Lund University

Master's Thesis

The Dirichlet-Neumann iteration

Three-field case: Methods and analyses

Author:
Nils Ivo Dravins

Supervisor:
Philipp Birken
Assistant supervisor:
Azahar Monge

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

Division of Numerical Analysis
Centre for Mathematical Sciences

March 2018

Simplicity is the final achievement. After one has played a vast quantity of notes and more notes, it is simplicity that emerges as the crowning reward of art.

Frédéric Chopin

* * *

In gratitude to Philipp and Azahar for their capable guidance
and to my friends & family for standing by me.

LUND UNIVERSITY

Abstract

Division of Numerical Analysis
Centre for Mathematical Sciences

Master of Science

The Dirichlet-Neumann iteration.
Three-field case: Methods and analyses

by Nils Ivo Dravins

We construct and analyze Dirichlet-Neumann iterations for the 1D Poisson equation. Specifically, we wish to gain insight into how the convergence depends on material coefficients when solving coupled linear heat equations on three non-overlapping domains. We first consider the two-domain case and then extend the method to three domains. A finite element method is used to discretize the Laplacian. Exact formulae are provided for the spectral radii of the iteration matrices for all methods considered. Their validity as predictors for the convergence rates is verified through numerical tests. We show that the different methods for the three-field case have distinct and complementary convergence properties and give an overview of problems, specifying which method is the most suitable.

Contents

Chapter 1

Introduction and motivation

1.1 Introduction

Numerical methods have, since their inception, been applied in support of development and design of numerous productive industries. The ability to simulate and predict behavior and properties without the expense of physical prototypes and tests offers a huge advantage with benefits in cost, time and quality of the end product. The exact area of application may vary, examples include structural stress, aerodynamic drag and distribution of heat.

1.2 Motivation

This work will deal specifically with algorithms intended for use in simulating heat distribution of materials with different heat properties while being in contact with each other. The specific problem that initiated this search for efficient algorithms was one raised by the introduction of reusable launch vehicles, in particular the first stage of the Falcon 9 developed by SpaceX [1]. The demands placed by reusability require a good understanding of the structural and heat properties of the rocket engine and also pose fundamental questions such as "for how many launches can the engine nozzle be expected to retain structural integrity before it would require replacement or repair?"

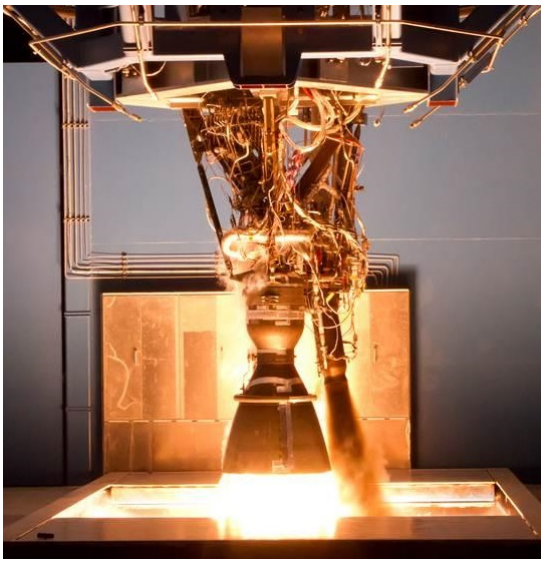

FIGURE 1.1: Engine test of a Merlin rocket engine, used by the company SpaceX [F1]

In this example, the engine nozzle is subject to massive forces and the heat distribution will depend on the interaction of several materials with different thermal properties. To name some: the composite engine nozzle is cooled by a cooling fluid led through pipes in its interior while the same composite will be subject to the fuel being ignited in the combustion chamber and then blasted out through the nozzle. The nozzle itself will be interacting with the surrounding air before it leaves the atmosphere.

While this was the motivating problem for this work, it is not hard to imagine other possible applications. One could be the brake system on a high-performance car. Here friction caused by a hard brake will cause immense heat which will distribute among the different components of the brake system and adjoining parts such as the wheels [2].

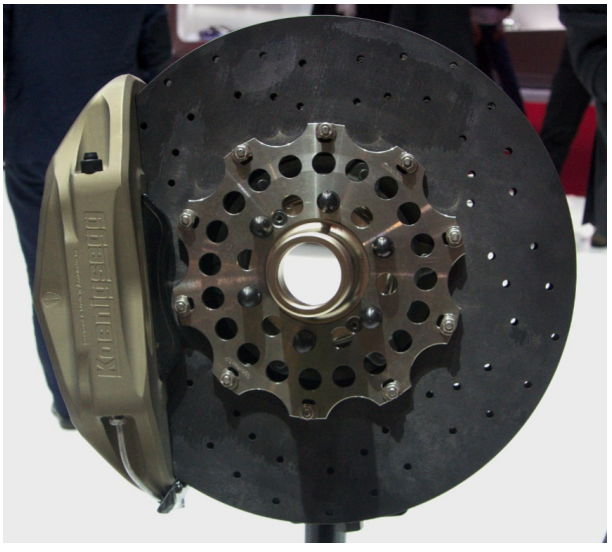

FIGURE 1.2: Disc brake of a high-performance car [F2]

Another possible application would be in simulating a refrigerator (or a heat pump) as this can also be modeled as a three-field problem.

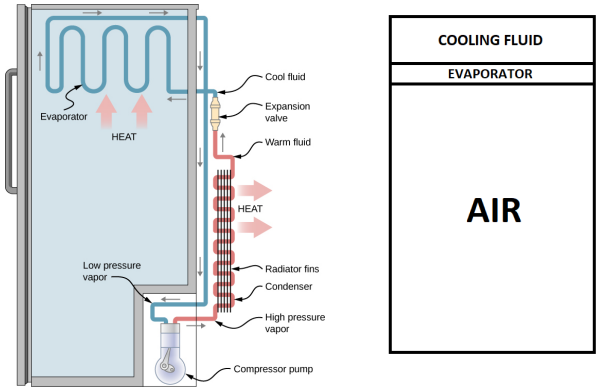

FIGURE 1.3: Refrigerator schematic [F3] and a simple three-field model.

Essentially, we wish to create algorithms that take advantage of the various thermal properties of different materials interacting with each other.

1.3 Method

The first decision when creating an algorithm to simulate the heat of a combination of materials is whether to use a monolithic approach in which the code is tailored to the problem specifically. The alternative is a partitioned approach in which the heat equation is solved independently for each material. Here the different solvers are being coordinated by a mother program and communicating by selected data transfers that are sending boundary temperature and gradients between the solvers. The partitioned method offers modular advantages allowing the solver to be quickly adapted for new problems as well as in coupling different materials such as in the numerical simulation of fluid-structure interaction [3]. A partitioned approach also offers the attractive advantage of reusing existing solvers for new problems. This work will study a partitioned approach.

Another decision must be whether the algorithm is to be a direct solver, solving the problem directly to machine accuracy or if an iterative method is to be used. In high dimensional systems the computational cost of a direct solver may become very high while an iterative method can deliver an approximate solution given a set tolerance but also offers challenges in predicting and guaranteeing the speed of convergence. In this work we will primarily study iterative methods. One of the basic methods of managing the coupling is the Dirichlet-Neumann iteration [3]. To satisfy the coupling conditions at the interface, Dirichlet- and Neumann data are sent between the sub-solvers in each iteration in a sequential manner. There have been several studies examining convergence

behavior of different implementations of the Dirichlet-Neumann iteration [4][5][6][7][8]. The author, however, is not aware of any study which presents an explicit analysis for the three field case. Our hope is that by focusing on the one-dimensional case, we will be able to gain explicit analytical predictions for the convergence behavior that will give us some predictive power in higher dimensions as well.

1.4 Overview

This work studies the convergence properties of variants of an iterative partitioned Dirichlet-Neumann iteration derived from the heat equation. We will be using a finite element method to discretize all sub-domains.

We begin by looking at the two-domain problem to confirm results from an earlier work and then move on to construct three different three-domain solvers. For all algorithms considered, we perform a convergence analysis by analytically reducing the full solver into a fixed-point iteration acting on the interfaces alone. This gives us an iteration matrix whose spectral radius determines the speed of convergence. This allows us to predict convergence rates given material coefficients.

It is worth noting that while we only consider the steady-state case, earlier analysis [3] shows that the convergence behavior with a time discretization added is dominated by the convergence rate predicted by material properties. The methods presented in this thesis are thus likely not limited to the steady state case but extend directly to the time-dependent case. Furthermore it has been shown [3] that in the two-field case, the asymptotic convergence rates of the 2D case are consistent with the 1D case. There are therefore grounds for hope that the same will hold true in the three-field case.

1.4.1 Organization

Chapter 2 introduces the mathematical background followed by the finite element discretization. In Chapter 3, the two field case is presented together with a convergence analysis of the iterative solver. In Chapter 4 we construct three iterative solvers for the three field case and present a convergence analysis for each. In Chapter 5 the predictions made in Chapters 3 and 4 are tested in numerical experiments and we give some examples of real-world applications. In Chapter 6 we present the conclusions of the work together with some recommendations for further study.

Chapter 2

Theoretical background

2.1 Linear fixed point iteration

The analysis in this work will be based on extracting a fixed point iteration from the Dirichlet-Neumann iteration. Specifically, we will formulate a Dirichlet-Neumann iteration that aims to solve a problem on two or three sub-domains. From this we will extract a linear fixed point iteration acting only on the boundary points separating the domains. This will allow us to use basic properties of linear fixed point iterations to predict the convergence of the underlying Dirichlet-Neumann iteration.

Given a (finite dimensional) linear system $A\boldsymbol{x} = \boldsymbol{b}$ where $A \in \mathbb{R}^{n\times n}$ and $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$, we may rewrite it in fixed point form as $\boldsymbol{x} = B\boldsymbol{x} + \boldsymbol{b}$, where $B = I - A$, is the iteration matrix. The fixed point iteration then becomes, starting with some initial guess $\boldsymbol{x}_0$;

$$\boldsymbol{x}_{n+1} = B\boldsymbol{x}_n + \boldsymbol{b}. \tag{2.1}$$

Here $\boldsymbol{x}_n$ is the nth approximation of the unique fixed point $\boldsymbol{x}_*$ satisfying $\boldsymbol{x}_* = B\boldsymbol{x}_* + \boldsymbol{b}$. Then we have the following theorem [9] :

Theorem 2.1. *For the (linear, finite dimensional) fixed point iteration* (2.1) *the following holds*

$$\boldsymbol{x}_n \to \boldsymbol{x}_*, \ \forall \boldsymbol{x}_0 \iff \rho(B) < 1$$

where $\rho(B) = max\{\ |\sigma|\ :\ \sigma$ *is eigenvalue of* $B\ \}$ *is the spectral radius.*

Furthermore, we have the following theorem about the rate of convergence [9] :

Theorem 2.2. *If* $\rho(B) < 1$ *then, for any* $\boldsymbol{x}_0$*, the iterates of the linear fixed point iteration* (2.1) *satisfy*

$$\limsup_{n\to\infty} ||\boldsymbol{x}_n - \boldsymbol{x}_*||^{\frac{1}{n}} \leq \rho(B)$$

so, if $0 < \rho(B) < 1$*, the linear fixed point iteration converges linearly with a root-convergence factor of at most* $\rho(B)$*.*

2.2 Boundary conditions

When solving differential equations on well-defined regions, one needs to specify the boundary conditions that are valid along the boundaries of those regions. In this work we will be dealing with two different boundary conditions, the Dirichlet boundary condition and the Neumann boundary condition. The Dirichlet condition gives the value of the function in a certain point. The Neumann condition gives the value of the normal derivative [10].

2.3 Dirichlet-Neumann iteration

The Dirichlet-Neumann iteration is a basic method in both domain decomposition and fluid-structure interaction [3]. Its working principle is dividing a domain into sub-domains and then solving a differential equation in each separately with alternating boundary conditions. A simple example is as follows: consider that we wish to solve some differential equation in $\Omega \subset \mathbb{R}$ with given Dirichlet boundary conditions. Ω is divided into two sub-domains $\Omega_1 \cup \Omega_2 = \Omega$. We denote the interface $\Gamma = \Omega_1 \cap \Omega_2$. Given an initial guess for the solution value on the interface u_Γ^0, we construct the iteration in Fig. 2.1:

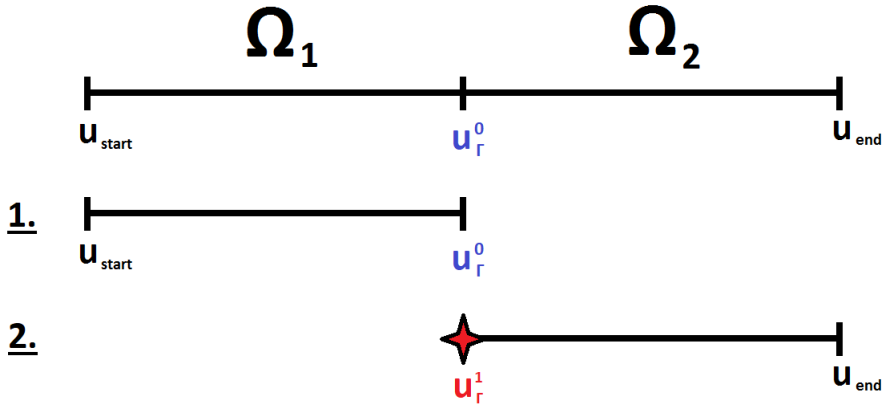

FIGURE 2.1: Example of one step of a Dirichlet-Neumann iteration Here each bar denotes a Dirichlet condition and the star denotes a Neumann condition.

Each phase of this iteration has two steps. First we solve the equation in Ω_1 with Dirichlet boundary conditions on both sides. Then, using the values obtained when

solving Ω_1, we construct a Neumann condition on the left side in Ω_2. With this we find a new value for the solution on the boundary u_Γ^1 as we solve the equation in Ω_2.

For an increasing number of domains, successively more variations become possible in how the Dirichlet and Neumann conditions can be ordered. In this work, we will begin with the two field case as presented in Fig. 2.1 and then move on to construct and investigate methods for the three field case.

2.4 Why the 1D Poisson's equation?

We wish to examine the convergence behavior for an algorithm designed to solve the heat equation for materials with different material properties. We begin by looking at the general form of the unsteady transmission problem [3]. We consider a domain $\Omega \subset \mathbb{R}^d$ which is divided into two sub-domains $\Omega_1 \cup \Omega_2 = \Omega$, with transmission conditions at the interface $\Gamma = \Omega_1 \cap \Omega_2$.

$$\begin{cases} \alpha_m \frac{\partial u_m(\boldsymbol{x},t)}{\partial t} - \nabla(\lambda_m \nabla u_m(\boldsymbol{x},t)) = 0 \ ; \ t \in [t_{start}, t_{end}], \boldsymbol{x} \in \Omega_m \subset \mathbb{R}^d \ ; \ m = 1,2 \\ u_m(\boldsymbol{x},t) = 0 \ ; \ t \in [t_{start}, t_{end}], u_m(\boldsymbol{x},t) \in \partial\Omega_m \backslash \Gamma \\ u_1(\boldsymbol{x},t) = u_2(\boldsymbol{x},t) \ ; \ \boldsymbol{x} \in \Gamma \\ \lambda_1 \frac{\partial u_1(\boldsymbol{x},t)}{\partial \boldsymbol{n_1}} = \lambda_2 \frac{\partial u_2(\boldsymbol{x},t)}{\partial \boldsymbol{n_2}} \ ; \ \boldsymbol{x} \in \Gamma \\ u_m(\boldsymbol{x},0) = u_m^0(\boldsymbol{x}) \ ; \ \boldsymbol{x} \in \Omega_m \end{cases} \tag{2.2}$$

where $\boldsymbol{n_m}$ is the outward normal to Ω_m. The constants λ_1 and λ_2 denote the thermal conductivities of the materials of Ω_1 and Ω_2. The constants α_1 and α_2 are given by $\alpha_m = \rho_m C_m$ where ρ_m is the density and C_m the heat capacity of the materials in Ω_1 and Ω_2. This transmission problem corresponds to solving two coupled heat equations, where $u(\boldsymbol{x},t)$ denotes the temperature.

