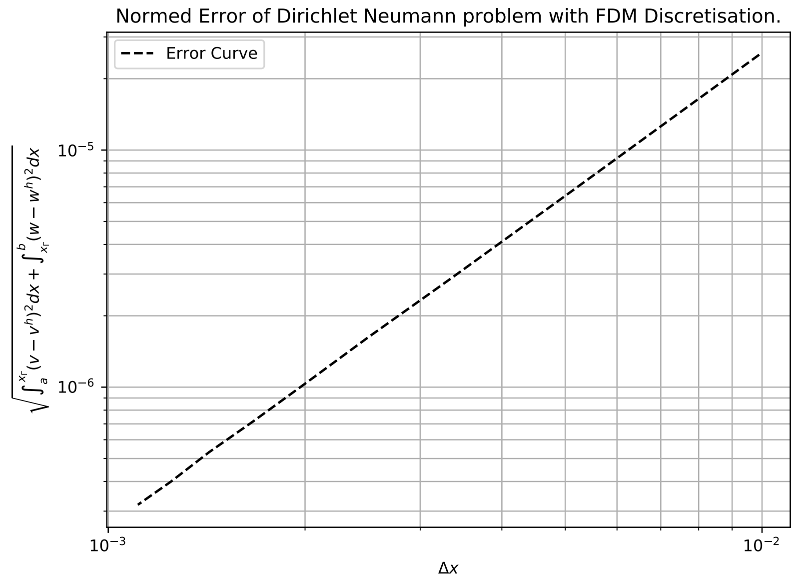

Figure 3.1: The error of the approximate solution compared to the correct solution of the Dirichlet-Neumann problem. Discretisations used are finite difference discretistaions, where $\Delta y = \Delta x$. We observe second order convergence.

Proof. First we use Lemma 5.0.1, motivated by Lemma 3.1.1:

$$e_{n_1}^T A^{-1} e_{n_1} = \Delta x \left(1 - \frac{\Delta x}{l}\right).$$

This is applied to the left hand side of (3.3) to get:

$$\frac{1}{2\Delta x}\left(3 - 4\left(1 - \frac{\Delta x}{l_1}\right) + e_{n_1-1}^T \frac{(A^{(1)})^{-1}}{\Delta x} e_{n_1}\right). \tag{3.4}$$

Here we use Lemma 5.0.2, again motivating with Lemma 3.1.1, to get the relation:

$$e_{n_1-1}^T A^{-1} e_{n_1} = \Delta x \left(1 - 2\frac{\Delta x}{l}\right).$$

Using this in (3.4) finishes the proof.

$$\frac{1}{2\Delta x}\left(3 - 4\left(1 - \frac{\Delta x}{l_1}\right) + \left(1 - 2\frac{\Delta x}{l_1}\right)\right) = \frac{1}{2\Delta x}\left(2\frac{\Delta x}{l_1}\right) = \frac{1}{l_1}.$$

□

Lemma 3.1.3.

$$\frac{1}{2\Delta y}\left(3 - \left(4e_1^T - e_2^T\right)\frac{(A^{(2)})^{-1}}{\Delta y} e_1\right) = -\frac{1}{l_2} \tag{3.5}$$

Proof. Lemma 3.1.1 tells us that we can use Lemma 5.0.1 to get the following relation:

$$e_n^T A^{-1} e_n = \Delta y \left(1 - \frac{\Delta y}{l}\right).$$

Using this on the left hand side of (3.5) gives:

$$\frac{1}{2\Delta y}\left(3 - 4\left(1 - \frac{\Delta y}{l_2}\right) + e_2^T \frac{(A^{(2)})^{-1}}{\Delta y} e_1\right). \tag{3.6}$$

Next we use Lemma 5.0.2, motivating with Lemma 3.1.1, to get another relation:

$$e_{n-1}^T A^{-1} e_n = \Delta y \left(1 - \frac{\Delta y}{l} - \frac{\Delta y}{l}\right)$$

Using this in (3.6) gives us the wanted result.

$$\frac{1}{2\Delta y}\left(3 - 4\left(1 - \frac{\Delta y}{l_2}\right) + \left(1 - 2\frac{\Delta y}{l_2}\right)\right) = \frac{1}{\Delta y}\left(-\frac{\Delta y}{l_2}\right) = -\frac{1}{l_2}.$$

□

Theorem 3.1.1. *The convergence rate of the Dirichlet-Neumann algorithm approximating* (2.6) *with the finite difference discretisations presented in Section 3.1.1 is:*

$$\mu = \left|\frac{\lambda_1 l_2}{\lambda_2 l_1}\right|,$$

where

$$l_1 = x_\Gamma - a,\ l_2 = b - x_\Gamma.$$

Proof. We know from (2.13) that the convergence rate is:

$$\mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{d_\Gamma^{(1)} - d^{(1)}(A^{(1)})^{-1} A_\Gamma^{(1)}}{d_\Gamma^{(2)} - d^{(2)}(A^{(2)})^{-1} A_\Gamma^{(2)}}\right|,$$

Using the terms for the finite difference discretisation, (3.1), gives the following:

$$\mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{\frac{-3}{2\Delta x} - \left(\frac{4e_{n_1}^T - e_{n_1-1}^T}{2\Delta x}\right)(A^{(1)})^{-1}\left(-\frac{e_{n_1}}{2\Delta x}\right)}{\frac{3}{2\Delta y} + \left(\frac{4e_1^T - e_2^T}{2\Delta y}\right)(A^{(2)})^{-1}\left(-\frac{e_1}{2\Delta y}\right)}\right|$$

$$\Leftrightarrow \mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{\frac{1}{2\Delta x}\left(3 - (4e_{n_1}^T - e_{n_1-1}^T)\frac{(A^{(1)})^{-1}}{\Delta x} e_{n_1}\right)}{\frac{1}{2\Delta y}\left(3 - (4e_1^T - e_2^T)\frac{(A^{(2)})^{-1}}{\Delta y} e_1\right)}\right|. \tag{3.7}$$

Next we analyze the nominator and denominator following terms separately in Lemma 3.1.2 and Lemma 3.1.3. To prove the theorem we use Lemma 3.1.2 and 3.1.3 in (3.7).

$$\mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{\frac{1}{l_1}}{-\frac{1}{l_2}}\right| = \left|\frac{\lambda_1 l_2}{\lambda_2 l_1}\right|.$$

□

Numerical confirmation

We know the convergence rate of the Dirichlet-Neumann algorithm solving the transmission problems with finite difference discretisations by Theorem 3.1.1. We tested this remark by performing the algorithm for a number of different transmission problems and comparing the convergence rates to the expected rates. Two tests were performed. The first test varied λ_1 and λ_2 and the second varied l_1 and l_2. In each test we set $\lambda_1 = \lambda_2 = 1$ and $l_1 = l_2 = 1$ unless stated otherwise. For each test we found the deviation from the expected convergence rates for three different resolutions. The deviation was assessed as:

$$\max_{k=0,\ldots,9} \left| \frac{\lambda_1 l_2}{\lambda_2 l_1} - \frac{e_{k+1}}{e_k} \right|,$$

where $e_k = |w^k(x_\Gamma) - w(x_\Gamma)|$. This deviation was calculated five separate times for each test and the mean value was the result. The results of these tests are presented in Table 3.1 and 3.2. They show that the deviation from the expected result is negligible.

Resolution	$\lambda_1 = 2$	$\lambda_2 = 0.5$	$\lambda_1 = 4$	$\lambda_2 = 0.25$
$n_1 = 100,\ n_2 = 100$	1.3e-11	1.3e-11	2.9e-11	2.9e-11
$n_1 = 20,\ n_2 = 100$	4.6e-12	4.6e-12	3.6e-11	3.6e-11
$n_1 = 100,\ n_2 = 20$	8.1e-12	8.1e-12	1.7e-11	1.7e-11

Table 3.1: Deviation of the convergence rate for the Dirichlet-Neumann algorithm with finite difference discretisations. In this test we vary λ_1 and λ_2.

Resolution	$l_1 = 2$	$l_2 = 0.5$	$l_1 = 4$	$l_2 = 0.25$
$n_1 = 100,\ n_2 = 100$	1.7e-11	7.4e-12	4.1e-11	4.9e-12
$n_1 = 20,\ n_2 = 100$	4.6e-12	6.4e-13	3.2e-12	3.2e-12
$n_1 = 100,\ n_2 = 100$	7.3e-11	4.5e-12	4.1e-11	3.4e-11

Table 3.2: Deviation of the convergence rate for the Dirichlet-Neumann algorithm with finite difference discretisations. In this test we vary l_1 and l_2.

3.2 The Dirichlet-Neumann Algorithm FEM - FEM

In this section we find the convergence rate of the Dirichlet-Neumann algorithm when using the finite element method presented in Section 2.6 for both discretisations. We use the result of Section 2.4.2, (2.13), to get an expression of the convergence rate. The result, Theorem 3.2.1, is then confirmed through numerical testing.

3.2.1 The Method

First we need to define our grids for both discretisations. The grid points for the Dirichlet part will be denoted $x^I \subset (a, x_\Gamma)$, $\#x^I = n_1$ and the grid points for the Neumann part will be denoted $y^I \subset (x_\Gamma, b)$, $\#y^I = n_2$. The distances between these grid points and boundary points will be denoted Δx for x^I and Δy for y^I. An illustration to visualize these terms is provided in Figure 3.2.

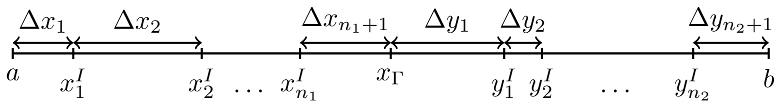

Figure 3.2: The grid used for the Dirichlet-Neumann algorithm.

With the grid defined, we set up finite element discretisations for both the Dirichlet and Neumann problems in (2.7). From these discretisations we get the following terms of (2.11):

$$
\begin{aligned}
A^{(1)} = A_x,\ \bar{f}^{(1)} = \bar{f}_x,\ b^{(1)} &= \frac{\lambda_1}{\Delta x_1} e_1,\ A_\Gamma^{(1)} = -\frac{1}{\Delta x_{n_1+1}} e_{n_1}. \\
A^{(2)} = A_y,\ \bar{f}^{(2)} = \bar{f}_y,\ b^{(2)} &= \frac{\lambda_2}{\Delta y_{n_2+1}} e_{n_2},\ A_\Gamma^{(2)} = -\frac{1}{\Delta y_1} e_1, \\
d^{(2)} = \frac{e_1^T}{\Delta y_1},\ d_\Gamma^{(2)} &= -\frac{1}{\Delta y_1},\ f_\Gamma^{(2)} = \int_{x_\Gamma}^{b} (\phi_y)_1^\Gamma f^{(2)} dx, \\
d^{(1)} = -\frac{e_{n_1}^T}{\Delta x_{n_1+1}},\ d_\Gamma^{(1)} &= \frac{1}{\Delta x_{n+1}},\ f_\Gamma^{(1)} = \int_{x_{n_1}}^{x_\Gamma} f^{(1)} (\phi_x)_2^\Gamma dx,
\end{aligned}
\tag{3.8}
$$

where ϕ_x and ϕ_y are the basis functions, A_x and A_y are the A matrices and $\bar{f}_x$ and $\bar{f}_y$ are $\bar{f}$ from the Dirichlet and Neumann discretisation respectably. To get $d^{(1)}$ and $d_\Gamma^{(1)}$, we use the same process getting the last equation for the Neumann problem in Section 2.6.2.

Numerical Confirmation

This discretisation was set up and tested. To analyze the convergence of the approximations v^h and w^h, we use the test problem (5.3). Just like in the finite difference case we quantify the error of the approximations as (3.2). This error was evaluated for different approximations of the test problem with equidistant Δx and Δy. We also let $\Delta x = \Delta y$ for simplicity. The errors are presented in Figure 3.2.1. The results show that we have convergence and that the convergence is second order.

3.2.2 Convergence Rate

Theorem 3.2.1. *The convergence rate of the Dirichlet-Neumann algorithm approximating* (2.6) *with the finite element discretisations in Section 2.6 is:*

$$\mu = \left| \frac{\lambda_1 l_2}{\lambda_2 l_1} \right|,$$

where

$$l_1 = x_\Gamma - a,\ l_2 = b - x_\Gamma.$$

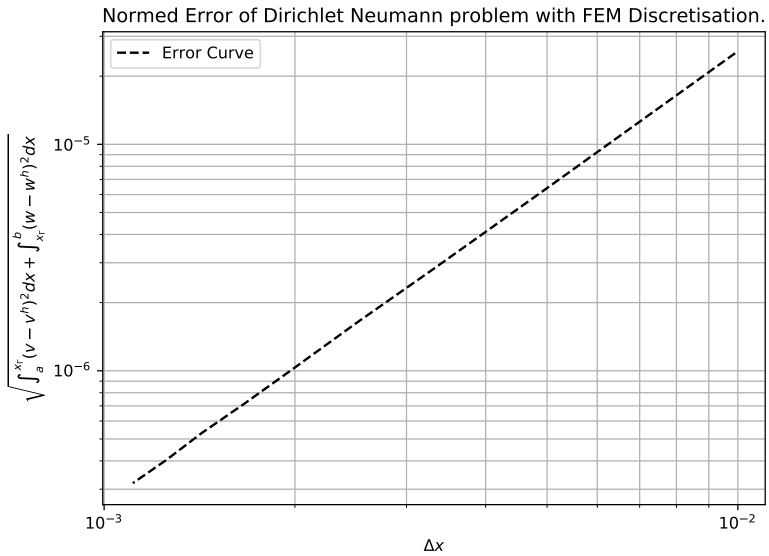

Figure 3.3: The norm of the error when using different Δx when $\Delta x = \Delta y$.

Proof. Using the result of Section 2.4.2, we know that the convergence rate of the algorithm is (2.13). Putting (3.8) into (2.13) leads to:

$$\mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{d_\Gamma^{(1)} - (d^{(1)})(A^{(1)})^{-1}(A_\Gamma^{(1)})}{d_\Gamma^{(2)} - (d^{(2)})(A^{(2)})^{-1}(A_\Gamma^{(2)})}\right|$$

$$= \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{\left(\frac{1}{\Delta x_{n_1+1}}\right) - \left(-\frac{e_{n_1}^T}{\Delta x_{n_1+1}}\right)(A^{(1)})^{-1}\left(-\frac{e_{n_1}}{\Delta x_{n_1+1}}\right)}{\left(-\frac{1}{\Delta y_1}\right) - \left(\frac{1}{\Delta y_1}e_1^T\right)(A^{(2)})^{-1}\left(-\frac{1}{\Delta y_1}e_1\right)}\right|$$

$$\Leftrightarrow \mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{\left(\frac{1}{\Delta x_{n_1+1}}\right)\left(1 - \frac{e_{n_1}^T(A^{(1)})^{-1}e_{n_1}}{\Delta x_{n_1+1}}\right)}{\left(-\frac{1}{\Delta y_1}\right)\left(1 - \frac{e_1^T(A^{(2)})^{-1}e_1}{\Delta y_1}\right)}\right|. \tag{3.9}$$

Next we use Lemma 5.0.1. We notice that the matrices $A^{(1)}$ and $A^{(2)}$ paired with Δx and Δy respectively have the same form as A in the lemma. Using this we know the following:

$$\begin{aligned} \frac{e_{n_1}^T(A^{(1)})^{-1}e_{n_1}}{\Delta x_{n_1+1}} &= 1 - \frac{\Delta x_{n_1+1}}{l_1} \\ \frac{e_1^T(A^{(2)})^{-1}e_1}{\Delta y_1} &= 1 - \frac{\Delta y_1}{l_2} \end{aligned} \tag{3.10}$$

Adding (3.9) into (3.9) gives the wanted result.

$$\mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left| \frac{\left(\frac{1}{\Delta x_{n_1+1}}\right)\left(1-\left(1-\frac{\Delta x_{n_1+1}}{l_1}\right)\right)}{\left(-\frac{1}{\Delta y_1}\right)\left(1-\left(1-\frac{\Delta y_1}{l_2}\right)\right)} \right| = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{\frac{1}{l_1}}{-\frac{1}{l_2}}\right|. = \left|\frac{\lambda_1 l_2}{\lambda_2 l_1}\right|.$$

□

The result of Section 2.4.2 states that the convergence rate is independent of what grid you use. The convergence rate is entirely dependent on the terms of the transmission problem.

Numerical Confirmation

We tested four different strategies when choosing the grids x^Is and y^Is for each test performed. If Theorem 3.2.1 is true, then there should be no difference which strategy is used.

The first strategy is the equidistant distribution. The second strategy is using pseudo-random uniform distribution, the third has grid points concentrated around the interface, x_Γ, and the fourth has grid points concentrated around the boundary points a and b. The points for the second and third points are generated by getting an uniformly distributed set between $[0, 1]$. These sets are then put into the functions:

$$f_l(x) = \frac{e^x - 1}{e - 1} \text{ and } f_u(x) = f_l^{-1}(x) = \ln(x(e-1)+1).$$

This creates new sets that are either concentrated around 0 (f_l) or 1 (f_u). Next the values in the sets are sorted, scaled, and offset to be between the wanted boundary values. So for the third strategy we used f_u for the x_i^Is and f_l for the y_i^Is, the fourth is vice versa.

Three tests were performed. The first test varied λ_1, λ_2, the second $l_1,\ l_2$, and the third the resolutions n_1 and n_2. In all tests we set $\lambda_1 = \lambda_2 = 1,\ l_1 = l_2 = 1$ and $n_1 = n_2 = 100$ unless stated otherwise. For each test we found the deviation from the expected convergence rates for each one of our four strategies. The deviation was assessed as:

$$\max_{k=0,\ldots,9} \left| \frac{\lambda_1 l_2}{\lambda_2 l_1} - \frac{e_k}{e_{k-1}} \right|,$$

where $e_k = |w^k(x_\Gamma) - w(x_\Gamma)|$. This deviation was found five separate times and the mean value was the result. The results of the tests are presented in Tables 3.3, 3.4 and 3.5. The results indicate that the deviation is negligible.