As we wish to study in particular the dependence on the material coefficients λ and to find explicit predictions for the rate of convergence, we keep things simple and look at the 1D steady-state case. Thus we consider the case where $d = 1$ and $\frac{\partial u_m(\boldsymbol{x},t)}{\partial t} = 0$. The first equation of (2.2) then becomes

$$-\lambda_m \Delta u_m(x) = 0 \ ; \ x \in \Omega_m \subset \mathbb{R} \ ; \ m = 1,2. \tag{2.3}$$

These are Laplace equations. The fact that we will concentrate on the dependence of the material properties represented by λ, makes the study of the Laplace equation alone somewhat troublesome. This is because, in (2.3), we can divide both sides with λ_m thus

eliminating it from this equation. In order to force a dependence we move onto the Poisson equation. The right-hand-side f can physically be interpreted as a heat source or sink within the material. We then wish to solve

$$\begin{cases} -\lambda_m \Delta u_m(x) = f_m \ ; \ x \in \Omega_m \subset \mathbb{R} \ ; \ m = 1, 2 \\ u_m(x) = 0 \ ; \ u_m(x) \in \partial\Omega_m \backslash \Gamma \\ u_1(x) = u_2(x) \ ; \ x \in \Gamma \\ \lambda_1 \frac{\partial u_1(x)}{\partial n_1} = \lambda_2 \frac{\partial u_2(x)}{\partial n_2} \ ; \ x \in \Gamma. \end{cases} \tag{2.4}$$

Here the first equation is the governing equation, the second and third gives us Dirichlet conditions and the fourth gives us a Neumann condition. Numerically, for a non-zero f, this will correspond to solving the general heat equation for a non-zero $\frac{\partial u}{\partial t}$ term.

2.5 Theoretical derivation

We use a finite element method for the discretization. Suppose we have a domain $\Omega \in \mathbb{R}$ that is divided into two sub-domains $\Omega = \Omega_1 \cup \Omega_2$. The sub-domains Ω_1 and Ω_2 are joined at an interface $\Gamma = \Omega_1 \cap \Omega_2$. We use the equations from (2.4), futhermore we restrict ourselves to the case where $f\big|_\Gamma = 0$.

Given a uniform grid $\{x_1, x_2, ..., x_n\}$, define test functions:

$$\phi_k(x) = \begin{cases} \frac{x - x_{k-1}}{\Delta x}; x \in (x_{k-1}, x_k] \\ \frac{x_{k+1} - x}{\Delta x}; x \in (x_k, x_{k+1}] \\ 0 \quad ; \quad else. \end{cases}$$

2.5.1 Interior points

We approximate the solution by a sum of test functions;

$$u \approx \sum_i \hat{u}_i \phi_i. \tag{2.5}$$

Then we insert this expression into the governing equation from (2.4), $\Delta u = -f$, and multiply both sides with a test function ϕ_j, and then integrate both sides. Here we set $\lambda = 1$ as this constant has no bearing on the derivation:

$$\int_I u_{xx} \phi_j dx = -\int_I f \phi_j dx. \tag{2.6}$$

We now apply partial integration:

$$\int_I (u)_{xx}\phi_j dx = (u)_x\phi_j\Big|_I - \int_I (u)_x(\phi_i)_x dx \tag{2.7}$$

and note that the first term disappears as all test functions have compact support. We need thus only consider the integral. Inserting the sum;

$$\int_I (u)_x(\phi_i)_x dx = \int_I (\sum_i \hat{u}_i\phi_i)_x(\phi_j)_x dx = \sum_i \hat{u}_i \int_I (\phi_i)_x(\phi_j)_x dx. \tag{2.8}$$

We look closer at the last integral: by construction of the test functions it follows that when i, j differ by more than one, the resulting integral is zero. When i=j:

$$\int_I (\phi_i)_x(\phi_i)_x dx = \frac{1}{\Delta x^2}\int_0^{2\Delta x} 1 dx = \frac{2}{\Delta x} \tag{2.9}$$

when i and j differ by one:

$$\int_I (\phi_i)_x(\phi_j)_x dx = \frac{1}{\Delta x^2}\int_0^{\Delta x} -1 dx = \frac{-1}{\Delta x}. \tag{2.10}$$

2.5.2 Boundary

Inserting the sum we get, for the left hand side (LHS):

$$\int_I u_{xx}\phi_j dx = \int_I (\sum_i \hat{u}_i\phi_i)_{xx}\phi_j dx = \sum_i \hat{u}_i \int_I (\phi_i)_{xx}\phi_j dx. \tag{2.11}$$

We now rewrite the last integral using partial integration.

Here we take note of the coupling condition from (2.4):

$$\lambda_1 \frac{\partial u_1}{\partial n_1}\Big|_\Gamma = -\lambda_2 \frac{\partial u_2}{\partial n_2}\Big|_\Gamma. \tag{2.12}$$

Next, we note that the active test functions along the boundary will be ϕ_1^n, ϕ_Γ and ϕ_2^1.

In respective cells Ω_1^n, Ω_2^1 we thus have:

In Ω_1^n:, $u_1 = \hat{u}_1^n\phi_1^n + \hat{u}_\Gamma\phi_\Gamma$ In Ω_2^1:, $u_2 = \hat{u}_\Gamma\phi_\Gamma + \hat{u}_2^1\phi_2^1$

First we note that, if ϕ_j is a nodal basis function for a node on Γ we can rewrite the fourth equation of (2.4) using Green's formula [3]

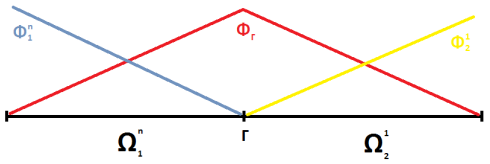

FIGURE 2.2: Boundary point with surrounding test functions

$$\int_{\partial\Omega_i} \frac{\partial u_i}{\partial n_i}\phi_j ds = \int_{\Omega_i} (\Delta u_i \phi_j + \nabla u_i \nabla \phi_j) dx = \int_{\Omega_i} (-f\phi_j + \nabla u_i \nabla \phi_j) dx. \tag{2.13}$$

Now if we consider the second part of the last expression in particular, the right hand side (RHS) of the coupling condition in (2.12) becomes:

$$\lambda_1 \int_{\partial\Omega_1^n} \frac{\partial u_1}{\partial n_1}\phi_j ds = \lambda_1 \int_{\Omega_1^n} (-f\phi_j) dx + \lambda_1 \int_{\Omega_1^n} (\nabla u_1 \nabla \phi_j) dx =: f_1^\Gamma + \lambda_1 \int_{\Omega_1^n} (\nabla u_1 \nabla \phi_j) \tag{2.14}$$

and with analog notation the left hand side becomes:

$$\lambda_2 \int_{\partial\Omega_2^1} \frac{\partial u_2}{\partial n_1}\phi_j ds =: f_2^\Gamma + \lambda_2 \int_{\Omega_2^1} (\nabla u_2 \nabla \phi_j). \tag{2.15}$$

We approximate further by assuming $f_1^\Gamma = f_2^\Gamma = 0$. This lets us ignore the f_i^Γ-terms which simplifies the analysis and for our intended applications these terms will be very small and thus should not affect the convergence analysis. Next we insert the expressions for u_1 and u_2 where they are expressed with their basis functions.

RHS:

$$\lambda_1 \int_{\Omega_1^n} (\nabla u_1 \nabla \phi_j) = \lambda_1 \int_{\Omega_1^n} (\nabla(\hat{u}_1^n \phi_1^n + \hat{u}_\Gamma \phi_\Gamma) \nabla \phi_j) = \lambda_1 \int_{\Omega_1^n} \big(\hat{u}_1^n \times (-1) + \hat{u}_\Gamma \times (+1)\big) \nabla \phi_j. \tag{2.16}$$

LHS:

$$-\lambda_2 \int_{\Omega_2^1} (\nabla u_2 \nabla \phi_j) = -\lambda_2 \int_{\Omega_2^1} (\nabla(\hat{u}_\Gamma \phi_\Gamma + \hat{u}_2^1 \phi_2^1) \nabla \phi_j) = -\lambda_2 \int_{\Omega_2^1} \big(\hat{u}_\Gamma \times (-1) + \hat{u}_2^1 \times (+1)\big) \nabla \phi_j. \tag{2.17}$$

We then get:

$$\frac{\lambda_1}{\Delta x} \int_{\Omega_1^n} \big(\hat{u}_1^n \times (-1) + \hat{u}_\Gamma \times (+1)\big) \nabla \phi_\Gamma =! \frac{-\lambda_2}{\Delta x} \int_{\Omega_2^1} \big(\hat{u}_\Gamma \times (-1) + \hat{u}_2^1 \times (+1)\big) \nabla \phi_\Gamma \tag{2.18}$$

which, after taking into account that the sign of $\nabla\phi_\Gamma$ changes:

$$\frac{\lambda_1}{\Delta x^2}(\hat{u}_\Gamma - \hat{u}_1^n) =! \frac{\lambda_2}{\Delta x^2}(\hat{u}_2^1 - \hat{u}_\Gamma). \tag{2.19}$$

Here $\hat{u}_1^n$ denotes the last entry in the vector $\hat{\boldsymbol{u}}_\mathbf{1}$ containing the interior points in the first domain. It can equivalently be expressed the dot product of vectors $\boldsymbol{e}_n^T \hat{\boldsymbol{u}}_1$, analogosly $\hat{u}_2^1 = \boldsymbol{e}_1^T \hat{\boldsymbol{u}}_2$.

Chapter 3

Two-field case

Here we consider the case where the domain Ω is split into two domains of equal length and identical discretization, the sole difference being that the interface between them is adjoined to the second domain Ω_2. $\Omega = \Omega_1 \cup \Omega_2$, $\Omega_1 \cap \Omega_2 = \Gamma$. For the purpose of this work, n will denote the number of interior points of Ω_1 and Ω_2 respectively. Ω as a whole will thus have $2n + 3$ points, the two domains, start, end and boundary. We will use the same iteration as described in Fig. 2.1. That is we first solve Ω_1 with Dirichlet conditions on both sides, and then solve Ω_2 with a Dirichlet condition on the right side and a Neumann condition on the left side, updating the boundary value.

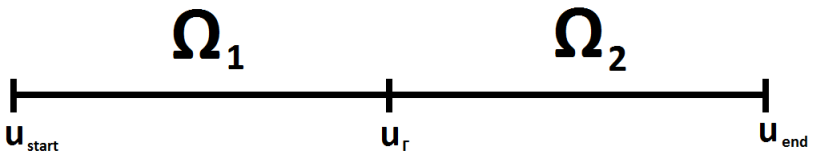

FIGURE 3.1: Division into two domains of equal length

We use the approximation for Δu as derived in (2.9) & (2.10). This discretization gives us the following system for Ω_1:

$$A_1 \boldsymbol{x} = \boldsymbol{b_1} \tag{3.1}$$

where

$$A_1 = \frac{1}{\Delta x^2}\begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{3.2}$$

and

$$\boldsymbol{b_1} = \begin{bmatrix} \frac{f_1^1}{\lambda_1} + \frac{u_{start}}{dx^2} \\ \frac{f_1^2}{\lambda_1} \\ \vdots \\ \frac{f_1^{n-1}}{\lambda_1} \\ \frac{f_1^n}{\lambda_1} + \frac{u_\Gamma}{dx^2} \end{bmatrix} \in \mathbb{R}^n. \tag{3.3}$$

Here the first and last entries in (3.3) contain the terms from the Dirichlet conditions. The solution obtained from solving this side will be denoted $\boldsymbol{u_1} = (u_1^1, u_1^2, \dots, u_1^n)^T$. For Ω_2, the system matrix is identical in structure but it has one more unknown, as solving this side also gives us a new value for u_Γ.

$$A_2 = \frac{1}{\Delta x^2}\begin{bmatrix} -\lambda_2 & \lambda_2 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \tag{3.4}$$

the corresponding vector is then:

$$\boldsymbol{b_2} = \begin{bmatrix} \frac{\lambda_1}{\Delta x^2}(\hat{u}_\Gamma - \hat{u}_1^n) \\ \frac{f_2^1}{\lambda_2} \\ \vdots \\ \frac{f_2^{n-1}}{\lambda_2} \\ \frac{f_2^n}{\lambda_2} + \frac{u_{end}}{dx^2} \end{bmatrix} \in \mathbb{R}^{(n+1)}. \tag{3.5}$$

The top entry of $\boldsymbol{b_2}$ combined with the first row of A_2 enforce the Neumann condition. The solution obtained by this system is denoted $\boldsymbol{u_2} = (u_\Gamma, u_2^1, u_2^2, \dots, u_2^n)^T$. The first row of this system corresponds to the equation (2.19).

Below is a pseudo-code of the iteration used.

Algorithm 1 Iteration given intial guess u_Γ^0 and endpoints u_{start}, u_{end}

flag=True
$u_\Gamma = u_\Gamma^0$
Construct A_1, A_2 as in (3.2) and (3.4)
for flag=True **do**
    Construct b_1 as in (3.3)
    Obtain $\mathbf{u_1}$ by solving $A_1 x = b_1$
    Construct b_2 as in (3.5)
    Obtain $\mathbf{u_2}$ by solving $A_2 x = b_2$
    $u_\Gamma^{old} = u_\Gamma$
    Extract new u_Γ from u_2
    If $|u_\Gamma^{old} - u_\Gamma| < tol$ set flag=False
end for
return $(u_{start}, \mathbf{u_1}, \mathbf{u_2}, u_{end})$ as solution.

3.1 Two-field case analysis

We rewrite the system presented by reordering the unknowns. The combined matrix for the two-field case can then be expressed as $A\boldsymbol{u} = \boldsymbol{f}$ where:

$$A = \begin{bmatrix} A_{II}^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma} \end{bmatrix} \tag{3.6}$$

and

$$\boldsymbol{u} = \begin{bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_\Gamma \end{bmatrix} \quad ; \quad \boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ f_\Gamma \end{bmatrix}. \tag{3.7}$$

Here $u_I^{(i)}$ denote the interior points of Ω_i and u_Γ denotes the boundary between Ω_1 and Ω_2. This notation allows us to separate the components as follows.

Interface to interface

$$A_{\Gamma\Gamma}^{(1)} = \frac{\lambda_1}{\Delta x^2} \; ; \; A_{\Gamma\Gamma}^{(2)} = \frac{\lambda_2}{\Delta x^2}. \tag{3.8}$$

Interior points

$$A_{II}^{(1)} = \frac{\lambda_1}{\Delta x^2} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}; A_{II}^{(2)} = \frac{\lambda_2}{\Delta x^2} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}; \tag{3.9}$$

points to interface

$$A_{I\Gamma}^{(1)} = \frac{-\lambda_1}{\Delta x^2}\boldsymbol{e_N} \ ; \ A_{I\Gamma}^{(2)} = \frac{-\lambda_2}{\Delta x^2}\boldsymbol{e_1}; \tag{3.10}$$

and

$$A_{\Gamma I}^{(1)} = \frac{-\lambda_1}{\Delta x^2}\boldsymbol{e_N}^T \ ; \ A_{\Gamma I}^{(2)} = \frac{-\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T. \tag{3.11}$$

Here $\boldsymbol{e_i}$ denotes the vector containing 1 at the i-th position and zeros everywhere else. We now extract the relevant sub-systems. We begin by looking at the first step of the method where Ω_1 is solved with Dirichlet boundary conditions on both sides. This gives us the following equation:

$$A_{II}^{(1)} u_I^{(1)^{(k+1)}} + A_{I\Gamma}^{(1)} u_\Gamma^k = \boldsymbol{f_1}.$$

Inserting the values and rearranging:

$$A_{II}^{(1)} u_I^{(1)^{(k+1)}} = \boldsymbol{f_1} + \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_N} u_\Gamma^k,$$

we thus have

$$u_I^{(1)^{(k+1)}} = A_{II}^{(1)^{-1}}(\boldsymbol{f_1} + \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n} u_\Gamma^k). \tag{3.12}$$

We move on to the second step of the iteration where we solve Ω_2 with Neumann condition on the left side and Dirichlet condition on the right side.

$$\begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & \widetilde{A}_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} u_I^{(2)^{(k+1)}} \\ u_\Gamma^{(k+1)} \end{bmatrix} = \begin{bmatrix} \widetilde{f}_2 \\ \widetilde{f}_\Gamma \end{bmatrix}. \tag{3.13}$$

Which in our case is:

$$\frac{\lambda_2}{\Delta x^2} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & 2 & -1 & 0 \\ 0 & \dots & 0 & -1 & 2 & 0 \\ -1 & 0 & \dots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_2^{1^{(k+1)}} \\ u_2^{2^{(k+1)}} \\ \vdots \\ \vdots \\ u_2^{n^{(k+1)}} \\ u_\Gamma^{k+1} \end{bmatrix} = \begin{bmatrix} f_2^1 \\ f_2^2 \\ \vdots \\ \vdots \\ f_2^n \\ \frac{\lambda_1}{\Delta x^2}(u_1^{n^{(k+1)}} - u_\Gamma^k) \end{bmatrix}. \tag{3.14}$$

From this we get two equations:

$$A_{II}^{(2)} u_I^{(2)^{(k+1)}} - \frac{\lambda_2}{\Delta x^2} u_\Gamma^{k+1} = \boldsymbol{f_2} \tag{3.15}$$

and

$$-\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T u_I^{(2)^{(k+1)}} + \frac{\lambda_2}{\Delta x^2} u_\Gamma^{k+1} = \frac{\lambda_1}{\Delta x^2} (\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - u_\Gamma^k). \tag{3.16}$$

Rewriting (3.15) we get

$$u_I^{(2)^{(k+1)}} = A_{II}^{(2)^{-1}} (\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1} u_\Gamma^{k+1}). \tag{3.17}$$

Rewriting (3.16):

$$\frac{\lambda_2}{\Delta x^2} u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\Delta x^2} \boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - \frac{\lambda_1}{\Delta x^2} u_\Gamma^k + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T u_I^{(2)^{(k+1)}} \tag{3.18}$$

multiplying both sides by $\frac{\Delta x^2}{\lambda_2}$:

$$u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\lambda_2} \boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - \frac{\lambda_1}{\lambda_2} u_\Gamma^k + \boldsymbol{e_1}^T u_I^{(2)^{(k+1)}}. \tag{3.19}$$

Inserting the expressions for the interior points (3.12) and (3.17) into (3.19):

$$u_\Gamma^{k+1} = \frac{\lambda_1}{\lambda_2} \boldsymbol{e_n}^T (A_{II}^{(1)^{-1}} (\boldsymbol{f_1} + \frac{\lambda_1}{\Delta x^2} \boldsymbol{e_n} u_\Gamma^k)) - \frac{\lambda_1}{\lambda_2} u_\Gamma^k + \boldsymbol{e_1}^T (A_{II}^{(2)^{-1}} (\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1} u_\Gamma^{k+1})).$$

As we are searching for a relation between u_Γ^k and u_Γ^{k+1}, we simplify the expression and collect all terms not dependent on either into ϕ:

$$u_\Gamma^{k+1} = \frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} u_\Gamma^k - \frac{\lambda_1}{\lambda_2} u_\Gamma^k + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1} u_\Gamma^{k+1} + \phi.$$

Rearranging:

$$(1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}) u_\Gamma^{k+1} = (\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}) u_\Gamma^k + \phi.$$

Finally, dividing both sides with $(1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1})$:

$$u_\Gamma^{k+1} = \frac{\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} u_\Gamma^k + \tilde{\phi}. \tag{3.20}$$

This is a fixed point iteration of the form seen in (2.1). As the iteration matrix is of dimension 1×1, it is a scalar and the spectral radius is given by its absolute value. We

thus have:

$$\rho(B) = \left| \frac{\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} \right| = \left| \frac{\lambda_1}{\lambda_2} \right| \left| \frac{\frac{\lambda_1}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - 1}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} \right|. \tag{3.21}$$

We now consider that the $A_{II}^{(i)}$'s are tridiagonal Toeplitz matrices, that is to say they are of the form $A_{II}^{(i)} = \text{Tridiagonal}(a, b, c)$. This is a well studied structure and in particular the general form of the eigendecomposition of symmetric tridiagonal Toeplitz-matrices is known [11]. Given:

$$A_{II}^{(i)^{-1}} = Tridiag(\frac{-\lambda_i}{\Delta x^2}, \frac{2\lambda_i}{\Delta x^2}, \frac{-\lambda_i}{\Delta x^2})^{-1} = V D_i^{-1} V \tag{3.22}$$

where D_i is the diagonal consisting of the eigenvalues. The matrix V is common for all symmetric, tridiagonal Toeplitz-matrices. It is given by [11]:

$$v_{ij} = \frac{1}{\sqrt{\sum_{k=1}^{n} \sin^2(\frac{k\pi}{n+1})}} \sin(\frac{ij\pi}{n+1}). \tag{3.23}$$

Further, if $A = \text{tridiag}(c, b, a)$ its eigenvalues and eigenvectors are given by [11]

$$\sigma_j = b + 2a\sqrt{\frac{c}{a}} \cos(\frac{j\pi}{n+1}) \tag{3.24}$$

$$x_{ij} = \frac{c}{a}^{\frac{i}{2}} \sin(\frac{ij\pi}{n+1}). \tag{3.25}$$

Looking at $A_{II}^{(i)^{-1}}$ specifically we have:

$$A_{\Gamma I}^{(m)} A_{II}^{(m)^{-1}} A_{I\Gamma}^{(m)} = \frac{-\lambda_m}{\Delta x^2} \boldsymbol{e}^T A_{II}^{(m)^{-1}} \frac{-\lambda_m}{\Delta x^2} \boldsymbol{e} = \frac{-\lambda_m}{\Delta x^2} \boldsymbol{e}^T V D_m^{-1} V \frac{-\lambda_m}{\Delta x^2} \boldsymbol{e} = (*). \tag{3.26}$$

Now we denote the diagonal matrix containing the inverse eigenvalues $D_m^{-1} = diag(\alpha_1, ..., \alpha_N)$, further we have symmetry in V as $v_{i,j} = v_{j,i}$:

$$(*) = \left(\frac{\lambda_m}{\Delta x^2}\right)^2 \sum_{i=1}^{n} \alpha_i v_{1,i}^2 = \left(\frac{\lambda_m}{\Delta x^2}\right)^2 \sum_{i=1}^{n} \frac{\sin^2(\frac{i\pi}{n+1})}{(\sum_{k=1}^{n} \sin^2(\frac{k\pi}{n+1}))(\frac{2\lambda_m}{\Delta x^2} + \lambda_m \cos(\frac{i\pi}{n+1}))}. \tag{3.27}$$

Pulling out λ_m

$$(*) = \Big(\frac{\lambda_m}{\Delta x^4}\Big) \sum_{i=1}^{n} \frac{\sin^2(\frac{i\pi}{n+1})}{(\sum_{k=1}^{n} \sin^2(\frac{k\pi}{n+1}))(\frac{2}{\Delta x^2} + \cos(\frac{i\pi}{n+1}))}. \tag{3.28}$$

In our case we have:

$$B = \frac{\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} = \frac{\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T V D_2^{-1} V \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T V D_2^{-1} V \boldsymbol{e_1}} \tag{3.29}$$

here (neglecting the normalization term in (3.23))

$$\boldsymbol{e_1}^T V = \sqrt{\frac{2}{n+1}} (\sin\big(\frac{1\pi}{n+1}\big), \sin\big(\frac{2\pi}{n+1}\big), ..., \sin\big(\frac{n\pi}{n+1}\big)) \tag{3.30}$$

thus:

$$\boldsymbol{e_1}^T V D^{-1} = \frac{\Delta x^2}{2\lambda_2} \sqrt{\frac{2}{n+1}} \Big(\frac{\sin\big(\frac{\pi}{n+1}\big)}{1 - \cos\big(\frac{\pi}{n+1}\big)}, \frac{\sin\big(\frac{2\pi}{n+1}\big)}{1 - \cos\big(\frac{2\pi}{n+1}\big)}, ..., \frac{\sin\big(\frac{n\pi}{n+1}\big)}{1 - \cos\big(\frac{n\pi}{n+1}\big)} \Big). \tag{3.31}$$

Further we have:

$$V \boldsymbol{e_1} = \sqrt{\frac{2}{n+1}} (\sin\big(\frac{1\pi}{n+1}\big), \sin\big(\frac{2\pi}{n+1}\big), ..., \sin\big(\frac{n\pi}{n+1}\big))^T \tag{3.32}$$

thus

$$\boldsymbol{e_1}^T A_{II}^{(i)^{-1}} \boldsymbol{e_1} = \boldsymbol{e_1}^T V D_j^{-1} V \boldsymbol{e_1} = \frac{\Delta x^2}{\lambda_j (n+1)} \sum_{i=1}^{n} \frac{\sin^2\big(\frac{i\pi}{n+1}\big)}{1 - \cos\big(\frac{i\pi}{n+1}\big)} = \frac{\Delta x^3}{\lambda_j} \sum_{i=1}^{n} \frac{\sin^2\big(\frac{i\pi}{n+1}\big)}{1 - \cos\big(\frac{i\pi}{n+1}\big)}. \tag{3.33}$$

We write out the other combinations as well:

$$V \boldsymbol{e_n} = \sqrt{\frac{2}{n+1}} (\sin\big(\frac{n\pi}{n+1}\big), \sin\big(\frac{2n\pi}{n+1}\big), ..., \sin\big(\frac{n^2\pi}{n+1}\big))^T \tag{3.34}$$

$$\boldsymbol{e_n}^T V = \sqrt{\frac{2}{n+1}} (\sin\big(\frac{n\pi}{n+1}\big), \sin\big(\frac{2n\pi}{n+1}\big), ..., \sin\big(\frac{n^2\pi}{n+1}\big)) \tag{3.35}$$

$$\boldsymbol{e_n}^T V D^{-1} = \frac{\Delta x^2}{2\lambda_2} \sqrt{\frac{2}{n+1}} \Big(\frac{\sin\big(\frac{n\pi}{n+1}\big)}{1 - \cos\big(\frac{\pi}{n+1}\big)}, \frac{\sin\big(\frac{2n\pi}{n+1}\big)}{1 - \cos\big(\frac{2\pi}{n+1}\big)}, ..., \frac{\sin\big(\frac{n^2\pi}{n+1}\big)}{1 - \cos\big(\frac{n\pi}{n+1}\big)} \Big).$$

Finally

$$\boldsymbol{e_n}^T V D_j^{-1} V \boldsymbol{e_n} = \frac{\Delta x^2}{2\lambda_j} \frac{2}{n+1} \sum_{i=1}^{n} \frac{\sin^2\big(\frac{in\pi}{n+1}\big)}{1 - \cos\big(\frac{i\pi}{n+1}\big)} = \frac{\Delta x^3}{\lambda_j} \sum_{i=1}^{n} \frac{\sin^2\big(\frac{in\pi}{n+1}\big)}{1 - \cos\big(\frac{i\pi}{n+1}\big)}. \tag{3.36}$$

Inserting:

$$B = \frac{\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} = \frac{\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \frac{\Delta x^3}{\lambda_1} \left(\sum_{i=1}^{n} \frac{\sin^2\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)} \right) - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \frac{\Delta x^3}{\lambda_2} \sum_{i=1}^{n} \frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}} \tag{3.37}$$

$$= \frac{\lambda_1}{\lambda_2} \frac{\Delta x \left(\sum_{i=1}^{n} \frac{\sin^2\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)} \right) - 1}{1 - \Delta x \sum_{i=1}^{n} \frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}} = \frac{-\lambda_1}{\lambda_2} \frac{1 - \Delta x \sum_{i=1}^{n} \frac{\sin^2\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}{1 - \Delta x \sum_{i=1}^{n} \frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}.$$

We wish to simplify this further, for this we look at the term $\sin^2\left(\frac{in\pi}{n+1}\right)$ and use the relation $\Delta x = 1/(n+1)$. We then have:

$$\sin^2\left(\frac{in\pi}{n+1}\right) = \sin^2\left((1-\Delta x)i\pi\right). \tag{3.38}$$

We now separate the two cases were i is either odd or even:

$$\sin\left((1-\Delta x)i\pi\right) = \sin\left(i\pi - \Delta x i\pi\right) = \begin{cases} -\sin\left(\Delta x i\pi\right); i \text{ even} \\ \sin\left(\Delta x i\pi\right); i \text{ odd.} \end{cases} \tag{3.39}$$

We can now rewrite B as:

$$B = \frac{-\lambda_1}{\lambda_2} \frac{1 - \Delta x \sum_{i=1}^{n} \frac{\sin^2\left(\Delta x i\pi\right)}{\left(1-\cos\left(i\pi\Delta x\right)\right)}}{1 - \Delta x \sum_{i=1}^{n} \frac{\sin^2\left(\Delta x i\pi\right)}{\left(1-\cos\left(i\pi\Delta x\right)\right)}} = \frac{-\lambda_1}{\lambda_2}. \tag{3.40}$$

We thus have:

$$\rho(B) = \frac{\lambda_1}{\lambda_2}. \tag{3.41}$$

By theorem 2.1. the necessary and sufficient condition for convergence is thus $\frac{\lambda_1}{\lambda_2} < 1$.

3.2 Whole domain solver

For the purpose of comparison we construct a separate solver which solves (2.4) on the whole domain directly as opposed to by iteration. The system is given by:

$$A = \frac{1}{\Delta x^2}\begin{bmatrix} 2\lambda_1 & -\lambda_1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ -\lambda_1 & 2\lambda_1 & -\lambda_1 & \ddots & & & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & \ddots & -\lambda_1 & 2\lambda_1 & -\lambda_1 & \ddots & & & \vdots \\ \vdots & & \ddots & -\lambda_1 & \lambda_1+\lambda_2 & -\lambda_2 & \ddots & & \vdots \\ \vdots & & & \ddots & -\lambda_2 & 2\lambda_2 & -\lambda_2 & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & \ddots & -\lambda_2 & 2\lambda_2 & -\lambda_2 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & -\lambda_2 & 2\lambda_2 \end{bmatrix} \in \mathbb{R}^{(2n+1)\times(2n+1)} \tag{3.42}$$

and

$$\boldsymbol{b_1} = \left[f_1^1 + \tfrac{u_{start}}{dx^2},\ f_1^2, \dots\ f_1^n,\ f_\Gamma,\ f_2^1,\ \dots\ f_2^{n-1},\ f_2^n + \tfrac{u_{end}}{dx^2}\right]^T \in \mathbb{R}^{2n+1} \tag{3.43}$$

with $A \in \mathbb{R}^{(3n+2)\times(3n+2)}$ and $b \in \mathbb{R}^{3n+2}$. Here u_{start} and u_{end} denote the start- and end-point values, respectively.