Strategy	$\lambda_1 = 2$	$\lambda_2 = 0.5$	$\lambda_1 = 4$	$\lambda_2 = 0.25$
Equidistant	5.8e-13	5.8e-13	7.1e-9	7.11e-9
Uniform	3.5e-10	3.6e-10	9.7e-8	2.8e-8
Dense Interface	1.5e-9	1.6e-10	3.5e-7	4.1e-8
Sparse Interface	2.8e-10	3.6e-9	1.6e-7	7.1e-8

Table 3.3: Deviation when varying λ_1 and λ_2

Strategy	$l_1 = 2$	$l_2 = 0.5$	$l_1 = 4$	$l_2 = 0.25$
Equidistant	6.3e-13	3.7e-11	7.1e-9	1.6 e-9
Uniform	1.8e-8	3.6e-10	1.2e-7	7.6e-8
Dense Interface	1.0e-10	7.2e-10	4.0e-8	4.3e-7
Sparse Interface	3.4e-10	4.9e-10	3.3e-7	8.8e-8

Table 3.4: Deviation when varying l_1 and l_2

Strategy	$l_1 = 2,\ n_1 = 10,\ n_2 = 100$	$l_1 = 2,\ n_1 = 100,\ n_2 = 10$
Equidistant	1.6e-12	8.0e-13
Uniform	2.9e-10	1.4e-11
Dense Interface	1.4e-9	7.5e-12
Sparse Interface	4.0e-10	2.2e-11

Table 3.5: Deviation when varying n_1 and n_2

3.3 The Dirichlet-Neumann Algorithm FVM - FVM

This section analyzes the convergence rate of the finite volume method. We first set up the Dirichlet-Neumann problem using the discretisations presented in Section 2.7.1. The terms relating to the convergence rate in the Dirichlet-Neumann algorithm are identical to the terms in the finite element case.

3.3.1 The Method

The grid chosen will be the same grid as in the finite element case, Figure 3.2. We set up the discretisations of the Dirichlet part and Neumann part of (2.11). From these discretisations we get the following terms (2.11):

$$\begin{aligned}
A^{(1)} = A_x,\ \bar{f}^{(1)} = -\bar{f}_x,\ b^{(1)} = \frac{\lambda_1}{\Delta x_1} e_1 \text{ and } A_\Gamma^{(1)} = -\frac{1}{\Delta x_{n_1+1}} e_{n_1} \\
A^{(2)} = A_y,\ \bar{f}^{(2)} = -\bar{f}_y,\ b^{(2)} = \frac{\lambda_2}{\Delta y_{n_2+1}} e_{n_2},\ A_\Gamma^{(2)} = -\frac{1}{\Delta y_1} e_1, \\
d^{(2)} = \frac{e_1^T}{\Delta y_1},\ d_\Gamma^{(2)} = -\frac{1}{\Delta y_1}, d^{(1)} = -\frac{e_{n_1}^T}{\Delta x_{n_1+1}},\ d_\Gamma^{(1)} = \frac{1}{\Delta x_{n+1}}, \\
f_\Gamma^{(2)} = \int_{x_\Gamma}^{y_{0.5}} f_2(x)dx,\ f_\Gamma^{(1)} = \int_{x_{n_1-0.5}}^{x_\Gamma} f_1(x)dx,
\end{aligned} \tag{3.11}$$

where A_x and A_y are A and $\bar{f}^{(1)}$ and $\bar{f}^{(2)}$ are $\bar{f}$ in the Dirichlet and Neumann part respectfully. The terms $d^{(1)}$, $d_\Gamma^{(1)}$, and $f_\Gamma^{(1)}$ are chosen using the same method when creating the last equation of the Neumann discretisation (2.32).

Numerical Confirmation

We perform the same procedure when testing the convergence rates as the finite difference (Section 3.1) and finite element (Section 3.2) discretisations. We evaluate the error in the modified L_2 error, (3.2), and use test equation (5.3) for testing. We simplify testing by choosing equidistant grids Δx and Δy with $\Delta x = \Delta y$. The approximation will be constant segments so we choose to analyze the approximation u^h, which is created by linear interpolation of the mean function values at x_i and y_i. These approximations are then compared to the exact solutions using the error we defined. These errors are presented in Figure 3.3.1. In Figure 3.3.1, we can see that the error seems to converge with second order convergence.

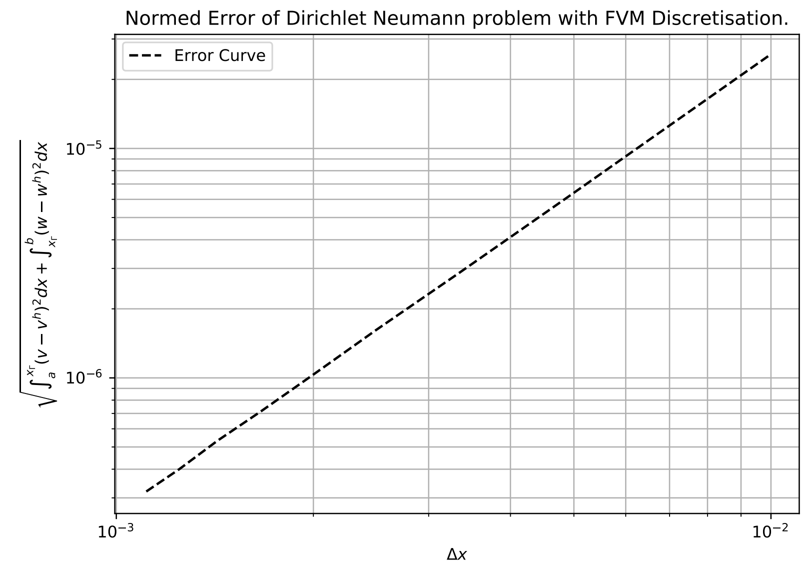

Figure 3.4: Error of the approximation of the Dirichlet-Neumann problem when using an equidistant grids with $\Delta x = \Delta y$ for the finite volume method.

3.3.2 Convergence Rate

Theorem 3.3.1. *The convergence rate of the Dirichlet-Neumann algorithm approximating* (2.11) *with the finite volume discretisations in Section 2.7.1 is:*

$$\mu = \left| \frac{\lambda_1 l_2}{\lambda_2 l_1} \right|,$$

where

$$l_1 = x_\Gamma - a, \;\; l_2 = b - x_\Gamma.$$

Proof. The finite volume and finite element discretisation of the discrete Dirichlet-Neumann algorithm, (2.11), has the same terms $d^{(1)}$, $d^{(2)}$, $d_\Gamma^{(1)}$, $d_\Gamma^{(2)}$, $A^{(1)}$, $A^{(2)}$, $A_\Gamma^{(1)}$, and $A_\Gamma^{(2)}$ when using the same grid. We know from (2.13) that these are the only terms that influence the convergence rate. Therefore the convergence rate of the finite volume discretisation and the finite element discretisation have to be the same. By Theorem 3.2.1 we know that the convergence rate is as stated. □

Numerical Confirmation

We know from Theorem 3.3.1 that the convergence rate is independent of the grid used. The convergence rate is entirely dependent on the terms of the transmission problem, as with the finite element case in Section 3.2.2. We therefore used the same four grid choosing strategies and tests presented in Section 3.2.2. The results of these tests are presented in Table 3.6, 3.7 and 3.8. We observe that the tables generated are similar to the finite element case (Table 3.5, 3.4 and 3.3). This makes sense as both of the discretisations becomes identical when $f = 0$. Moreover the deviations are negligible.

Strategy	$\lambda_1 = 2$	$\lambda_2 = 0.5$	$\lambda_1 = 4$	$\lambda_2 = 0.25$
Equidistant	1.3e-12	1.3e-12	2.8e-8	2.8e-8
Uniform	3.6e-10	1.2e-9	3.9e-7	1.0e-7
Dense Interface	5.1e-9	6.8e-10	8e-7	4.7e-7
Sparse Interface	4.1e-10	1.8e-10	1.8e-7	6.1e-8

Table 3.6: Deviation when varying λ_1 and λ_2

Strategy	$l_1 = 2$	$l_2 = 0.5$	$l_1 = 4$	$l_2 = 0.25$
Equidistant	1.4e-12	7.4e-11	2.8e-8	5.6 e-9
Uniform	3.1e-10	3.3e-10	7e-7	7.8e-7
Dense Interface	5.3e-10	7.9e-10	7.9e-7	3.8e-7
Sparse Interface	6.5e-10	3.6e-10	4.0e-7	2.6e-7

Table 3.7: Deviation when varying l_1 and l_2

Strategy	$l_1 = 2,\ n_1 = 10,\ n_2 = 100$	$l_1 = 2,\ n_1 = 100,\ n_2 = 10$
Equidistant	3.8e-12	1.5e-12
Uniform	8.6e-10	4.6e-11
Dense Interface	5.2e-10	4.6e-11
Sparse Interface	5.2e-10	1.1e-11

Table 3.8: Deviation when varying n_1 and n_2

3.4 Convergence Rate for Mixed Discretisations

In this section we analyze the convergence of the Dirichlet-Neumann algorithm when different discretisations are used. We use conclusions made from Chapters 3.1, 3.2 and 3.3 to combine them into one theorem about the convergence rate.