Chapter 4

Three-field case

We now extend the method to a three-field domain. We extend the two field problem

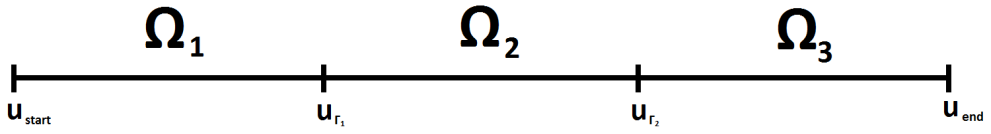

FIGURE 4.1: Division into three domains of equal length

from (2.4), we have $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ and interfaces $\Gamma_1 = \Omega_1 \cap \Omega_2$ and $\Gamma_2 = \Omega_2 \cap \Omega_3$.

$$\begin{cases} -\lambda_m \Delta u_m(x) = f_m \ ; \ x \in \Omega_m \subset \mathbb{R} \ ; \ m = 1,2,3 \\ u_m(x) = 0 \ ; \ u_m(x) \in \partial\Omega_m \backslash \{\Gamma_1, \Gamma_2\} \\ u_1(x) = u_2(x) \ ; \ x \in \Gamma_1 \\ u_2(x) = u_3(x) \ ; \ x \in \Gamma_2 \\ \lambda_1 \frac{\partial u_1(x)}{\partial n_1} = \lambda_2 \frac{\partial u_2(x)}{\partial n_2} \ ; \ x \in \Gamma_1 \\ \lambda_2 \frac{\partial u_2(x)}{\partial n_2} = \lambda_3 \frac{\partial u_3(x)}{\partial n_3} \ ; \ x \in \Gamma_2. \end{cases} \tag{4.1}$$

The previous derivations still hold true, what is new is that we now have two section boundary points in u_{Γ_1} and u_{Γ_2}. The general idea for the iteration will be to start with an initial guess for $(u_{\Gamma_1}, u_{\Gamma_2})$, then to solve the sections, starting by using the initial guesses as end-points. There are several ways one could go about constructing such an algorithm, in this work three variants are studied in detail.

The first method solves the equation (4.1) on Ω_1 using Dirichlet boundary conditions on both sides and then sequentially solving the equation on Ω_2 and Ω_3 with Dirichlet condition on the right hand side and with Neumann conditions on the left hand side, each updating the value at the boundary.

The second method solves the equation on section Ω_1 and Ω_3 with the initial guesses as endpoint and startpoint respectively, then solving the equation on Ω_2 and using the boundary condition derived above to obtain new values for both section boundary points. The models are explained in detail below.

The third method is analogous in structure to the second method but it begins by solving the equation on Ω_2.

4.1 Three-section split - Method 1

One iteration step of Method 1 is presented in Fig. 4.2:

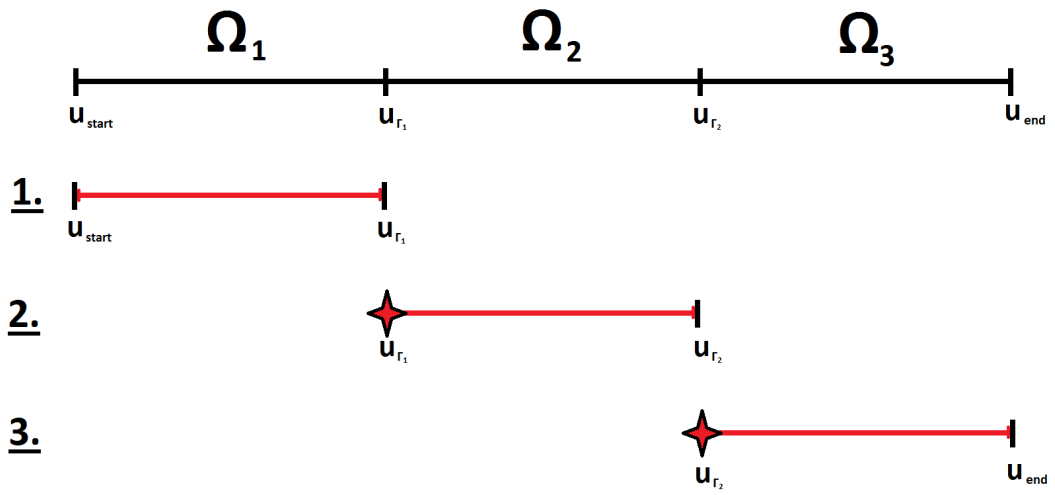

FIGURE 4.2: Division into three domains of equal length, star denotes Neumann condition being enforced, bar denotes Dirichlet

We write the combined matrix:

$$\begin{bmatrix} A_{II}^{(1)} & 0 & 0 & A_{I\Gamma_1}^{(1)} & 0 \\ 0 & A_{II}^{(2)} & 0 & A_{I\Gamma_1}^{(2)} & A_{I\Gamma_2}^{(2)} \\ 0 & 0 & A_{II}^{(3)} & 0 & A_{I\Gamma_2}^{(3)} \\ A_{\Gamma_1 I}^{(1)} & A_{\Gamma_1 I}^{(2)} & 0 & A_{\Gamma_1\Gamma_1} & 0 \\ 0 & A_{\Gamma_2 I}^{(2)} & A_{\Gamma_2 I}^{(3)} & 0 & A_{\Gamma_2\Gamma_2} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_I^{(3)} \\ u_{\Gamma_1} \\ u_{\Gamma_2} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_{\Gamma_1} \\ f_{\Gamma_2} \end{bmatrix}. \tag{4.2}$$

Here $u_I^{(i)}$ denotes the solution values in the interior points of Ω_i and u_{Γ_i} denotes the solution values on the boundaries. We now wish to extract subsystems, and begin by step 2 of the algorithm, that is solving the equation on Ω_2 with Neumann conditions on the left hand side and Dirichlet boundary condition on the right hand side. When we extract sub-systems, we have to rearrange the system as we no longer have access to all the solution values. For example, in the second step of the iteration, the value of u_{Γ_2} is treated as a Dirichlet boundary condition and is thus moved into the term $\widetilde{f_{\Gamma_1}}$. The second step can thus be expressed:

$$\begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma_1}^{(2)} \\ A_{\Gamma_1 I}^{(2)} & \widetilde{A}_{\Gamma_1\Gamma_1}^{(2)} \end{bmatrix} \begin{bmatrix} u_I^{(2)^{(k+1)}} \\ u_{\Gamma_1}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \widetilde{f_2} \\ \widetilde{f_{\Gamma_1}} \end{bmatrix} \tag{4.3}$$

which in our case is:

$$\frac{\lambda_2}{\Delta x^2} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & 2 & -1 & 0 \\ 0 & \dots & 0 & -1 & 2 & 0 \\ -1 & 0 & \dots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_2^{1(k+1)} \\ u_2^{2(k+1)} \\ \vdots \\ \vdots \\ u_2^{n(k+1)} \\ u_{\Gamma_1}^{(k+1)} \end{bmatrix} = \begin{bmatrix} f_2^1 \\ f_2^2 \\ \vdots \\ \vdots \\ f_2^n + \frac{\lambda_2 u_{\Gamma_2}^k}{\Delta x^2} \\ \frac{\lambda_1}{\Delta x^2}(u_1^{n(k+1)} - u_{\Gamma_1}^k) \end{bmatrix}. \tag{4.4}$$

From this we get two equations:

$$A_{II}^{(2)} u_I^{(2)^{(k+1/2)}} + A_{I\Gamma_1}^{(2)} u_{\Gamma_1}^{(k+1)} = A_{II}^{(2)} u_I^{(2)^{(k+1/2)}} - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1} u_{\Gamma_1}^{(k+1)} = \boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n} u_{\Gamma_2}^k \tag{4.5}$$

and

$$-\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T u_I^{(2)^{(k+1)}} + \frac{\lambda_2}{\Delta x^2} u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\Delta x^2} (\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - u_{\Gamma_1}^k). \tag{4.6}$$

Rewriting (4.5) we get

$$\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)} = A_{II}^{(2)}u_I^{(2)^{(k+1)}} - \boldsymbol{f_2} - \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^k. \tag{4.7}$$

Rewriting (4.6):

$$\frac{\lambda_2}{\Delta x^2}u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - \frac{\lambda_1}{\Delta x^2}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T u_I^{(2)^{(k+1)}}. \tag{4.8}$$

Now we do the same thing for step 3; this system is analogous to step two in (4.4), here we solve the equation on Ω_3 with Neumann conditions on the left hand side and Dirichlet boundary condition on the right hand side. The system is:

$$\frac{\lambda_3}{\Delta x^2}\begin{bmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & 2 & -1 & 0 \\ 0 & \dots & 0 & -1 & 2 & 0 \\ -1 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} u_3^{1(k+1)} \\ u_3^{2(k+1)} \\ \vdots \\ \vdots \\ u_3^{n(k+1)} \\ u_{\Gamma_2}^{(k+1)} \end{bmatrix} = \begin{bmatrix} f_3^1 \\ f_3^2 \\ \vdots \\ \vdots \\ f_3^n \\ \frac{\lambda_2}{\Delta x^2}(u_2^{n(k+1)} - u_{\Gamma_2}^k) \end{bmatrix}. \tag{4.9}$$

From this we get two equations:

$$A_{II}^{(3)}u_I^{(3)^{(k+1)}} + A_{I\Gamma_1}^{(3)}u_{\Gamma_2}^{(k+1)} = A_{II}^{(3)}u_I^{(3)^{(k+1)}} - \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_2}^{(k+1)} = \boldsymbol{f_3} \tag{4.10}$$

and

$$-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T u_I^{(3)^{(k+1)}} + \frac{\lambda_3}{\Delta x^2}u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}^T u_I^{(2)^{(k+1)}} - \frac{\lambda_2}{\Delta x^2}u_{\Gamma_2}^k. \tag{4.11}$$

We now rewrite equations (4.8) and (4.11) that we have obtained from the two systems and divide both sides with the respective $\frac{\lambda_i}{\Delta x^2}$:

$$u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\lambda_2}\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - \frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k + \boldsymbol{e_1}^T u_I^{(2)^{(k+1)}} \tag{4.12}$$

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2}{\lambda_3}\boldsymbol{e_n}^T u_I^{(2)^{(k+1)}} - \frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k + \boldsymbol{e_1}^T u_I^{(3)^{(k+1)}}. \tag{4.13}$$

Writing out the formulas for the interior points, the first equation is simply solving on Ω_1 with Dirichlet conditions on both sides, the other two follow from (4.5) and (4.10).

$$u_I^{(1)^{(k+1)}} = A_{II}^{(1)^{-1}}(\boldsymbol{f_1} + \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_1}^k) =: A_{II}^{(1)^{-1}}\boldsymbol{b_1} \tag{4.14}$$

$$u_I^{(2)^{(k+1)}} = A_{II}^{(2)^{-1}}(\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^k) \tag{4.15}$$

$$u_I^{(3)^{(k+1)}} = A_{II}^{(3)^{-1}}(\boldsymbol{f_3} + \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_2}^{(k+1)}). \tag{4.16}$$

Inserting into (4.12):

$$u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\lambda_2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}(\boldsymbol{f_1}+\frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_1}^k)-\frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k+\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}(\boldsymbol{f_2}+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)}+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^k).$$

Rewriting and collecting all terms not dependent on the boundary values into Φ_1:

$$u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}u_{\Gamma_1}^k-\frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)}+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k+\Phi_1.$$

As we wish to find $u_{\Gamma_1}^{(k+1)}(u_{\Gamma_1}^k, u_{\Gamma_2}^k)$, we move all $u_{\Gamma_1}^{(k+1)}$ terms to one side:

$$(1-\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1})u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}u_{\Gamma_1}^k-\frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k+\Phi_1.$$

Dividing:

$$u_{\Gamma_1}^{(k+1)} = \frac{\frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}u_{\Gamma_1}^k - \frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k + \Phi_1}{1 - \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}}. \tag{4.17}$$

As we later wish to represent this as a linear fixed point iteration, we separate the u_Γ's; all terms not dependent on the boundary values merge into $\widetilde{\Phi}_1$.

$$u_{\Gamma_1}^{(k+1)} = \frac{(\frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2})}{1 - \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}}u_{\Gamma_1}^k + \frac{\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}}{1 - \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}}u_{\Gamma_2}^k + \widetilde{\Phi}_1. \tag{4.18}$$

We now do the same for u_Γ^2, inserting (4.15) and (4.16) into (4.13).

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2}{\lambda_3}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}(\boldsymbol{f_2}+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)}+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^k)-\frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k+\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}(\boldsymbol{f_3}+\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_2}^{(k+1)}).$$

Rewriting and collecting all terms not dependent on the boundary values into Φ_2:

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)}+\frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k-\frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k+\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}u_{\Gamma_2}^{(k+1)}+\Phi_2.$$

As before, we wish to find $u_{\Gamma_2}^{(k+1)}(u_{\Gamma_1}^k, u_{\Gamma_2}^k)$, we move all $u_{\Gamma_2}^{(k+1)}$ terms to one side:

$$(1-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1})u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)}+\frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k-\frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k+\Phi_2.$$

To integrate this into the linear fixed point iteration, we separate the u_Γ's; all terms not dependent on the boundary values merge into $\widetilde{\Phi}_2$.

$$u_{\Gamma_2}^{(k+1)} = \frac{\frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}}{1-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}}u_{\Gamma_1}^{k+1} + \frac{(\frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}-\frac{\lambda_2}{\lambda_3})}{1-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}}u_{\Gamma_2}^k + \widetilde{\Phi}_2. \tag{4.19}$$

We may write (4.18) and (4.19) as:

$$u_{\Gamma_1}^{(k+1)} = a_1u_{\Gamma_1}^k + b_1u_{\Gamma_2}^k + \widetilde{\Phi}_1 \tag{4.20}$$

$$u_{\Gamma_2}^{(k+1)} = a_2u_{\Gamma_1}^{(k+1)} + b_2u_{\Gamma_2}^k + \widetilde{\Phi}_2 = a_2(a_1u_{\Gamma_1}^k + b_1u_{\Gamma_2}^k + \widetilde{\Phi}_1) + b_2u_{\Gamma_2}^k + \widetilde{\Phi}_2 \tag{4.21}$$

where:

$$a_1 = \frac{\frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}-\frac{\lambda_1}{\lambda_2}}{1-\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}} \tag{4.22}$$

$$b_1 = \frac{\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}}{1-\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}} \tag{4.23}$$

$$a_2 = \frac{\frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}}{1-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}} \tag{4.24}$$

$$b_2 = \frac{\frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}-\frac{\lambda_2}{\lambda_3}}{1-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}}. \tag{4.25}$$

We get the following fixed point iteration (as defined in (2.1)):

$$\begin{bmatrix} u_{\Gamma_1}^{(k+1)} \\ u_{\Gamma_2}^{(k+1)} \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_1a_2 & (a_2b_1+b_2) \end{bmatrix} \begin{bmatrix} u_{\Gamma_1}^k \\ u_{\Gamma_2}^k \end{bmatrix} + \begin{bmatrix} \widetilde{\Phi}_1 \\ a_2\widetilde{\Phi}_1 + \widetilde{\Phi}_2 \end{bmatrix}. \tag{4.26}$$

As we wish to find $max_i(|\sigma_i|)$, we would want to have the explicit forms of the eigenvalues. To calculate these, we have to recall some of the basic properties of the inverses of tridiagonal Toeplitz matrices. Recall that as we saw in (3.22),(3.23) and (3.24), the matrix:

$$M = \frac{\lambda}{\Delta x^2}\begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{4.27}$$

has eigenvalues

$$\sigma_j = \frac{2\lambda}{\Delta x^2} - \frac{2\lambda}{\Delta x^2}\cos\left(\frac{j\pi}{n+1}\right) \tag{4.28}$$

with corresponding eigenvectors:

$$v_j = \begin{bmatrix} \sin\left(\frac{1j\pi}{n+1}\right) \\ \sin\left(\frac{2j\pi}{n+1}\right) \\ \sin\left(\frac{3j\pi}{n+1}\right) \\ \vdots \\ \sin\left(\frac{nj\pi}{n+1}\right) \end{bmatrix}. \tag{4.29}$$

As M is a real valued positive definite symmetric matrix, we can decompose it as:

$$M = VDV^T \tag{4.30}$$

where V are orthogonal matrices composed of the eigenvectors. Further we have symmetry such that $V = V^T$. The inverse then becomes:

$$M^{-1} = VD^{-1}V \tag{4.31}$$

and we have

$$M = VDV \tag{4.32}$$

where:

$$V = \sqrt{\frac{2}{n+1}}\begin{bmatrix} \sin\left(\frac{1\pi}{n+1}\right) & \sin\left(\frac{2\pi}{n+1}\right) & \dots & \sin\left(\frac{n\pi}{n+1}\right) \\ \sin\left(\frac{2\pi}{n+1}\right) & \sin\left(\frac{4\pi}{n+1}\right) & \ddots & \vdots \\ \sin\left(\frac{3\pi}{n+1}\right) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \sin\left(\frac{(n-1)(n-1)\pi}{n+1}\right) & \sin\left(\frac{n(n-1)\pi}{n+1}\right) \\ \sin\left(\frac{n\pi}{n+1}\right) & \dots & \sin\left(\frac{n(n-1)\pi}{n+1}\right) & \sin\left(\frac{n^2\pi}{n+1}\right) \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{4.33}$$

Note that V is identical for all matrices of the form Tridiag$(\frac{-\lambda_i}{\Delta x^2}, \frac{2\lambda_i}{\Delta x^2}, \frac{-\lambda_i}{\Delta x^2})$, only D differs.

And D is the diagonal matrix with

$$D = \begin{bmatrix} \frac{2\lambda}{\Delta x^2} - \frac{2\lambda}{\Delta x^2}\cos\left(\frac{\pi}{n+1}\right) & & & \\ & \frac{2\lambda}{\Delta x^2} - \frac{2\lambda}{\Delta x^2}\cos\left(\frac{2\pi}{n+1}\right) & & \\ & & \ddots & \\ & & & \frac{2\lambda}{\Delta x^2} - \frac{2\lambda}{\Delta x^2}\cos\left(\frac{n\pi}{n+1}\right) \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{4.34}$$

and D^{-1} is the diagonal matrix containing the corresponding inverses.

Now we look at the terms used in the iteration matrix in (4.26). The first thing to note is that a_1 in (4.22) is identical to B in (3.40). We thus have:

$$a_1 = \frac{\frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}} = \frac{-\lambda_1}{\lambda_2}. \tag{4.35}$$

As we move on to the other terms, we will need to use (3.33) and (3.36). We also write out the other relevant combinations $\boldsymbol{e_i}^T V D_i^{-1} V \boldsymbol{e_j}$, due to the symmetry of V we have:

$$\boldsymbol{e_1}^T V D_i^{-1} V \boldsymbol{e_n} = \boldsymbol{e_n}^T V D_i^{-1} V \boldsymbol{e_1} = \frac{\Delta x^3}{\lambda_i}\sum_{i=1}^{n}\frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}. \tag{4.36}$$

We can now begin with b_1 from (4.23):

$$b_1 = \frac{\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}}{1 - \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}} = \frac{\frac{\lambda_2}{\Delta x^2}\frac{\Delta x^3}{\lambda_2}\sum_{i=1}^{n}\frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}{1 - \frac{\lambda_2}{\Delta x^2}\frac{\Delta x^3}{\lambda_2}\sum_{i=1}^{n}\frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}$$

$$= \frac{\Delta x\sum_{i=1}^{n}\frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}}{1 - \Delta x\sum_{i=1}^{n}\frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}} = \frac{\Delta x\sum_{i=1}^{n}\frac{\sin\left(i\pi\Delta x\right)(-1)^{i+1}\sin\left(i\pi\Delta x\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}{1 - \Delta x\sum_{i=1}^{n}\frac{\sin^2\left(i\pi\Delta x\right)}{1-\cos\left(i\pi\Delta x\right)}}$$

$$= \frac{\Delta x\sum_{i=1}^{n}\frac{(-1)^{i+1}\sin^2\left(i\pi\Delta x\right)}{1-\cos\left(i\pi\Delta x\right)}}{1 - \Delta x\sum_{i=1}^{n}\frac{\sin^2\left(i\pi\Delta x\right)}{1-\cos\left(i\pi\Delta x\right)}}, = \frac{\Delta x\sum_{i=1}^{n}(-1)^{i+1}(\cos(i\pi\Delta x)+1)}{1 - \Delta x\sum_{i=1}^{n}(\cos(i\pi\Delta x)+1)}.$$

Here we used (3.39) and the Pythagorean identity. Before we go on simplifying, let us first consider a_2 from (4.24), using the symmetry from (4.36) and the fact that all λ_i in the divisor cancel out as seen in (3.37):

$$\frac{\lambda_3}{\lambda_2}a_2 = \frac{\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}}{1-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}} = \frac{\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}}{1-\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}} = b_1. \tag{4.37}$$

As a_2 and b_1 only differ by a scalar multiplication, finding the value of one will give us the value of both. Numerical tests were carried out for different discretizations, suggesting that:

$$b_1 = \frac{\lambda_3}{\lambda_2}a_2 = \frac{\Delta x \sum_{i=1}^{n} \frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}}{1-\Delta x\sum_{i=1}^{n}\frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}} =_! 1. \tag{4.38}$$

We now wish to analytically show that this is true for all n. Using this assumption will be helpful as it allows us to simplify the problem into a single sum. Rewriting:

$$\Delta x\sum_{i=1}^{n}\frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)} =_! 1-\Delta x\sum_{i=1}^{n}\frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}$$

$$\sum_{i=1}^{n}\frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)+\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)} =_! n+1$$

using the identities from (3.39) and splitting the sums into even and odd parts:

$$=\sum_{i,odd}^{n}\frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{i\pi}{n+1}\right)+\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}+\sum_{i,even}^{n}\frac{-\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{i\pi}{n+1}\right)+\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}.$$

The even part is zero, we are left with the odd part:

$$=\sum_{i,odd}^{n}\frac{2\sin^2\left(\frac{i\pi}{n+1}\right)}{\left(1-\cos\left(\frac{i\pi}{n+1}\right)\right)} =_{trig.identity} \sum_{i,odd}^{n}2\frac{1-\cos^2\left(\frac{i\pi}{n+1}\right)}{\left(1-\cos\left(\frac{i\pi}{n+1}\right)\right)} = \sum_{i,odd}^{n}2\big(\cos(\frac{i\pi}{n+1})+1\big).$$

$$=\sum_{i,odd}^{n}2+\sum_{i,odd}^{n}2\cos(\frac{i\pi}{n+1}) =_! n+1. \tag{4.39}$$

We now consider the cases where n is even and odd separately[1].

[1]The author thanks D. Svensson Seth for his help in completing this proof.

n **even**

We then have:

$$\sum_{i,odd}^{n} 2 = n. \tag{4.40}$$

It remains to show that

$$\sum_{i,odd}^{n} 2\cos(\frac{i\pi}{n+1}) =_! 1. \tag{4.41}$$

We re-index the sum

$$2\sum_{j=0}^{n/2-1}\cos\left(\frac{(2j+1)\pi}{n+1}\right) = 2\sum_{k=1}^{n/2}\cos\left(\frac{(2k-1)\pi}{n+1}\right)$$

and note

$$2\sum_{k=1}^{n/2}\cos\left(\frac{(2k-1)\pi}{n+1}\right) + 2\sum_{l=1}^{n/2}\cos\left(\frac{2l\pi}{n+1}\right) =$$

$$2\sum_{k=1}^{n/2}\cos\left(\frac{(2k-1)\pi}{n+1}\right) + 2\sum_{k=1}^{n/2}\cos\left(\frac{(n+2-2k)\pi}{n+1}\right) = 0$$

where we reversed the order of summation in the second sum. Since [12]

$$\begin{aligned}&\cos\left(\frac{(2k-1)\pi}{n+1}\right) + \cos\left(\frac{(n+2-2k)\pi}{n+1}\right)\\ &\quad= 2\cos\left(\frac{1}{2}\left(\frac{(2k-1)\pi}{n+1} + \frac{(n+2-2k)\pi}{n+1}\right)\right)\cos\left(\frac{1}{2}\left(\frac{(2k-1)\pi}{n+1} - \frac{(n+2-2k)\pi}{n+1}\right)\right)\end{aligned} \tag{4.42}$$

where

$$\cos\left(\frac{1}{2}\left(\frac{(2k-1)\pi}{n+1} + \frac{(n+2-2k)\pi}{n+1}\right)\right) = \cos\left(\frac{\pi}{2}\right) = 0.$$

Hence it is sufficient to show

$$2\sum_{l=1}^{n/2} cos\left(\frac{2l\pi}{n+1}\right) = -1.$$

We apply the formula for the Dirichlet kernel [13]

$$1 + 2\sum_{l=1}^{m} cos(lx) = \frac{sin((m+1/2)x)}{sin(x/2)}.$$

With $m = n/2$ and $x = \frac{2\pi}{n+1}$ the right hand side becomes

$$\frac{sin\left((n/2+1/2)\frac{2\pi}{n+1}\right)}{sin\left(\frac{2\pi}{2(n+1)}\right)} = \frac{sin\left(\pi\right)}{sin\left(\frac{\pi}{n+1}\right)} = 0$$

so the desired formula (4.38) follows, i.e., that when n is even, we have $b_1 = 1$.

n odd

We then have:

$$\sum_{i,odd}^{n} 2 = n + 1. \tag{4.43}$$

It remains to show that (re-indexing the sum again);

$$\sum_{i,odd}^{n} 2\cos(\frac{i\pi}{n+1}) = 2\sum_{j=0}^{(n-1)/2} \cos\left(\frac{(2j+1)\pi}{n+1}\right) = 2\sum_{k=1}^{(n-1)/2+1} \cos\left(\frac{(2k-1)\pi}{n+1}\right) =_! 0. \tag{4.44}$$

We can pair the k:th term with the $((n+3)/2-k)$:th term, for any $1 \leq k \leq (n+1)/2$. Summing these terms gives the same expression as in (4.42) so the terms in the sum cancel pairwise this way. However, there is one exceptional case if $(n+1)/2$ is odd. This means that we have an odd number of terms in the sum so naturally we get one term which we cannot pair with another term in this way. However it is the $((n-1)/4+1)$:th term with $j = (n-1)/4$ so we can evaluate this term separately and get

$$cos\left(\frac{(2(n-1)/4+1)\pi}{n+1}\right) = cos\left(\frac{\pi}{2}\right) = 0.$$

Thus (4.38) also holds true when n is odd, and we have:

$$b_1 = \frac{\Delta x \sum_{i=1}^{n} \frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}}{1 - \Delta x \sum_{i=1}^{n} \frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(i\pi\Delta x\right)}} = 1. \tag{4.45}$$

□

Finally we go onto b_2 from (4.25):

$$b_2 = \frac{\frac{\lambda_2^2}{\lambda_3 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(3)^{-1}} \boldsymbol{e_n} - \frac{\lambda_2}{\lambda_3}}{1 - \frac{\lambda_3}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} = \frac{\frac{\lambda_2^2}{\lambda_3 \Delta x^2} \frac{\Delta x^3}{\lambda_2} \sum_{i=1}^{n} \frac{\sin^2\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)} - \frac{\lambda_2}{\lambda_3}}{1 - \frac{\lambda_3}{\Delta x^2} \frac{\Delta x^3}{\lambda_3} \sum_{i=1}^{n} \frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}.$$

We can rewire this in the same way we did for B in (3.37):

$$b_2 = \frac{-\lambda_2}{\lambda_3} \frac{1 - \Delta x \sum_{i=1}^{n} \frac{\sin^2\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}{1 - \Delta x \sum_{i=1}^{n} \frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}} = \frac{-\lambda_2}{\lambda_3}.$$

We thus have:

$$a_1 = \frac{\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} = \frac{-\lambda_1}{\lambda_2} \tag{4.46}$$

$$b_1 = \frac{\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} = 1 \tag{4.47}$$

$$a_2 = \frac{\frac{\lambda_2^2}{\lambda_3 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}}{1 - \frac{\lambda_3}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1}} = \frac{\lambda_2}{\lambda_3} \frac{\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}}{1 - \frac{\lambda_3}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1}} = \frac{\lambda_2}{\lambda_3} \tag{4.48}$$

$$b_2 = \frac{\frac{\lambda_2^2}{\lambda_3 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n} - \frac{\lambda_2}{\lambda_3}}{1 - \frac{\lambda_3}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1}} = \frac{-\lambda_2}{\lambda_3}. \tag{4.49}$$

Inserting the values into the matrix in (4.26) we have:

$$\begin{bmatrix} \frac{-\lambda_1}{\lambda_2} & 1 \\ \frac{-\lambda_1}{\lambda_3} & \frac{\lambda_2}{\lambda_3} + \frac{-\lambda_2}{\lambda_3} \end{bmatrix} = \begin{bmatrix} \frac{-\lambda_1}{\lambda_2} & 1 \\ \frac{-\lambda_1}{\lambda_3} & 0 \end{bmatrix} \tag{4.50}$$

with eigenvalues:

$$\sigma_{1,2} = -\frac{1}{2}\frac{\lambda_1}{\lambda_2} \pm \sqrt{\frac{1}{4}\frac{\lambda_1^2}{\lambda_2^2} - \frac{\lambda_1}{\lambda_3}}. \tag{4.51}$$

Note: This result shows us that as $\frac{\lambda_1}{\lambda_3} \to 0$ the spectral radius of the iteration matrix $|\sigma|_{max} \to \frac{\lambda_1}{\lambda_2}$, which is the convergence speed of the two field case considered earlier.

Below is the pseudo-code for Method 1.

Algorithm 2 Method 1: Iteration given intial guess $(u^0_{\Gamma_1}, u^0_{\Gamma_2})$ and endpoints u_{start}, u_{end}

flag=True
$(u_{\Gamma_1}, u_{\Gamma_2}) = (u^0_{\Gamma_1}, u^0_{\Gamma_2})$
Construct A_1, A_2, A_3 as in (3.9), (4.4) and (4.9)
while flag=True **do**
    Construct b_1 as in (4.14)
    Obtain $\mathbf{u_1}$ by solving $A_1 x = b_1$
    Construct b_2 as in (4.4)
    Obtain $\mathbf{u_2}$ by solving $A_2 x = b_2$
    $u^{old}_{\Gamma_1} = u_{\Gamma_1}$
    Extract new u_Γ from u_2
    Construct b_3 as in (4.9)
    Obtain $\mathbf{u_3}$ by solving $A_3 x = b_3$
    $u^{old}_{\Gamma_2} = u_{\Gamma_2}$
    Extract new u_{Γ_2} from u_3
    If $||(u^{old}_{\Gamma_1}, u^{old}_{\Gamma_2}) - (u_{\Gamma_1}, u_{\Gamma_2})|| < tol$ set flag=False
end while
return $(u_{start}, \mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, u_{end})$ as solution.