3.4.1 The Method

Mixed discretisations use one discretisation method to approximate the Dirichlet part and another to approximate Neumann part of the problem. Doing so you end up with the terms:

$$A^{(1)},\ b^{(1)},\ \bar{f}^{(1)},\ A_\Gamma^{(1)},\ d^{(1)},\ d_\Gamma^{(1)} \text{ and } f_\Gamma^{(1)}, \tag{3.12}$$

from the Dirichlet discretisations and

$$A^{(2)},\ b^{(2)},\ \bar{f}^{(2)},\ A_\Gamma^{(2)},\ d^{(2)},\ d_\Gamma^{(2)} \text{ and } f_\Gamma^{(2)}, \tag{3.13}$$

from the Neumann discretisation in (2.11).

Numerical Confirmation

The L_2 error, (3.2), was found for four different combinations with different grids. The combinations were:

- Finite difference for v and finite element for w
- Finite volume for v and finite difference for w
- Finite element for v and finite volume for w
- Finite volume for v and finite element for w

The grids were chosen to be equidistant where $\Delta x = \Delta y$. The test problem used was (5.3). With the error defined, problem set, and discretisations chosen, we generate and analyze a wide range of approximations using the four different discretisation pairs. To generate the approximations for the different Δx the Dirichlet-Neumann algorithm is used. We let the algorithm go until $|w^k(x_\Gamma) - w^{k+1}(x_\Gamma)| < 10^{-8}$. All of these errors are presented in Figure 3.4.1. The results show that all the approximations converge to the exact solutions. Moreover, they all have seem to have second order convergence.

3.4.2 Convergence Rate

Now we combine Theorem 3.1.1, 3.2.1, and 3.3.1 into one general theorem, Theorem 3.4.1.

Theorem 3.4.1. *The convergence rate of the Dirichlet-Neumann algorithm approximating* (2.6) *with any mix of the discretisations presented in Section 2.5.1, 2.7.1 and 2.6.2 is:*

$$\mu = \left|\frac{\lambda_1 l_2}{\lambda_2 l_1}\right|,$$

where

$$l_1 = x_\Gamma - a,\ l_2 = b - x_\Gamma.$$

Proof. We know that the terms effecting the rate of convergence for the finite element and finite volume cases are identical. Therefore mixing the terms for the two discretisations will have no effect on the convergence rate. We can consider them equivalent from a convergence rate point of view.

That leaves mixing a finite element discretisation with a finite difference discretisation. To analyze these cases we first write out the formula for the convergence rate (2.13).

$$\mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{d_\Gamma^{(1)} - d^{(1)}(A^{(1)})^{-1}A_\Gamma^{(1)}}{d_\Gamma^{(2)} - d^{(2)}(A^{(2)})^{-1}A_\Gamma^{(2)}}\right|.$$

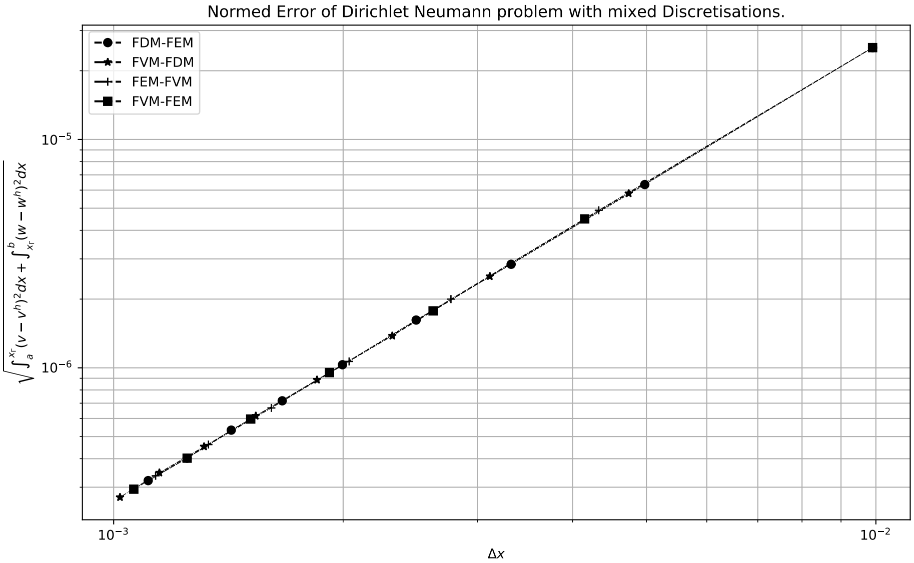

Figure 3.5: Error of the approximation of the Dirichlet-Neumann problem when using an equidistant grids with $\Delta x = \Delta y$ for different mixed discretisations. Δx was chosen slightly irregularly so that each of the markers is shown in the graph.

We re-write this as:

$$\mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{C_{\text{DIR}}}{C_{\text{NEU}}}\right|, \tag{3.14}$$

where

$$C_{\text{DIR}} = d_\Gamma^{(1)} - d^{(1)}(A^{(1)})^{-1}A_\Gamma^{(1)} \text{ and } C_{\text{NEU}} = d_\Gamma^{(2)} - d^{(2)}(A^{(2)})^{-1}A_\Gamma^{(2)}.$$

This isolates the terms that are dependent on the Dirichlet discretisation in C_{DIR} and the terms that are dependent on the Neumann discretisation in C_{NEU}. We now analyze what these variables are for the finite difference, finite element, and finite volume discretisation.

For the finite difference discretisation we know that:

$$C_{\text{DIR}}^{\text{FDM}} = \frac{1}{l_1} \text{ and } C_{\text{NEU}}^{\text{FDM}} = -\frac{1}{l_2},$$

from Lemma 3.1.2 and 3.1.3. For the finite element and finite volume case we get:

$$C_{\text{DIR}}^{\text{FVM}} = C_{\text{DIR}}^{\text{FEM}} = \frac{1}{l_1} \text{ and } C_{\text{NEU}}^{\text{FVM}} = C_{\text{NEU}}^{\text{FEM}} = -\frac{1}{l_2},$$

by using (3.9) in (3.10). From this we get the following result:

$$\begin{aligned} |C_{\text{DIR}}^{\text{FVM}}| = |C_{\text{DIR}}^{\text{FEM}}| = |C_{\text{DIR}}^{\text{FDM}}| = \left|\frac{1}{l_1}\right| \\ |C_{\text{NEU}}^{\text{FVM}}| = |C_{\text{NEU}}^{\text{FEM}}| = |C_{\text{NEU}}^{\text{FDM}}| = \left|\frac{1}{l_2}\right|. \end{aligned} \tag{3.15}$$

This means that for all combinations of the three discretisations we get:

$$\mu = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{C_{\text{DIR}}^{\text{D1}}}{C_{\text{NEU}}^{\text{D2}}}\right| = \left|\frac{\lambda_1 l_2}{\lambda_2 l_1}\right|, \ \forall \text{D1,D2} \in \{\text{FDM,FEM,FVM}\},$$

which is what we wanted to prove. □

Numerical Confirmation

To ensure that Theorem 3.4.1 is true, we performed several numerical tests. The three discretisation combinations we analyzed was:

1. Finite difference for v and finite element for w
2. Finite volume for v and finite difference for w
3. Finite element for v and finite volume for w
4. Finite volume for v and finite element for w

We set the resolution of the finite difference discretisation to $n = 100$ and use the same grid choosing strategies as in Section 3.2.2 for the finite element and finite volume discretisations. Only one Dirichlet-Neumann problem was used when testing. The terms of this Dirichlet-Neumann problem was $\lambda_1 = 1,\ \lambda_2 = 2,\ l_1 = 1,\ l_2 = 1,\ f^{(1)} = f^{(2)} = 0$. We then found the devation of the expected convergence rate to the actual rate by defining and then computing the following devation:

$$\max_{k=1,\dots,10} \left|\frac{\lambda_1 l_2}{\lambda_2 l_1} - \frac{e_k}{e_{k-1}}\right|,$$

where $e_k = |w^k(x_\Gamma) - w(x_\Gamma)|$. This deviation was calculated five times for each of the tests and the mean value was chosen as the result. The devations are presented in Table 3.9. We observe by the very small deviations the results are as expected.

Strategy	FDM-FEM	FVM-FDM	FEM-FVM	FVM-FEM
Equidistant	3.1e-12	2.9e-13	3.8e-12	1.6e-12
Uniform	7.2e-10	3.6e-11	8.4e-10	2.6e-11
Dense Interface	2.5e-8	6.9e-11	1.2e-9	2.9e-11
Sparse Interface	2.6e-9	4.0e-11	3.6e-10	9.4e-12

Table 3.9: Deviation of theoretical convergence rate with numerical when mixing discretisations.