4.2 Three-section split - Method 2

One iteration step of Method 2 is presented in Fig. 4.3:

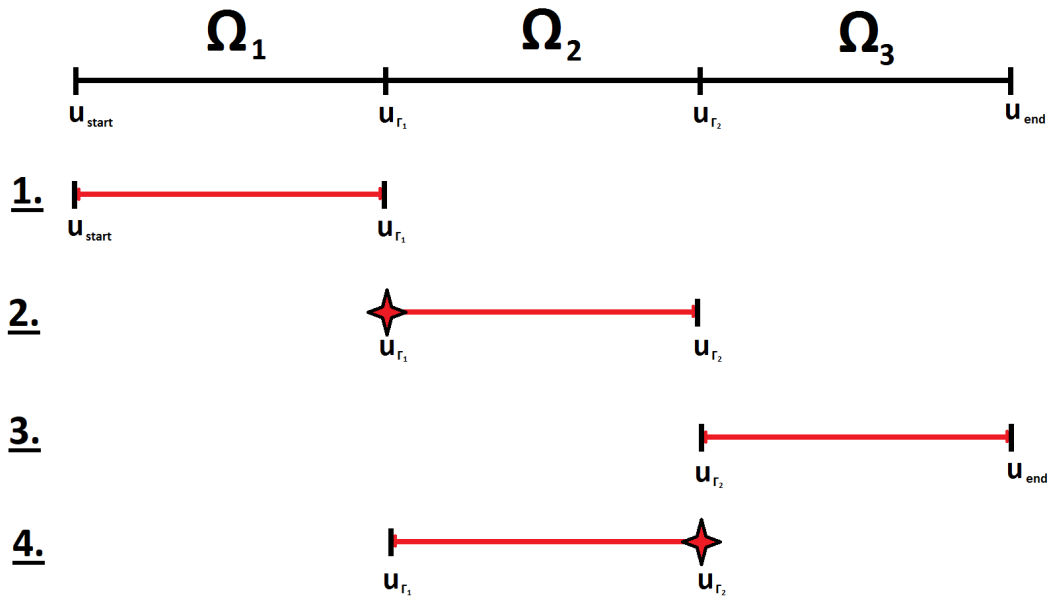

FIGURE 4.3: Division into three domains of equal length, star denotes Neumann condition being enforced, bar denotes Dirichlet

We write the combined matrix:

$$\begin{bmatrix} A_{II}^{(1)} & 0 & 0 & A_{I\Gamma_1}^{(1)} & 0 \\ 0 & A_{II}^{(2)} & 0 & A_{I\Gamma_1}^{(2)} & A_{I\Gamma_2}^{(2)} \\ 0 & 0 & A_{II}^{(3)} & 0 & A_{I\Gamma_2}^{(3)} \\ A_{\Gamma_1 I}^{(1)} & A_{\Gamma_1 I}^{(2)} & 0 & A_{\Gamma_1\Gamma_1} & 0 \\ 0 & A_{\Gamma_2 I}^{(2)} & A_{\Gamma_2 I}^{(3)} & 0 & A_{\Gamma_2\Gamma_2} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_I^{(3)} \\ u_{\Gamma_1} \\ u_{\Gamma_2} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_{\Gamma_1} \\ f_{\Gamma_2} \end{bmatrix}. \tag{4.52}$$

We now wish to extract subsystems, we apply the same procedure as we did in (4.3). We begin by step 2 of the algorithm, that is solving Ω_2 with a Neumann condition on the left hand side and a Dirichlet boundary condition on the right hand side.

$$\begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma_1}^{(2)} \\ A_{\Gamma_1 I}^{(2)} & \widetilde{A}_{\Gamma_1\Gamma_1}^{(2)} \end{bmatrix} \begin{bmatrix} u_I^{(2)^{(k+1/2)}} \\ u_{\Gamma_1}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \widetilde{f_2} \\ \widetilde{f_{\Gamma_1}} \end{bmatrix}. \tag{4.53}$$

Which in our case is:

$$\frac{\lambda_2}{\Delta x^2} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & 2 & -1 & 0 \\ 0 & \dots & 0 & -1 & 2 & 0 \\ -1 & 0 & \dots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_2^{1^{(k+1/2)}} \\ u_2^{2^{(k+1/2)}} \\ \vdots \\ \vdots \\ u_2^{n^{(k+1/2)}} \\ u_{\Gamma_1}^{(k+1)} \end{bmatrix} = \begin{bmatrix} f_2^1 \\ f_2^2 \\ \vdots \\ \vdots \\ f_2^n + \frac{\lambda_2 u_{\Gamma_2}^k}{\Delta x^2} \\ \frac{\lambda_1}{\Delta x^2}(u_1^{n^{(k+1)}} - u_{\Gamma_1}^k) \end{bmatrix}. \tag{4.54}$$

From this we get two equations:

$$A_{II}^{(2)} u_I^{(2)^{(k+1/2)}} + A_{I\Gamma_1}^{(2)} u_{\Gamma_1}^{(k+1)} = A_{II}^{(2)} u_I^{(2)^{(k+1/2)}} - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1} u_{\Gamma_1}^{(k+1)} = \boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n} u_{\Gamma_2}^k \tag{4.55}$$

and

$$-\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T u_I^{(2)^{(k+1/2)}} + \frac{\lambda_2}{\Delta x^2} u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\Delta x^2} (\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - u_{\Gamma_1}^k). \tag{4.56}$$

Rewriting (4.55) we get

$$\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1} u_{\Gamma_1}^{(k+1)} = A_{II}^{(2)} u_I^{(2)^{(k+1/2)}} - \boldsymbol{f_2} - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n} u_{\Gamma_2}^k. \tag{4.57}$$

Rewriting (4.56):

$$\frac{\lambda_2}{\Delta x^2}u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - \frac{\lambda_1}{\Delta x^2}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T u_I^{(2)^{(k+1/2)}}. \tag{4.58}$$

Now we do the same thing for step 4 where we solve the equation on Ω_2 with a Neumann condition on the right hand side, updating u_{Γ_2}, and a Dirichlet condition on the left hand side.

$$\begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma_1}^{(2)} \\ A_{\Gamma_1 I}^{(2)} & \widetilde{A}_{\Gamma_2\Gamma_2}^{(2)} \end{bmatrix} \begin{bmatrix} u_I^{(2)^{(k+1)}} \\ u_{\Gamma_2}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \widetilde{f_2} \\ \widetilde{f_{\Gamma_2}} \end{bmatrix}. \tag{4.59}$$

In our case this corresponds to:

$$\frac{\lambda_2}{\Delta x^2}\begin{bmatrix} 2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & 2 & -1 & 0 \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & 0 & \dots & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_2^{1^{(k+1)}} \\ u_2^{2^{(k+1)}} \\ \vdots \\ \vdots \\ u_2^{n^{(k+1)}} \\ u_{\Gamma_2}^{(k+1)} \end{bmatrix} = \begin{bmatrix} f_2^1 + \frac{\lambda_2 u_{\Gamma_1}^{(k+1)}}{\Delta x^2} \\ f_2^2 \\ \vdots \\ \vdots \\ f_2^n \\ \frac{\lambda_3}{\Delta x^2}(u_3^{1^{(k+1)}} - u_{\Gamma_2}^k) \end{bmatrix}. \tag{4.60}$$

Again this gives us two equations:

$$A_{II}^{(2)}u_I^{(2)^{(k+1)}} + A_{I\Gamma_1}^{(2)^2}u_{\Gamma_2}^{(k+1)} = A_{II}^{(2)}u_I^{(2)^{(k+1)}} - \boldsymbol{e_n}\frac{\lambda_2}{\Delta x^2}u_{\Gamma_2}^{(k+1)} = \boldsymbol{f_2} + \boldsymbol{e_1}\frac{\lambda_2}{\Delta x^2}u_{\Gamma_1}^{(k+1)} \tag{4.61}$$

and

$$A_{\Gamma_1 I}^{(2)^2}u_I^{(2)^{(k+1)}} + \widetilde{A}_{\Gamma_2\Gamma_2}^{(2)}u_{\Gamma_2}^{(k+1)} = -\boldsymbol{e_n}^T\frac{\lambda_2}{\Delta x^2}u_I^{(2)^{(k+1)}} + \frac{\lambda_2}{\Delta x^2}u_{\Gamma_2}^{(k+1)} = \frac{\lambda_3}{\Delta x^2}(\boldsymbol{e_1}^T u_I^{(3)^{(k+1)}} - u_{\Gamma_2}^k). \tag{4.62}$$

Rewriting (4.61):

$$\boldsymbol{e_n}\frac{\lambda_2}{\Delta x^2}u_{\Gamma_2}^{(k+1)} = A_{II}^{(2)}u_I^{(2)^{(k+1)}} - \boldsymbol{f_2} - \boldsymbol{e_1}\frac{\lambda_2 u_{\Gamma_1}^{(k+1)}}{\Delta x^2}. \tag{4.63}$$

Rewriting (4.62):

$$\frac{\lambda_2}{\Delta x^2}u_{\Gamma_2}^{(k+1)} = \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T u_I^{(3)^{(k+1)}} - \frac{\lambda_3}{\Delta x^2}u_{\Gamma_2}^k + \boldsymbol{e_n}^T\frac{\lambda_2}{\Delta x^2}u_I^{(2)^{(k+1)}}. \tag{4.64}$$

We now rewrite equations (4.58) and (4.64) that we have obtained from the two systems and divide both sides with the respective $\frac{\lambda_i}{\Delta x^2}$:

$$u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\lambda_2}\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} - \frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k + \boldsymbol{e_1}^T u_I^{(2)^{(k+1/2)}} \tag{4.65}$$

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_3}{\lambda_2}\boldsymbol{e_1}^T u_I^{(3)^{(k+1)}} - \frac{\lambda_3}{\lambda_2}u_{\Gamma_2}^k + \boldsymbol{e_n}^T u_I^{(2)^{(k+1)}}. \tag{4.66}$$

We now write out the formulas for the interior points, (4.67) and (4.69) come from solving the system on Ω_1 and Ω_3 with Dirichlet conditions on both sides, (4.68) and (4.70) follow from (4.57) and (4.63).

$$u_I^{(1)^{(k+1)}} = A_{II}^{(1)^{-1}}(\boldsymbol{f_1} + \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_1}^k) \tag{4.67}$$

$$u_I^{(2)^{(k+1/2)}} = A_{II}^{(2)^{-1}}(\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^k) \tag{4.68}$$

$$u_I^{(3)^{(k+1)}} = A_{II}^{(3)^{-1}}(\boldsymbol{f_3} + \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_2}^k) =: A_{II}^{(3)^{-1}}\boldsymbol{b_3} \tag{4.69}$$

$$u_I^{(2)^{(k+1)}} = A_{II}^{(2)^{-1}}(\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^{(k+1)}). \tag{4.70}$$

Inserting into (4.65):

$$u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1}{\lambda_2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}(\boldsymbol{f_1}+\frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_1}^k)-\frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k+\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}(\boldsymbol{f_2}+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)}+\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^k)$$

$$= \frac{\lambda_1}{\lambda_2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{f_1} + \frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}u_{\Gamma_1}^k - \frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k + \boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{f_2}+$$
$$\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}u_{\Gamma_1}^{(k+1)} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k.$$

Moving the $u_{\Gamma_1}^{(k+1)}$ to one side and collecting terms not depending on earlier iterations into Φ:

$$(1 - \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1})u_{\Gamma_1}^{(k+1)} = \frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}u_{\Gamma_1}^k - \frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k + \Phi_1$$

where

$$\Phi_1 = \frac{\lambda_1}{\lambda_2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{f_1} + \boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{f_2}.$$

Dividing:

$$u_{\Gamma_1}^{(k+1)} = \frac{\frac{\lambda_1^2}{\lambda_2\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}u_{\Gamma_1}^k - \frac{\lambda_1}{\lambda_2}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k + \Phi_1}{1 - \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}} \tag{4.71}$$

here $A_{II}^{(1)^{-1}}$ and $A_{II}^{(2)^{-1}}$ are inverses of tridiagonal Toeplitz matrices as seen in (3.22).

As we later wish to represent the relation between $u_{\Gamma_i}^{k+1}$ and $u_{\Gamma_i}^{k}$ as a matrix, we separate those terms. The terms that do not depend on the boundary points are collected into $\widetilde{\Phi}_1$.

$$u_{\Gamma_1}^{(k+1)} = \frac{(\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2})}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} u_{\Gamma_1}^k + \frac{\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} u_{\Gamma_2}^k + \widetilde{\Phi}_1. \tag{4.72}$$

We now wish to find the analogous equation for $u_{\Gamma_2}^{(k+1)}$, for this we use equation (4.66):

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_3}{\lambda_2} \boldsymbol{e_1}^T u_I^{(3)^{(k+1)}} - \frac{\lambda_3}{\lambda_2} u_{\Gamma_2}^k + \boldsymbol{e_n}^T u_I^{(2)^{(k+1)}}. \tag{4.73}$$

Inserting the formulae for $u_I^{(3)^{(k+1)}}$ (4.69) and $u_I^{(2)^{(k+1)}}$ (4.70):

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_3}{\lambda_2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} (\boldsymbol{f_3} + \frac{\lambda_3}{\Delta x^2} \boldsymbol{e_1} u_{\Gamma_2}^k) - \frac{\lambda_3}{\lambda_2} u_{\Gamma_2}^k + \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} (\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1} u_{\Gamma_1}^{(k+1)} + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n} u_{\Gamma_2}^{(k+1)})$$

$$= \frac{\lambda_3}{\lambda_2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{f_3} + \frac{\lambda_3^2}{\lambda_2 \Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1} u_{\Gamma_2}^k - \frac{\lambda_3}{\lambda_2} u_{\Gamma_2}^k + \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{f_2} +$$
$$\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1} u_{\Gamma_1}^{(k+1)} + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n} u_{\Gamma_2}^{(k+1)}.$$

We collect all terms not dependent on the boundary values into Φ_2:

$$= \frac{\lambda_3^2}{\lambda_2 \Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1} u_{\Gamma_2}^k - \frac{\lambda_3}{\lambda_2} u_{\Gamma_2}^k + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1} u_{\Gamma_1}^{(k+1)} + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n} u_{\Gamma_2}^{(k+1)} + \Phi_2,$$

where:

$$\Phi_2 = \frac{\lambda_3}{\lambda_2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{f_3} + \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{f_2}. \tag{4.74}$$

Dividing:

$$u_{\Gamma_2}^{(k+1)} = \frac{\frac{\lambda_3^2}{\lambda_2 \Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1} u_{\Gamma_2}^k - \frac{\lambda_3}{\lambda_2} u_{\Gamma_2}^k + \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1} u_{\Gamma_1}^{(k+1)} + \Phi_2}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}}. \tag{4.75}$$

Separating, all terms not dependent on the boundary values are collected into $\widetilde{\Phi}_2$:

$$u_{\Gamma_2}^{(k+1)} = \frac{(\frac{\lambda_3^2}{\lambda_2 \Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1} - \frac{\lambda_3}{\lambda_2})}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}} u_{\Gamma_2}^k + \frac{\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}} u_{\Gamma_1}^{(k+1)} + \widetilde{\Phi}_2. \tag{4.76}$$

Now note: on the right hand side we have dependance on $u_{\Gamma_1}^{(k+1)}$. For simpler further calculations we introduce the following notation for (4.72) and (4.76):

$$u_{\Gamma_1}^{(k+1)} = a_1 u_{\Gamma_1}^k + b_1 u_{\Gamma_2}^k + \widetilde{\Phi}_1 \tag{4.77}$$

$$u_{\Gamma_2}^{(k+1)} = a_2 u_{\Gamma_1}^{(k+1)} + b_2 u_{\Gamma_2}^k + \widetilde{\Phi}_2. \tag{4.78}$$

We now insert the formula for $u_{\Gamma_1}^{(k+1)}$ into the second equation:

$$u_{\Gamma_1}^{(k+1)} = a_1 u_{\Gamma_1}^k + b_1 u_{\Gamma_2}^k + \widetilde{\Phi}_1 \tag{4.79}$$

$$u_{\Gamma_2}^{(k+1)} = a_2(a_1 u_{\Gamma_1}^k + b_1 u_{\Gamma_2}^k + \widetilde{\Phi}_1) + b_2 u_{\Gamma_2}^k + \widetilde{\Phi}_2 = a_1 a_2 u_{\Gamma_1}^k + a_2 b_1 u_{\Gamma_2}^k + b_2 u_{\Gamma_2}^k + a_2\widetilde{\Phi}_1 + \widetilde{\Phi}_2. \tag{4.80}$$

This allows us to write the system as a fixed point iteration of the form seen in (2.1):

$$\begin{bmatrix} u_{\Gamma_1}^{(k+1)} \\ u_{\Gamma_2}^{(k+1)} \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_1 a_2 & (a_2 b_1 + b_2) \end{bmatrix} \begin{bmatrix} u_{\Gamma_1}^k \\ u_{\Gamma_2}^k \end{bmatrix} + \begin{bmatrix} \widetilde{\Phi}_1 \\ a_2\widetilde{\Phi}_1 + \widetilde{\Phi}_2 \end{bmatrix} \tag{4.81}$$

where two coefficients can be directly read off those calculated for Method 1. From (4.46) we have:

$$a_1 = \frac{\frac{\lambda_1^2}{\lambda_2 \Delta x^2} \boldsymbol{e_n}^T A_{II}^{(1)^{-1}} \boldsymbol{e_n} - \frac{\lambda_1}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} = \frac{-\lambda_1}{\lambda_2} \tag{4.82}$$

and from (4.47):

$$b_1 = \frac{\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}} = 1. \tag{4.83}$$

The others are also obtained directly from earlier results, but require some additional rewriting.

$$\begin{aligned} a_2 = \frac{\frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_1}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}} &\overset{(3.36)(4.36)}{=} \frac{\Delta x \sum_{i=1}^n \frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}{1 - \Delta x \sum_{i=1}^n \frac{\sin^2\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}} \\ &\overset{(3.39)}{=} \frac{\Delta x \sum_{i=1}^n \frac{\sin\left(\frac{i\pi}{n+1}\right)\sin\left(\frac{in\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}}{1 - \Delta x \sum_{i=1}^n \frac{\sin^2\left(\frac{i\pi}{n+1}\right)}{1-\cos\left(\frac{i\pi}{n+1}\right)}} \overset{(4.45)}{=} 1. \end{aligned} \tag{4.84}$$

Finally;

$$b_2 = \frac{\frac{\lambda_3^2}{\lambda_2 \Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1} - \frac{\lambda_3}{\lambda_2}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}} = \frac{-\lambda_3}{\lambda_2} \frac{1 - \frac{\lambda_3}{\Delta x^2} \boldsymbol{e_1}^T A_{II}^{(3)^{-1}} \boldsymbol{e_1}}{1 - \frac{\lambda_2}{\Delta x^2} \boldsymbol{e_n}^T A_{II}^{(2)^{-1}} \boldsymbol{e_n}} =_{(4.49)} \frac{-\lambda_3}{\lambda_2}. \tag{4.85}$$

The eigenvalues, denoted σ_i, of the iteration matrix in (4.81) can then be expressed as:

$$\sigma_{1,2} = \frac{a_1 + (a_2 b_1 + b_2)}{2} \pm \sqrt{\left(\frac{a_1 + (a_2 b_1 + b_2)}{2}\right)^2 - a_1(a_2 b_1 + b_2) + a_1 a_2 b_1}. \tag{4.86}$$

We now insert the values into our system matrix:

$$\begin{bmatrix} \frac{-\lambda_1}{\lambda_2} & 1 \\ \frac{-\lambda_1}{\lambda_2} & 1 + \frac{-\lambda_3}{\lambda_2} \end{bmatrix} \tag{4.87}$$

and we have eigenvalues:

$$\sigma_{1,2} = \frac{\frac{-\lambda_1}{\lambda_2} + 1 + \frac{-\lambda_3}{\lambda_2}}{2} \pm \sqrt{\left(\frac{\frac{-\lambda_1}{\lambda_2} + 1 + \frac{-\lambda_3}{\lambda_2}}{2}\right)^2 - \frac{\lambda_1 \lambda_3}{\lambda_2^2}}. \tag{4.88}$$

Below is the pseudo-code for Method 2.

Algorithm 3 Method 2: Iteration given intial guess $(u_{\Gamma_1}^0, u_{\Gamma_2}^0)$ and endpoints u_{start}, u_{end}

flag=True
$(u_{\Gamma_1}, u_{\Gamma_2}) = (u_{\Gamma_1}^0, u_{\Gamma_2}^0)$
Construct A_1, A_3, as in (3.9) A_2^1, A_2^2 as in (4.54) and (4.60).
while flag=True **do**
    Construct b_1 as in Method 1
    Obtain $\mathbf{u_1}$ by solving $A_1 x = b_1$
    Construct b_2^1 as in (4.54)
    Obtain $\mathbf{u_2^1}$ by solving $A_2^1 x = b_2^1$
    $u_{\Gamma_1}^{old} = u_{\Gamma_1}$
    Extract new u_Γ from u_2
    Construct b_3 as in (4.69)
    Obtain $\mathbf{u_3}$ by solving $A_3 x = b_3$
    Construct b_2^2 as in (4.60)
    Obtain $\mathbf{u_2^2}$ by solving $A_2^2 x = b_2^2$
    $u_{\Gamma_2}^{old} = u_{\Gamma_2}$
    Extract new u_{Γ_2} from $\mathbf{u_2^2}$
    If $||(u_{\Gamma_1}^{old}, u_{\Gamma_2}^{old}) - (u_{\Gamma_1}, u_{\Gamma_2})|| < tol$ set flag=False
end while
return $(u_{start}, \mathbf{u_1}, \mathbf{u_2^2}, \mathbf{u_3}, u_{end})$ as solution.

4.3 Three-section split - Method 3

Finally we introduce a third algorithm, this method is built with the same "building blocks" we used in Method 2, the only difference being the location of the respective Dirichlet/Neumann conditions. One iteration step of Method 3 is presented in Fig. 4.4:

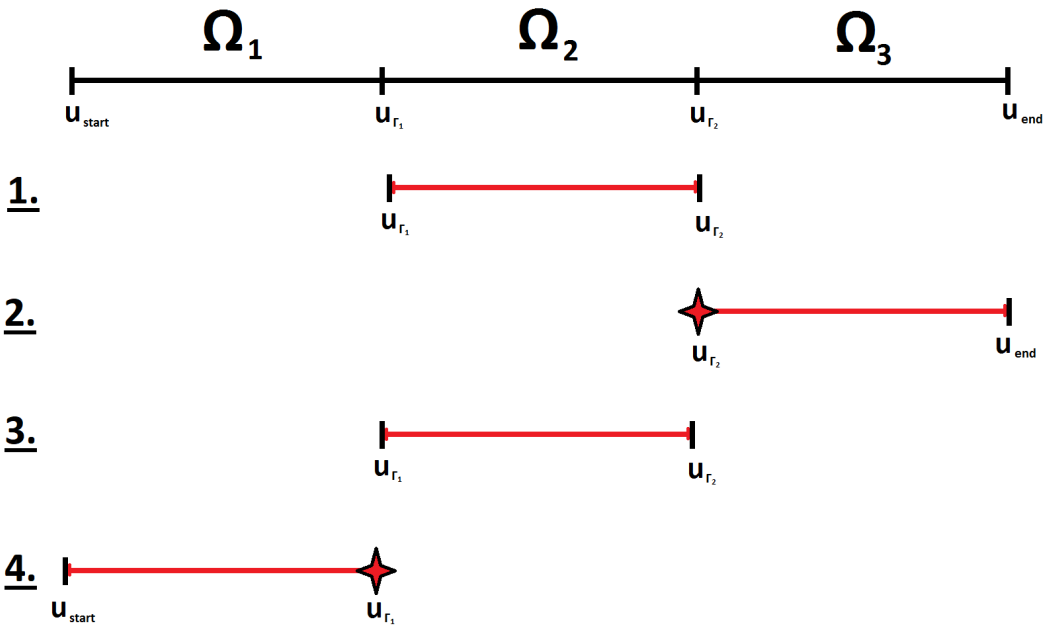

FIGURE 4.4: Method 3 - here we start by solving the middle part, using the initial guess as Dirichlet conditions. As before, star denotes Neumann condition, bar denotes Dirichlet condition

For example, step 1 in Method 3 is structurally step 1 in Method 2 but it is moved from Ω_1 to Ω_2, step 2 in Method 3 is structurally step 2 in Method 2 but it is moved from Ω_2 to Ω_3 and so on. Thus, for the sake of brevity we leave out the explicit construction as it is analogous to Method 2. We can further exploit this to significantly shorten our analysis of Method 3, as we already know the form of the needed equations from Method 2. We write out the formulas for the interior points:

$$u_I^{(2)^{(k+1/2)}} = A_{II}^{(2)^{-1}}(\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^k) \tag{4.89}$$

$$u_I^{(3)^{(k+1)}} = A_{II}^{(3)^{-1}}(\boldsymbol{f_3} + \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_2}^{(k+1)}) \tag{4.90}$$

$$u_I^{(2)^{(k+1)}} = A_{II}^{(2)^{-1}}(\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^{(k+1)}) \tag{4.91}$$

$$u_I^{(1)^{(k+1)}} = A_{II}^{(1)^{-1}}(\boldsymbol{f_1} + \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_1}^{(k+1)}). \tag{4.92}$$

For the boundaries we then have, analogous to (4.64) and (4.58):

$$-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T u_I^{(3)^{(k+1)}} + \frac{\lambda_3}{\Delta x^2}u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2}{\Delta x^2}(\boldsymbol{e_n}^T u_I^{(2)^{(k+1/2)}} - u_{\Gamma_2}^k) \tag{4.93}$$

and

$$-\frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} + \frac{\lambda_1}{\Delta x^2}u_{\Gamma_1}^{(k+1)} = \frac{\lambda_2}{\Delta x^2}(\boldsymbol{e_1}^T u_I^{(2)^{(k+1)}} - u_{\Gamma_1}^k). \tag{4.94}$$

Rewriting and inserting the formulas for the interior points:

$$-\frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T u_I^{(3)^{(k+1)}} + \frac{\lambda_3}{\Delta x^2}u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2}{\Delta x^2}(\boldsymbol{e_n}^T u_I^{(2)^{(k+1/2)}} - u_{\Gamma_2}^k) \tag{4.95}$$

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2}{\lambda_3}(\boldsymbol{e_n}^T u_I^{(2)^{(k+1/2)}} - u_{\Gamma_2}^k) + \boldsymbol{e_1}^T u_I^{(3)^{(k+1)}} \tag{4.96}$$

$$= \frac{\lambda_2}{\lambda_3}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}(\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^k) - \frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k + \boldsymbol{e_1}^T u_I^{(3)^{(k+1)}}.$$

Collecting the terms not dependent on the boundary points into Φ_2:

$$= \frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}u_{\Gamma_1}^k + \frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k - \frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k + \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}u_{\Gamma_2}^{(k+1)} + \Phi_2.$$

Moving all $u_{\Gamma_2}^{(k+1)}$ terms to one side:

$$u_{\Gamma_2}^{(k+1)}(1 - \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}) = \frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}u_{\Gamma_1}^k + \frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^k - \frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k + \Phi_2.$$

Dividing and collecting all terms not dependent on boundary values into $\widetilde{\Phi}_2$:

$$u_{\Gamma_2}^{(k+1)} = \frac{\frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}}{1 - \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}}u_{\Gamma_1}^k + \frac{(\frac{\lambda_2^2}{\lambda_3\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n} - \frac{\lambda_2}{\lambda_3})}{1 - \frac{\lambda_3}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(3)^{-1}}\boldsymbol{e_1}}u_{\Gamma_2}^k + \widetilde{\Phi}_2.$$

Using the values of Method 1, specifically (4.48) and (4.46), we get:

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2}{\lambda_3}u_{\Gamma_1}^k - \frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k + \widetilde{\Phi}_2. \tag{4.97}$$

Moving on to the Γ_1 boundary, inserting (4.91) and (4.92) into (4.94):

$$-\frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}^T u_I^{(1)^{(k+1)}} + \frac{\lambda_1}{\Delta x^2}u_{\Gamma_1}^{(k+1)} = \frac{\lambda_2}{\Delta x^2}(\boldsymbol{e_1}^T u_I^{(2)^{(k+1)}} - u_{\Gamma_1}^k) \tag{4.98}$$

$$u_{\Gamma_1}^{(k+1)} = \frac{\lambda_2}{\lambda_1}\boldsymbol{e_1}^T u_I^{(2)^{(k+1)}} - \frac{\lambda_2}{\lambda_1}u_{\Gamma_1}^k + \boldsymbol{e_n}^T u_I^{(1)^{(k+1)}}$$

$$= \frac{\lambda_2}{\lambda_1}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}(\boldsymbol{f_2} + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}u_{\Gamma_1}^k + \frac{\lambda_2}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_2}^{(k+1)}) - \frac{\lambda_2}{\lambda_1}u_{\Gamma_1}^k + \boldsymbol{e_n}^T A_{II}^{(1)^{-1}}(\boldsymbol{f_1} + \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}u_{\Gamma_1}^{(k+1)}).$$

Collecting all terms not dependent on the boundary values into Φ_1:

$$= \frac{\lambda_2^2}{\lambda_1\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1}u_{\Gamma_1}^k + \frac{\lambda_2^2}{\lambda_1\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^{(k+1)} - \frac{\lambda_2}{\lambda_1}u_{\Gamma_1}^k + \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}u_{\Gamma_1}^{(k+1)} + \Phi_1.$$

Moving all $u_{\Gamma_1}^{(k+1)}$ terms to one side:

$$u_{\Gamma_1}^{(k+1)}(1 - \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}) = \frac{\lambda_2}{\lambda_1}(\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1} - 1)u_{\Gamma_1}^k + \frac{\lambda_2^2}{\lambda_1\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}u_{\Gamma_2}^{(k+1)} + \Phi_1.$$

Dividing, collecting all terms not dependent on the boundary values into $\widetilde{\Phi}_1$

$$u_{\Gamma_1}^{(k+1)} = \frac{\lambda_2}{\lambda_1}\frac{(\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_1} - 1)}{1 - \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}}u_{\Gamma_1}^k + \frac{\lambda_2}{\lambda_1}\frac{\frac{\lambda_2}{\Delta x^2}\boldsymbol{e_1}^T A_{II}^{(2)^{-1}}\boldsymbol{e_n}}{1 - \frac{\lambda_1}{\Delta x^2}\boldsymbol{e_n}^T A_{II}^{(1)^{-1}}\boldsymbol{e_n}}u_{\Gamma_2}^{(k+1)} + \widetilde{\Phi}_1.$$