3.5 Convergence Rate for General Discretisations

Using semi-discrete analysis gives statements on the asymptotic case. In this section we will take that approach. First we introduce a set of discretisations of (2.4) of the form (2.9) and then we introduce a set of discretisations of (2.5) of the form (2.10). First (2.4):

$$v''(x) = f,\ x \in [a, x_\Gamma],\ v(a) \text{ and } v(x_\Gamma) \text{ given.} \tag{3.16}$$

In this section we impose that f is chosen so the differential equation has a unique solution in a given function space $\mathcal{V}$. The homogeneous solutions to this problem are lines, so we impose that polynomials of degree one are included in $\mathcal{V}$. To approximate v in (3.16) we set up a system of the following form:

$$A^{(1)}\bar{v} = b^{(1)}v(a) - A_\Gamma^{(1)}v(x_\Gamma) + \bar{f}, \tag{3.17}$$

where $\bar{f} = \mathbf{0}$ if $f = 0$. With this we define a convergent Poisson-Dirichlet discretisation.

Definition 3.5.1. *Let $v_1^h, \dots, v_n^h$ be n functions on $[a, x_\Gamma]$. Then let $v^h(x) = \sum_{i=1}^n \bar{v}_i v_i^h(x)$, where $\bar{v}$ is defined by solving a system on the form (3.17) given some differential equation (3.16) with exact solution v. A discretisation on this form is called a* ***convergent Poisson-Dirichlet discretisation*** *on $\mathcal{V}$ if $\|v - v^h\|_{L_2} \to 0,\ n \to \infty$ for all $v \in \mathcal{V}$ and pointwise convergent on the boundary.*

Now we handle Neumann problems. We wish to approximate w in the following differential equation:

$$w''(x) = f,\ x \in [a, x_\Gamma],\ w'(x_\Gamma) \text{ and } w(b) \text{ given,} \tag{3.18}$$

where f is chosen so the differential equation has a unique solution in some function space $\mathcal{W}$. The homogeneous solution to this problem is a first degree polynomial, so we impose that they are included in $\mathcal{W}$. To approximate w in this equation we will set up a system on the following form:

$$\begin{bmatrix} A^{(2)} & A_\Gamma^{(2)} \\ d^{(2)} & d_\Gamma^{(2)} \end{bmatrix} \begin{bmatrix} \bar{w} \\ w^h(x_\Gamma) \end{bmatrix} = \begin{bmatrix} w(b)b^{(2)} + \bar{f} \\ w'(x_\Gamma) \end{bmatrix}, \tag{3.19}$$

where $\bar{f} = \mathbf{0}$ if $f = 0$. With this we define a convergent Poisson-Neumann discretisation.

Definition 3.5.2. *Let $w_1^h, \dots, w_n^h$ be n functions on $[x_\Gamma, b]$. Then let $w^h(x) = \sum_{i=1}^n \bar{w}_i w_i^h(x)$, where $\bar{w}$ is defined by solving a system on the form (3.17) given some differential equation (3.18) with exact solution w. A discretisation on this form is called a* ***convergent Poisson-Neumann discretisation*** *on $\mathcal{W}$ if $\|w - w^h\|_{L_2} \to 0,\ n \to \infty$ for all $w \in \mathcal{W}$, pointwise convergent on the boundary, and its limit function satisfies the Neumann condition.*

With these definitions the following theorem is presented.

Theorem 3.5.1. *Choose any convergent Poisson-Dirichlet discretisation paired with a function space $\mathcal{P}_1 \subset \mathcal{V}$ and n_1 functions $v_1^h, \dots, v_{n_1}^h$ and Poisson-Neumann discretisation paired with a function space $\mathcal{P}_1 \subset \mathcal{W}$ and n_2 functions $w_1^h, \dots, w_{n_2}^h$. If these discretisations together with a discrete approximation of the derivative at x_Γ, $\forall v \in \mathcal{V}: \ d_\Gamma^{(1)} v^h(x_\Gamma) + d^{(1)}\bar{v} \to v'(x_\Gamma),\ n_1 \to \infty$, are used in the Dirichlet-Neumann algorithm,* (2.11)*, then the convergence rate is*

$$\left|\frac{\lambda_1 l_2}{\lambda_2 l_1}\right|, \ n_1, n_2 \to \infty,$$

were $l_1 = x_\Gamma - a,\ l_2 = b - x_\Gamma$.

Proof. From (2.13) we know that the convergence rate of the algorithm is:

$$\mu = |\Sigma| = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{d_\Gamma^{(1)} - d^{(1)}(A^{(1)})^{-1}A_\Gamma^{(1)}}{d_\Gamma^{(2)} - d^{(2)}(A^{(2)})^{-1}A_\Gamma^{(2)}}\right|. \tag{3.20}$$

The discretisations are expressed on the same form as (2.9) and (2.10). In this proof we will first analyze the nominator and then the denominator:

$$d_\Gamma^{(1)} - d^{(1)}(A^{(1)})^{-1}A_\Gamma^{(1)} = d_\Gamma^{(1)} + d^{(1)}(A^{(1)})^{-1}(-A_\Gamma^{(1)}) = d_\Gamma^{(1)} + d^{(1)}\bar{h},$$

if $A^{(1)}\bar{h} = -A_\Gamma^{(1)}$. From (3.17) we know that:

$$A^{(1)}\bar{v}^{k+1} = -A_\Gamma^{(1)} w^k(x_\Gamma) + b^{(1)} v(a) + \bar{f}^{(1)}$$

is a discretisation of the following problem:

$$(v^{k+1})''(x) = f^{(1)}(x),\ v^{k+1}(a) = v(a) \text{ and } v^{k+1}(x_\Gamma) = w^k(x_\Gamma),$$

where $f^{(1)}$ is some function. If we let $\bar{f}^{(1)} = 0,\ v(a) = 0$ and $w^k(x_\Gamma) = 1$ we get the expression for $\bar{h}$:

$$A^{(1)}\bar{h} = -A_\Gamma^{(1)},$$

which is a discretisation of:

$$h''(x) = 0,\ h(a) = 0 \text{ and } h(x_\Gamma) = 1,$$

where $h(x) = \frac{x-a}{l_1} \in \mathcal{P}_1 \subset \mathcal{V}$. Because h^h defined by $\bar{h}$ is a Poisson-Dirichlet discretisation of h we get pointwise convergence on the boundary. Using this we attain:

$$\begin{aligned} d_\Gamma^{(1)} h^h(x_\Gamma) + d^{(1)}\bar{h} &\to d_\Gamma^{(1)} h(x_\Gamma) + d^{(1)}\left(\lim_{n_1\to\infty} \bar{h}\right) \\ &= d_\Gamma^{(1)} + d^{(1)}\left(\lim_{n_1\to\infty} \bar{h}\right) = h'(x_\Gamma) = \frac{1}{l_1},\ n_1 \to \infty. \end{aligned} \tag{3.21}$$

Next we analyze the denominator:

$$d_\Gamma^{(2)} - d^{(2)}(A^{(2)})^{-1}A_\Gamma^{(2)} = d_\Gamma^{(2)} + d^{(2)}(A^{(2)})^{-1}(-A_\Gamma^{(1)}) = d_\Gamma^{(2)} + d^{(2)}\bar{g},$$

if $A^{(2)}\bar{g} = -A_\Gamma^{(2)}$. We discretize the following differential equation with the Neumann discretisation:

$$g''(x) = 0,\ g'(x_\Gamma) = -\frac{1}{l_2},\ g(b) = 0,$$

where the exact solution is $g(x) = \frac{b-x}{l_2} \in \mathcal{P}_1 \subset \mathcal{W}$. The corresponding system defining the coefficients $\bar{g}$ of this discretisation is:

$$\begin{bmatrix} A^{(2)} & A_\Gamma^{(2)} \\ d^{(2)} & d_\Gamma^{(2)} \end{bmatrix} \begin{bmatrix} \bar{g} \\ g^h(x_\Gamma) \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{l_2} \end{bmatrix},$$

where we see that the first equation is the definition of $\bar{g}$. Because this is a convergent Poisson-Neumann discretisation we have pointwise convergence at the boundary. From this we get: $g^h(x_\Gamma) \to g(x_\Gamma) = 1,\ \ n_2 \to \infty$. Adding this information into the system yields:

$$\begin{bmatrix} A^{(2)} & A_\Gamma^{(2)} \\ d^{(2)} & d_\Gamma^{(2)} \end{bmatrix} \begin{bmatrix} \bar{g} \\ g^h(x_\Gamma) \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{l_2} \end{bmatrix} \to \begin{bmatrix} A^{(2)} & A_\Gamma^{(2)} \\ d^{(2)} & d_\Gamma^{(2)} \end{bmatrix} \begin{bmatrix} \lim_{n_2\to\infty} \bar{g} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{l_2} \end{bmatrix},\ n_2 \to \infty.$$

From the last equation of this system we get:

$$d_\Gamma^{(2)} + d^{(2)} \left(\lim_{n_2\to\infty} \bar{g} \right) = -\frac{1}{l_2}. \tag{3.22}$$

If this didn't hold we would have a Poisson-Neumann discretisation who's limit does not satisfy the Neumann boundary condition, which would be a contradiction. Combining (3.20), (3.21) and (3.22) completes the proof. □

From this theorem we get information about the asymptotic convergence rate. This could be interesting, but the general case is more interesting. To get statements of the general case, using this approach, we need to remove the limits. In the proof of Theorem 3.5.1 we use limits to make the discrete approximation, u^h, converge to the exact solution, u. If we can show that the discrete approximation is the exact solution, independent of n, we can remove all the limits in the proof. This reasoning gives the following corollary.

Corollary 3.5.1. *If the convergent Poisson-Dirichlet and Poisson-Neumann discretisations in Theorem 3.5.1 approximates a function in $\mathcal{P}_1$ exactly, for any n_1 and n_2, then the convergence rate is:*

$$\mu = \left| \frac{\lambda_1 l_2}{\lambda_2 l_1} \right|,\ \forall n_1, n_2 \in \mathbb{Z}^+.$$

Proof. Follow the proof of Theorem 3.5.1 and remove all the limits by using the extra condition. □

The second order finite element discretisations are Poisson-Dirichlet and Poisson-Neumann discretisation and have the extended property required to use Corollary 3.5.1. Instead of solving the large linear systems to get the result from Section 3.2, we can use this corollary and the properties of the finite element discretisation. We showed that the convergence rate of the finite volume discretisation is the same as the finite element discretisation in Section 3.3. This form of reasoning can be useful. If we can show that two discretisations are equivalent convergence wise and one has the properties to use Corollary 3.5.1, then the property must extend to the other discretisation. This reasoning can be used for any time-independent one-dimensional transmission problems. If you wish to show that a discretisation has a convergence property as in Corollary

(3.5.1), then check if the discretisation solves the homogenous solution exactly. If it doesn't, then you might be able to find a discretisation that has the properties and show that they have the same convergence rate.

There are issues generalizing these results. There is no direct way to generalize this to multivariable problems. In the proof we use that $\mathbb{R}$ has a unit, $\mathbb{R}^n$ does not. If we have time-dependent problems then the homogenous problem will change. This could effect the convergence rate. It might be interesting to extend parts of these results to the harder problems, considering this is a very powerful statement, but there is no easy way to do so.

Chapter 4

Conclusion

This thesis began by introducing the Poisson's equation and three different discretisation methods to find an approximation to the solution of the Poisson's equation. These approximations are a second order finite difference, a second order finite element, and a second order finite volume discretisation. We also introduced the general Dirichlet-Neumann problem, algorithm, and convergence rate. The general Dirichlet-Neumann problem requires two discretisations; one for the Dirichlet part and one for the Neumann part.

The convergence rate of the Dirichlet-Neumann algorithm when using the same discretisation for both the Dirichlet and Neumann part was analyzed first. In each of the three discretisations the convergence rate was proven and tested. Different elements of an inverse mass matrix were calculated to prove the convergence rate of each discretisation exactly. The theoretical convergence rate was tested against several numerical tests. Theses tests showed that the deviation between the expected results and the actual numerical results were negligible.

To bring the convergence rates together, a theorem was presented. This theorem gives the convergence rate of the Dirichlet-Neumann algorithm with any mix of the discretisations presented. This theorem was proven using elements of the proofs of the theorems regarding the convergence rate for non-mixed discretisations. This theorem states that the convergence rate is the same no matter which combination of the three discretisations is used. Moreover, the convergence rate is independent of the grid used. The convergence rate is only dependent on the parameters of the Dirichlet-Neumann problem. To confirm this property, numerical tests were performed for mixed discretisations. Negligible deviations compared to the theoretical results were observed. Inspired by this, a general theorem for what discretisations this results holds for was presented in the asymptotic case. A corollary for when the limits can be dropped was also provided.

The convergence rate of the three discretisations presented in this thesis was calculated exactly. Given that we find the convergence rates for the equations using a pair of discretisations exactly, one might be able to do so for time-dependent or multidimensional problems. The systems will be more complex, but they will still have structure. Generalizing the general theorem of this thesis to the multidimensional case is not straight forward. In the proof, we use that $\mathbb{R}$ has a unit whereas $\mathbb{R}^n$ does not. If there is a way around this issue then it could give general information about the convergence rate. The proof of the general theorem can be modified to handle different time-independent transmission problems. The homogenous solution will be different, but the process will be identical. Moreover, we showed that if the discretisations approximates the homogenous solution exactly then we have the result from the general theorem without the limit. In this thesis, we analyze a differential equation with a homogenous solution in $\mathcal{P}_1$. A lot of discretisations should give the exact solution, especially higher order discretisations.

Chapter 5

Appendix

A1: Test problems

Poisson's equation with Dirichlet Boundary conditions:
The following Poisson equation:

$$u''(x) = 2,\ x \in [0,1],\ u(0) = 1 \text{ and } u(1) = 1, \tag{5.1}$$

has the exact solution $u(x) = 1 - x + x^2$.

Poisson's equation with Dirichlet and Neumann Boundary conditions:
The following Poisson equation:

$$u''(x) = 2,\ x \in [0,1],\ u'(0) = 1 \text{ and } u(1) = 1, \tag{5.2}$$

has the exact solution $u(x) = -1 + x + x^2$.

Transmission problem:
The following transmission problem:

$$\begin{aligned} 0.5v''(x) = 1,\ x \in [0,1] \\ w''(x) = 2,\ x \in [1,2] \\ v(1) = w(1),\ 0.5v'(1) = w'(1),\ v(0) = 2,\ w(2) = 1, \end{aligned} \tag{5.3}$$

has the exact solution $v(x) = 2 - (8/3)x + x^2$ and $w(x) = -3 - (7/3)(x-2) + x^2$.

A2: Lemmas about the general Discretistation matrix

Lemma 5.0.1. *Let the matrix $A \in \mathbb{R}^{n \times n}$ have the form:*

$$[A]_{i,j} = \begin{cases} \frac{1}{\Delta x_i} + \frac{1}{\Delta x_{i+1}},\ i = j \\ -\frac{1}{\Delta x_{i+1}},\ i = j+1 \\ -\frac{1}{\Delta x_{i+1}},\ j = i+1 \\ 0,\ \textit{else} \end{cases}$$

where $\Delta x_i \in \mathbb{R},\ \Delta x_i > 0$ *for* $i = 1, \ldots, n+1$. *Then:*

$$e_n^T A^{-1} e_n = \Delta x_{n+1}(1 - \frac{\Delta x_{n+1}}{l}) \tag{5.4}$$

and

$$e_1^T A^{-1} e_1 = \Delta x_1(1 - \frac{\Delta x_1}{l}), \tag{5.5}$$

if

$$\sum_{i=1}^{n+1} \Delta x_i = l.$$

Proof. We will start by proving (5.4). Let $v = (v_1, \ldots, v_n)^T \in \mathbb{R}$ be defined by:

$$v = A^{-1} e_n.$$

We know that $e_n^T A^{-1} e_n = e_n^T v = v_n$, so we analyze the linear system $Av = e_n$.

$$\begin{cases} \left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2}\right) v_1 - \frac{1}{\Delta x_2} v_2 = 0 \\ -\frac{1}{\Delta x_2} v_1 + \left(\frac{1}{\Delta x_2} + \frac{1}{\Delta x_3}\right) v_2 - \frac{1}{\Delta x_3} v_3 = 0 \\ \vdots \\ -\frac{1}{\Delta x_{n-1}} v_{n-2} + \left(\frac{1}{\Delta x_{n-1}} + \frac{1}{\Delta x_n}\right) v_{n-1} - \frac{1}{\Delta x_n} v_n = 0 \\ -\frac{1}{\Delta x_n} v_{n-1} + \left(\frac{1}{\Delta x_n} + \frac{1}{\Delta x_{n+1}}\right) v_n = 1 \end{cases} \tag{5.6}$$

Remark 5.0.1.

$$(5.6) \Rightarrow \left(\frac{1}{\sum_{i=1}^{k} \Delta x_i} + \frac{1}{\Delta x_{k+1}}\right) v_k - \frac{1}{\Delta x_{k+1}} v_{k+1} = 0,\ k = 1, \ldots, n-1.$$

Proof. We will prove this by induction. The base case ($k = 1$)

$$\left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2}\right) v_1 - \frac{1}{\Delta x_2} v_2 = 0$$

is the first equation in (5.6). Next we prove the induction step. Assume that:

$$\left(\frac{1}{\sum_{i=1}^{k} \Delta x_i} + \frac{1}{\Delta x_{k+1}}\right) v_k - \frac{1}{\Delta x_{k+1}} v_{k+1} = 0,\ 1 \le k \le n-2 \tag{5.7}$$

then prove that:

$$\left(\frac{1}{\sum_{i=1}^{k+1} \Delta x_i} + \frac{1}{\Delta x_{k+2}}\right) v_{k+1} - \frac{1}{\Delta x_{k+2}} v_{k+2} = 0. \tag{5.8}$$

First we work with the induction assumption (5.7).

$$\left(\frac{1}{\sum_{i=1}^{k} \Delta x_i} + \frac{1}{\Delta x_{k+1}}\right) v_k - \frac{1}{\Delta x_{k+1}} v_{k+1} = 0$$

$$\Leftrightarrow \left(\frac{1}{l_k} + \frac{1}{\Delta x_{k+1}}\right) v_k - \frac{1}{\Delta x_{k+1}} v_{k+1} = 0,\ l_k = \sum_{i=1}^{k} \Delta x_i$$

$$\Leftrightarrow \left(\frac{l_k + \Delta x_{k+1}}{l_k \Delta x_{k+1}}\right) v_k - \frac{1}{\Delta x_{k+1}} v_{k+1} = 0. \tag{5.9}$$

Now we use the $(k+1)$st equation from (5.6). We know that it will never be the first or last equation. This is because $1 < k+1 < n$. Therefore our equation is:

$$-\frac{1}{\Delta x_{k+1}} v_k + \left(\frac{1}{\Delta x_{k+1}} + \frac{1}{\Delta x_{k+2}}\right) v_{k+1} - \frac{1}{\Delta x_{k+2}} v_{k+2} = 0. \tag{5.10}$$

With this we add (5.9) multiplied by $\frac{l_k}{l_k + \Delta x_{k+1}}$ to (5.10). This results in:

$$\left(-\frac{l_k}{\Delta x_{k+1}(l_k + \Delta x_{k+1})} + \frac{1}{\Delta x_{k+1}} + \frac{1}{\Delta x_{k+2}}\right) v_{k+1} - \frac{1}{\Delta x_{k+2}} v_{k+2} = 0$$

$$\Leftrightarrow \left(-\frac{l_k}{\Delta x_{k+1}(l_k + \Delta x_{k+1})} + \frac{l_k + \Delta x_{k+1}}{\Delta x_{k+1}(l_k + \Delta x_{k+1})} + \frac{1}{\Delta x_{k+2}}\right) v_{k+1} - \frac{1}{\Delta x_{k+2}} v_{k+2} = 0$$

$$\Leftrightarrow \left(\frac{\Delta x_{k+1}}{\Delta x_{k+1}(l_k + \Delta x_{k+1})} + \frac{1}{\Delta x_{k+2}}\right) v_{k+1} - \frac{1}{\Delta x_{k+2}} v_{k+2} = 0$$

$$\Leftrightarrow \left(\frac{1}{l_k + \Delta x_{k+1}} + \frac{1}{\Delta x_{k+2}}\right) v_{k+1} - \frac{1}{\Delta x_{k+2}} v_{k+2} = 0$$

$$\Leftrightarrow \left(\frac{1}{\sum_{i=1}^{k+1} \Delta x_i} + \frac{1}{\Delta x_{k+2}}\right) v_{k+1} - \frac{1}{\Delta x_{k+2}} v_{k+2} = 0.$$

Because we have shown that the remark holds for $k = 1$ and proved the induction step (5.7) $\Rightarrow$ (5.8) for $1 < k \leq n-2$ we have proved the remark. $\square$

Using Remark 5.0.1 when $k = n-1$ we know that:

$$\left(\frac{1}{l_{n-1}} + \frac{1}{\Delta x_n}\right) v_{n-1} - \frac{1}{\Delta x_n} v_n = 0,\ l_{n-1} = \sum_{i=1}^{n-1} \Delta x_i \tag{5.11}$$

$$\Leftrightarrow \left(\frac{l_{n-1} + \Delta x_n}{l_{n-1} \Delta x_n}\right) v_{n-1} - \frac{1}{\Delta x_n} v_n = 0.$$

We multiply this with $\frac{l_{n-1}}{l_{n-1} + \Delta x_n}$ and add it to the last equation in (5.6). The result is:

$$\left(-\frac{l_{n-1}}{\Delta x_n (l_{n-1} + \Delta x_n)} + \frac{1}{\Delta x_n} + \frac{1}{\Delta x_{n+1}}\right) v_n = 1. \tag{5.12}$$

To prove (5.4) holds we continue to work with (5.12).

$$\left(-\frac{l_{n-1}}{\Delta x_n (l_{n-1} + \Delta x_n)} + \frac{l_{n-1} + \Delta x_n}{\Delta x_n (l_{n-1} + \Delta x_n)} + \frac{1}{\Delta x_{n+1}}\right) v_n = 1$$

$$\Leftrightarrow \left(\frac{\Delta x_n}{\Delta x_n (l_{n-1} + \Delta x_n)} + \frac{1}{\Delta x_{n+1}}\right) v_n = 1$$

$$\Leftrightarrow \left(\frac{1}{(l_{n-1} + \Delta x_n)} + \frac{1}{\Delta x_{n+1}}\right) v_n = 1$$

$$\left(\frac{1}{(\sum_{i=1}^{n} \Delta x_i)} + \frac{1}{\Delta x_{n+1}}\right) v_n = 1.$$

Multiplying both sides with Δx_{n+1} gives:

$$\left(\frac{\Delta x_{n+1}}{\sum_{i=1}^{n} \Delta x_i} + 1\right) v_n = \left(\frac{\Delta x_{n+1} + \sum_{i=1}^{n} \Delta x_i}{\sum_{i=1}^{n} \Delta x_i}\right) v_n = \Delta x_{n+1}.$$

Next we use the definition of l:

$$\left(\frac{\sum_{i=1}^{n+1} \Delta x_i}{\sum_{i=1}^{n} \Delta x_i}\right) v_n = \left(\frac{l}{l - \Delta x_{n+1}}\right) v_n = \Delta x_{n+1}.$$

With this we have finished the proof of Equation 5.4 because:

$$e_n^T A^{-1} e_n = e_n^T v = v_n = \Delta x_{n+1} \left(\frac{l - \Delta x_{n+1}}{l}\right) = \Delta x_{n+1} \left(1 - \frac{\Delta x_{n+1}}{l}\right)$$

Next we prove (5.5). Let $w = (w_1, \ldots, w_n)^T \in \mathbb{R}^n$ be defined as:

$$Aw = e_1. \tag{5.13}$$

We know that $e_1^T A^{-1} e_1 = e_1 w = w_1$, so we write out the system $Aw = e_1$.

$$\begin{cases} \left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2}\right) w_1 - \frac{1}{\Delta x_2} w_2 = 1 \\ -\frac{1}{\Delta x_2} w_1 + \left(\frac{1}{\Delta x_2} + \frac{1}{\Delta x_3}\right) w_2 - \frac{1}{\Delta x_3} w_3 = 0 \\ \vdots \\ -\frac{1}{\Delta x_{n-1}} w_{n-2} + \left(\frac{1}{\Delta x_{n-1}} + \frac{1}{\Delta x_n}\right) w_{n-1} - \frac{1}{\Delta x_n} w_n = 0 \\ -\frac{1}{\Delta x_n} w_{n-1} + \left(\frac{1}{\Delta x_n} + \frac{1}{\Delta x_{n+1}}\right) w_n = 0 \end{cases} \tag{5.14}$$

Now we define the vector $\hat{w} = (\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n) = (w_n, w_{n-1}, \ldots, w_1)$ and substitute w with $\hat{w}$ in (5.14).

$$\begin{cases} \left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2}\right) \hat{w}_n - \frac{1}{\Delta x_2} \hat{w}_{n-1} = 1 \\ -\frac{1}{\Delta x_2} \hat{w}_n + \left(\frac{1}{\Delta x_2} + \frac{1}{\Delta x_3}\right) \hat{w}_{n-1} - \frac{1}{\Delta x_3} \hat{w}_{n-2} = 0 \\ \vdots \\ -\frac{1}{\Delta x_{n-1}} \hat{w}_3 + \left(\frac{1}{\Delta x_{n-1}} + \frac{1}{\Delta x_n}\right) \hat{w}_2 - \frac{1}{\Delta x_n} \hat{w}_1 = 0 \\ -\frac{1}{\Delta x_n} \hat{w}_2 + \left(\frac{1}{\Delta x_n} + \frac{1}{\Delta x_{n+1}}\right) \hat{w}_1 = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \left(\frac{1}{\Delta x_n} + \frac{1}{\Delta x_{n+1}}\right) \hat{w}_1 - \frac{1}{\Delta x_n} \hat{w}_2 = 0 \\ -\frac{1}{\Delta x_n} \hat{w}_1 + \left(\frac{1}{\Delta x_{n-1}} + \frac{1}{\Delta x_n}\right) \hat{w}_2 - \frac{1}{\Delta x_{n-1}} \hat{w}_3 = 0 \\ \vdots \\ -\frac{1}{\Delta x_3} \hat{w}_{n-2} + \left(\frac{1}{\Delta x_2} + \frac{1}{\Delta x_3}\right) \hat{w}_{n-1} - \frac{1}{\Delta x_2} \hat{w}_n = 0 \\ -\frac{1}{\Delta x_2} \hat{w}_{n-1} \left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2}\right) \hat{w}_n = 1 \end{cases} . \tag{5.15}$$

Here we introduce $\Delta\hat{x}_i = \Delta x_{n-i+2},\ i = 1,\dots,n+1$ and substitute Δx with $\Delta\hat{x}$ in (5.15) to get (5.16).

$$\begin{cases} \left(\frac{1}{\Delta\hat{x}_1} + \frac{1}{\Delta\hat{x}_2}\right)\hat{w}_1 - \frac{1}{\Delta\hat{x}_2}\hat{w}_2 = 0 \\ -\frac{1}{\Delta\hat{x}_2}\hat{w}_1 + \left(\frac{1}{\Delta\hat{x}_2} + \frac{1}{\Delta\hat{x}_3}\right)\hat{w}_2 - \frac{1}{\Delta\hat{x}_3}\hat{w}_3 = 0 \\ \vdots \\ -\frac{1}{\Delta\hat{x}_{n-1}}\hat{w}_{n-2} + \left(\frac{1}{\Delta\hat{x}_{n-1}} + \frac{1}{\Delta\hat{x}_n}\right)\hat{w}_{n-1} - \frac{1}{\Delta\hat{x}_n}\hat{w}_n = 0 \\ -\frac{1}{\Delta\hat{x}_n}\hat{w}_{n-1} + \left(\frac{1}{\Delta\hat{x}_n} + \frac{1}{\Delta\hat{x}_{n+1}}\right)\hat{w}_n = 1 \end{cases}. \tag{5.16}$$

This has the same structure as (5.6). We know that:

$$\hat{w}_n = \Delta\hat{x}_{n+1}\left(1 - \frac{\Delta\hat{x}_{n+1}}{l}\right),\ \sum_{i=1}^{n+1}\Delta\hat{x}_i = \sum_{i=1}^{n+1}\Delta x_{n+2-i} = l.$$

Reverting back to w and Δx will result in:

$$e_1^T A^{-1} e_1 = e_1^T w = w_1 = \hat{w}_n = \Delta x_1\left(1 - \frac{\Delta x_1}{l}\right).$$

With that we have proven both (5.4) and (5.5). □

Lemma 5.0.2. *Let the matrix $A \in \mathbb{R}^{n\times n}$ have the form:*

$$[A]_{i,j} = \begin{cases} \frac{1}{\Delta x_i} + \frac{1}{\Delta x_{i+1}},\ i = j \\ -\frac{1}{\Delta x_{i+1}},\ i = j+1 \\ -\frac{1}{\Delta x_{i+1}},\ j = i+1 \\ 0,\ else \end{cases}$$

where $\Delta x_i \in \mathbb{R},\ \Delta x_i > 0$ for $i = 1,\dots,n+1$. Then:

$$e_2^T A^{-1} e_1 = \Delta x_1\left(1 - \frac{\Delta x_1}{l} - \frac{\Delta x_2}{l}\right) \tag{5.17}$$

and

$$e_{n-1}^T A^{-1} e_n = \Delta x_{n+1}\left(1 - \frac{\Delta x_{n+1}}{l} - \frac{\Delta x_n}{l}\right) \tag{5.18}$$

if

$$\sum_{i=1}^{n+1}\Delta x_i = l.$$

Proof. We start by proving (5.18). The A matrix has the same shape as the A matrix in Lemma 5.0.1. Therefore we can use Remark 5.0.1 to get:

$$\left(\frac{1}{\sum_{i=1}^{n-1}\Delta x_i} + \frac{1}{\Delta x_n}\right)v_{n-1} - \frac{1}{\Delta x_n}v_n = 0,$$

where $v = Ae_n$, $v_{n-1} = e_{n-1}^T A e_n$ and $v_n = e_n^T A e_n$. We know what v_n is from Remark **??**, so we get:

$$\left(\frac{1}{l - \Delta x_n - \Delta x_{n+1}} + \frac{1}{\Delta x_n}\right)v_{n-1} - \frac{\Delta x_{n+1}}{\Delta x_n}\left(1 - \frac{\Delta x_{n+1}}{l}\right) = 0$$

$$\Leftrightarrow \left(\frac{1}{l-\Delta x_n-\Delta x_{n+1}}+\frac{1}{\Delta x_n}\right)v_{n-1}=\frac{\Delta x_{n+1}}{\Delta x_n}\left(1-\frac{\Delta x_{n+1}}{l}\right).$$

Next we just continue to simplify until we have shown (5.18).

$$\left(\frac{1}{l-\Delta x_n-\Delta x_{n+1}}+\frac{1}{\Delta x_n}\right)v_{n-1}=\frac{\Delta x_{n+1}}{\Delta x_n}\left(1-\frac{\Delta x_{n+1}}{l}\right)$$

$$\Leftrightarrow \left(\frac{\Delta x_n}{l-\Delta x_n-\Delta x_{n+1}}+1\right)v_{n-1}=\Delta x_{n+1}\left(1-\frac{\Delta x_{n+1}}{l}\right)$$

$$\Leftrightarrow (l-\Delta x_{n+1})\,v_{n-1}=(l-\Delta x_n-\Delta x_{n+1})\Delta x_{n+1}\left(1-\frac{\Delta x_{n+1}}{l}\right)$$

$$\Leftrightarrow v_{n-1}=\frac{(l-\Delta x_n-\Delta x_{n+1})\Delta x_{n+1}\left(1-\frac{\Delta x_{n+1}}{l}\right)}{l-\Delta x_{n+1}}$$

$$\Leftrightarrow v_{n-1}=\frac{(l-\Delta x_n-\Delta x_{n+1})\Delta x_{n+1}\,(l-\Delta x_{n+1})}{l(l-\Delta x_{n+1})}$$

$$\Leftrightarrow v_{n-1}=\Delta x_{n+1}\frac{l-\Delta x_n-\Delta x_{n+1}}{l}\Leftrightarrow (5.18).$$

Using the same process as the second part of the proof for Remark **??**, we find $e_2^T A^{-1}e_1$ by first solving (5.13). Then we get $e_2^T w = e_2^T A^{-1}e_1 \Rightarrow w_2 = e_2^T A^{-1}e_1$. With this we introduce $\hat{w}=(\hat{w}_1\ldots,\hat{w}_n)=(w_n,\ldots,w_1)$ and $\Delta\hat{x}_i = \Delta x_{n-i+2},\ i=1\ldots n+1$. Using these new variables we can construct an equivalent system (5.16). We want to find an expression for $w_2=\hat{w}_{n-1}$. We know from the first part of this proof that:

$$\hat{w}_{n-1}=e_{n-1}^T\hat{A}^{-1}e_n=\Delta\hat{x}_{n+1}\left(1-\frac{\Delta\hat{x}_{n+1}}{l}-\frac{\Delta\hat{x}_n}{l}\right),$$

where $\hat{A}$ has the same form as A and $\hat{A}\hat{w}=e_n$ is presented in (5.16). Changing back to Δx and w finishes the proof:

$$w_2=\hat{w}_{n-1}=\Delta x_1\left(1-\frac{\Delta x_1}{l}-\frac{\Delta x_2}{l}\right),$$

□

Bibliography

[1] Zhichun Yang, Shun He, Yingsong Gu. *Transonic limit cycle oscillation behavior of an aeroelastic airfoil with free-play.* Journal of Fluids and Structures, 66:1-186, 2016.

[2] Christine M. Scotti, Ender A. Finol. *Compliant biomechanics of abdominal aortic aneurysms: A fluid–structure interaction study.* Computers & Structures, 85:1097-1113, 2007.

[3] D. Kowollik, V. Tini, S. Reese, M. Haupt. *3D fluid-structure interaction analysis of a typical liquid rocket engine cycle based on a novel viscoplastic damage model.* International Journal for Numerical Methods in Engineering, 94:1165-1190, 2013.

[4] Azahar Monge, Philipp Birken. *On the convergence rate of the Dirichlet-Neumann iteration for unsteady thermal fluid structure interaction.* Computational Mechanics, 62(3):525-541, 2018.

[5] Azahar Monge. *Partitioned methods for time-dependent thermal fluid-structure interaction.* Phd Thesis, Lund University, Sweden, 2018.

[6] Thomas J.R. Hughes *The Finite Element Method Linear Static and Dynamic Finite Element Analysis.* Dover Publications, New York, 2000.

[7] James M. Ortega, William G. Poole Jr *An Introduction to Numerical Methods for Differential Equations.* Pitman Publishing, Massachusetts, 1981.

[8] Randall J. Leveque *Finite Volume methods for Hyperbolic Problems.* University of Cambridge, Cambridge, 2002.

[9] Fuzhen Zhang *The Schur Complement and Its Applications.* Springer, United States of America, 2005.

[10] L. Fox *An introduction to Numerical Linear Algebra.* Oxford, New York, 1965.

[11] A. Toselli, O.Widlund *Domain Decomposition Methods-Algorithms and Theory.* Springer, Berlin, 2010.

Master's Theses in Mathematical Sciences 2019:E21
ISSN 1404-6342

LUNFNA-3028-2019

Numerical Analysis
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/