Again, using the formula from Method 2, specifically (4.85) and (4.84) where we use the symmetry property (4.36) for the second term:

$$u_{\Gamma_1}^{(k+1)} = -\frac{\lambda_2}{\lambda_1}u_{\Gamma_1}^k + \frac{\lambda_2}{\lambda_1}u_{\Gamma_2}^{(k+1)} + \widetilde{\Phi}_1. \tag{4.99}$$

Inserting (4.97) into (4.99):

$$u_{\Gamma_1}^{(k+1)} = -\frac{\lambda_2}{\lambda_1}u_{\Gamma_1}^k + \frac{\lambda_2}{\lambda_1}(\frac{\lambda_2}{\lambda_3}u_{\Gamma_1}^k - \frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k + \widetilde{\Phi}_2) + \widetilde{\Phi}_1. \tag{4.100}$$

We thus have:

$$u_{\Gamma_1}^{(k+1)} = -\frac{\lambda_2}{\lambda_1}u_{\Gamma_1}^k + \frac{\lambda_2^2}{\lambda_1\lambda_3}u_{\Gamma_1}^k - \frac{\lambda_2^2}{\lambda_1\lambda_3}u_{\Gamma_2}^k + \widetilde{\Phi}_2 + \widetilde{\Phi}_1 \tag{4.101}$$

$$u_{\Gamma_2}^{(k+1)} = \frac{\lambda_2}{\lambda_3}u_{\Gamma_1}^k - \frac{\lambda_2}{\lambda_3}u_{\Gamma_2}^k + \widetilde{\Phi}_2 \tag{4.102}$$

which gives us the fixed point iteration:

$$\begin{bmatrix} u_{\Gamma_1}^{(k+1)} \\ u_{\Gamma_2}^{(k+1)} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda_2}{\lambda_1} + \frac{\lambda_2^2}{\lambda_1\lambda_3} & -\frac{\lambda_2^2}{\lambda_1\lambda_3} \\ +\frac{\lambda_2}{\lambda_1} & -\frac{\lambda_2}{\lambda_1} \end{bmatrix} \begin{bmatrix} u_{\Gamma_1}^{k} \\ u_{\Gamma_2}^{k} \end{bmatrix} + \begin{bmatrix} \widetilde{\Phi}_1 + \widetilde{\Phi}_2 \\ \widetilde{\Phi}_2 \end{bmatrix} \tag{4.103}$$

where the matrix has the eigenvalues:

$$\sigma_{1,2} = \frac{\lambda_2}{2\lambda_1\lambda_3}(-\lambda_1 + \lambda_2 - \lambda_3 \pm \sqrt{(\lambda_1 - \lambda_2 + \lambda_3)^2 - 4\lambda_1\lambda_3}). \tag{4.104}$$

Below is the pseudo-code for Method 3. Note that the equations referenced give the structure of the matrices and vectors, while the values have to be changed to account for the changed positioning.

Algorithm 4 Method 3: Iteration given intial guess $(u_{\Gamma_1}^0, u_{\Gamma_2}^0)$ and endpoints u_{start}, u_{end}

flag=True
$(u_{\Gamma_1}, u_{\Gamma_2}) = (u_{\Gamma_1}^0, u_{\Gamma_2}^0)$
Construct A_2 analogously to (3.9), A_3 and A_1 analogously to (4.54) and (4.60).
while flag=True **do**
    Construct b_2^1 analogously to b_1 in Method 1
    Obtain $\mathbf{u_2^1}$ by solving $A_2 x = b_2^1$
    Construct b_3 analogously to b_2^1 in (4.54)
    Obtain $\mathbf{u_3}$ by solving $A_3 x = b_3$
    $u_{\Gamma_2}^{old} = u_{\Gamma_2}$
    Extract new u_{Γ_2} from u_3
    Construct b_2^2 analogously to b_1 in Method 1
    Obtain $\mathbf{u_2^2}$ by solving $A_2 x = b_2^2$
    Construct b_1 analogously to b_2^2 as in (4.60)
    Obtain $\mathbf{u_1}$ by solving $A_1 x = b_1$
    $u_{\Gamma_1}^{old} = u_{\Gamma_1}$
    Extract new u_{Γ_1} from $\mathbf{u_1}$
    If $||(u_{\Gamma_1}^{old}, u_{\Gamma_2}^{old}) - (u_{\Gamma_1}, u_{\Gamma_2})|| < tol$ set flag=False
end while
return $(u_{start}, \mathbf{u_1}, \mathbf{u_2^2}, \mathbf{u_3}, u_{end})$ as solution.

4.4 Summary - comparing the three methods

We now have explicit formulae for the spectral radii of the fixed point iterations of all three methods. The spectral radius associated with Method 1 is given by (4.51):

$$\max|\sigma_{1,2}^{M1}| = \max\left| -\frac{1}{2}\frac{\lambda_1}{\lambda_2} \pm \sqrt{\frac{1}{4}\frac{\lambda_1^2}{\lambda_2^2} - \frac{\lambda_1}{\lambda_3}}\right|. \tag{4.105}$$

The spectral radius associated with Method 2 is given by (4.88):

$$\max|\sigma_{1,2}^{M2}| = \max\left|\frac{\frac{-\lambda_1}{\lambda_2} + 1 + \frac{-\lambda_3}{\lambda_2}}{2} \pm \sqrt{\left(\frac{\frac{-\lambda_1}{\lambda_2} + 1 + \frac{-\lambda_3}{\lambda_2}}{2}\right)^2 - \frac{\lambda_1\lambda_3}{\lambda_2^2}}\right|, \tag{4.106}$$

finally, spectral radius associated with Method 3 is given by (4.104):

$$\max|\sigma_{1,2}^{M3}| = \max\left|\frac{\lambda_2}{2\lambda_1\lambda_3}(-\lambda_1 + \lambda_2 - \lambda_3 \pm \sqrt{(\lambda_1 - \lambda_2 + \lambda_3)^2 - 4\lambda_1\lambda_3})\right|. \tag{4.107}$$

We now wish to examine closer where the respective methods are convergent. To this end, we consider some limit cases of the formulae for the spectral radii.

Method 1 (4.51) is efficient when $\lambda_1 \ll \lambda_2$ and $\lambda_1 \ll \lambda_3$;

$$\max|\sigma_{1,2}^{M1}| = \max\left| -\frac{1}{2}\frac{\lambda_1}{\lambda_2} \pm \sqrt{\frac{1}{4}\frac{\lambda_1^2}{\lambda_2^2} - \frac{\lambda_1}{\lambda_3}}\right| \to 0 \ ; \ \text{when } \frac{\lambda_1}{\lambda_2} \to 0 \text{ and } \frac{\lambda_1}{\lambda_3} \to 0. \tag{4.108}$$

Method 2 (4.88) is efficient when $\lambda_1 \approx \lambda_2$ and $\lambda_3 \ll \lambda_2$ but equally so when $\lambda_2 \approx \lambda_3$ and $\lambda_1 \ll \lambda_2$;

$$\max|\sigma_{1,2}^{M2}| = \max\left|\frac{\frac{-\lambda_1}{\lambda_2} + 1 + \frac{-\lambda_3}{\lambda_2}}{2} \pm \sqrt{\left(\frac{\frac{-\lambda_1}{\lambda_2} + 1 + \frac{-\lambda_3}{\lambda_2}}{2}\right)^2 - \frac{\lambda_1\lambda_3}{\lambda_2^2}}\right| \to 0 \ ; \tag{4.109}$$

$$\text{when either } \left(\frac{\lambda_1}{\lambda_2} \to 1 \text{ and } \frac{\lambda_3}{\lambda_2} \to 0\right) \text{ or } \left(\frac{\lambda_3}{\lambda_2} \to 1 \text{ and } \frac{\lambda_1}{\lambda_2} \to 0\right).$$

Method 3 (4.104) is efficient when $\lambda_2 \ll \lambda_1$ and $\lambda_2 \ll \lambda_3$;

$$\max|\sigma_{1,2}^{M3}| = \max\left|\frac{\lambda_2}{2\lambda_1\lambda_3}(-\lambda_1 + \lambda_2 - \lambda_3 \pm \sqrt{(\lambda_1 - \lambda_2 + \lambda_3)^2 - 4\lambda_1\lambda_3})\right| \to 0 \ ; \tag{4.110}$$

$$\text{when } \left(\frac{\lambda_2}{2\lambda_1} \to 0\right) \text{ and } \left(\frac{\lambda_2}{2\lambda_3} \to 0\right).$$

4.5 Whole domain solver

As in the two field case, we construct a direct solver so that we may compare the results to that of the iterations. It solves (4.1) directly and the system is given by:

$$A = \frac{1}{\Delta x^2}\begin{bmatrix}
2\lambda_1 & -\lambda_1 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\
-\lambda_1 & 2\lambda_1 & -\lambda_1 & \ddots & & & & & & & & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & & & & & & & \vdots \\
\vdots & \ddots & -\lambda_1 & 2\lambda_1 & -\lambda_1 & \ddots & & & & & & \vdots \\
\vdots & & \ddots & -\lambda_1 & \lambda_1+\lambda_2 & -\lambda_2 & \ddots & & & & & \vdots \\
\vdots & & & \ddots & -\lambda_2 & 2\lambda_2 & -\lambda_2 & \ddots & & & & \vdots \\
\vdots & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\
\vdots & & & & & \ddots & -\lambda_2 & 2\lambda_2 & -\lambda_2 & \ddots & & \vdots \\
\vdots & & & & & & \ddots & -\lambda_2 & \lambda_2+\lambda_3 & -\lambda_3 & \ddots & \vdots \\
\vdots & & & & & & & \ddots & -\lambda_3 & 2\lambda_3 & -\lambda_3 & 0 \\
\vdots & & & & & & & & \ddots & \ddots & \ddots & \ddots \\
0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 & -\lambda_3 & 2\lambda_3
\end{bmatrix}$$

and

$$b_1 = \left[f_1^1 + \tfrac{u_{start}}{dx^2},\ f_1^2,\ \dots\ f_1^n,\ f_{\Gamma_1},\ f_2^1,\ \dots\ f_2^n,\ f_{\Gamma_2},\ f_3^1,\ \dots\ f_3^{n-1},\ f_3^n + \tfrac{u_{end}}{dx^2}\right]^T$$

with $A \in \mathbb{R}^{(3n+2)\times(3n+2)}$ and $b \in \mathbb{R}^{3n+2}$. Here u_{start} and u_{end} denote the start- and end-point values, respectively.

Chapter 5

Numerical results

The aim of the first part of this chapter is to confirm that the formulas found analytically for the spectral radii do indeed predict the rate of convergence for each of the three algorithms discussed. To this end we will present four sets of examples for each of the three algorithms. Our measure of success will be if the behavior of the error is parallel to a power-function of the spectral radius (see Theorem 2.2). Specifically we want to see:

$$C|\sigma|^{iteration} \quad \text{be parallel to} \quad ||\boldsymbol{u}_{direct} - \boldsymbol{u}_{iteration}||_2. \tag{5.1}$$

Where C is an arbitrary constant, $|\sigma|$ is the spectral radius, $\boldsymbol{u}_{direct}$ denotes the solution obtained by the whole domain solver and $\boldsymbol{u}_{iteration}$ is the solution given at the current iteration.

To illustrate the behavior of the different solvers we will also plot the solutions obtained from the Dirichlet-Neumann iterations (denoted: Iterative) together with the analytical solutions as in 5.1.1 and 5.2.1 below (Analytical), and the solutions obtained from the full system solvers in 3.2 and 4.5 (Direct).

5.1 Two field domain

We wish to numerically test if the spectral radius given in (3.41) provides an accurate estimator for the convergence rates. Toward this purpose we define a test problem and find its solution analytically.

5.1.1 Test problem and analytical solution

We consider a system as depicted in Fig. 2.1, with $\Omega_1 = [0,1], \Omega_2 = [1,2], \Gamma = \Omega_1 \cap \Omega_2$:

$$\begin{cases} -\Delta u_m(x) = \frac{\pi^2}{\lambda_m} \sin(\pi x) \; ; x \in \Omega_m \subset \mathbb{R} \; ; \; m = 1,2 \\ u_m(x) = 0 \; ; \; u_m(x) \in \partial\Omega_m \backslash \Gamma \\ u_1(x) = u_2(x) \; ; \; x \in \Gamma \\ \lambda_1 \frac{\partial u_1(x)}{\partial x} = \lambda_2 \frac{\partial u_2(x)}{\partial x} \; ; \; x \in \Gamma. \end{cases} \tag{5.2}$$

The equation $-\Delta u(x) = \frac{\pi^2}{\lambda} \sin(\pi x)$ has the general solution:

$$u(x) = \frac{\sin(\pi x)}{\lambda} + C_2 x + C_1 \tag{5.3}$$

with:

$$u'(x) = \frac{\pi \cos(\pi x)}{\lambda} + C_2. \tag{5.4}$$

With different values for λ, we have to fulfill boundary conditions as well as continuity in u_Γ and the coupling condition for the first derivate in u_Γ.

$$u_1(0) = 0 \tag{5.5}$$

$$u_1(1) = u_2(1) \tag{5.6}$$

$$\lambda_1 u_1'(1) = \lambda_2 u_2'(1) \tag{5.7}$$

$$u_2(2) = 0. \tag{5.8}$$

Solving the resulting linear system gives us the general solution

$$u_1(x) = \frac{\sin(\pi x)}{\lambda_1} \tag{5.9}$$

$$u_2(x) = \frac{\sin(\pi x)}{\lambda_2}. \tag{5.10}$$

This gives us an exact form which we can compare against numerical results for different values of λ_1 and λ_2.

5.1.2 Numerical results

In the tests below the following values are held constant: $u_{start} = u_{end} = 0$ and $u_\Gamma^0 = 0.5$. We begin by looking at the case where $\lambda_1 = \lambda_2 = 1$, i.e., we predict a spectral radius of 1, and thus we expect to see no convergence.

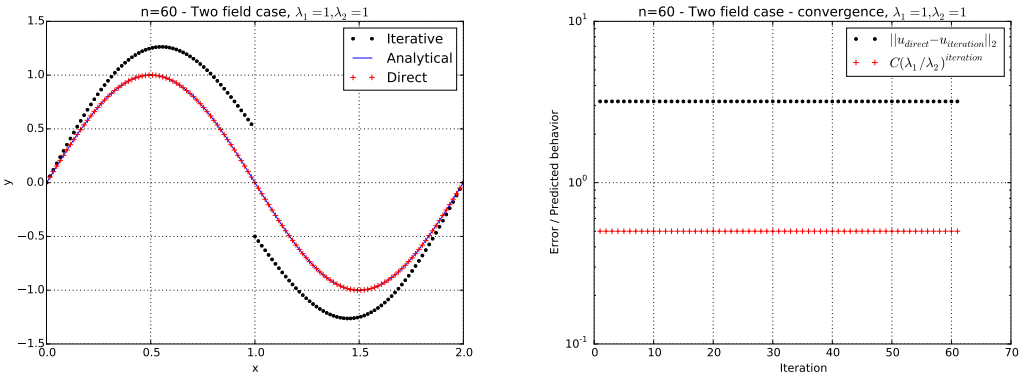

FIGURE 5.1: Two-field case – $\lambda_1 = 1, \lambda_2 = 1$

Indeed, in Fig. 5.1 (left) we that the iterative solution has not converged onto the direct solution. We also see a flat behavior of the error on the convergence plot on the right. We now look at the case where $\lambda_1 = 2, \lambda_2 = 4$, here we expect convergence as the spectral radius is now 0.5: Fig. 5.2

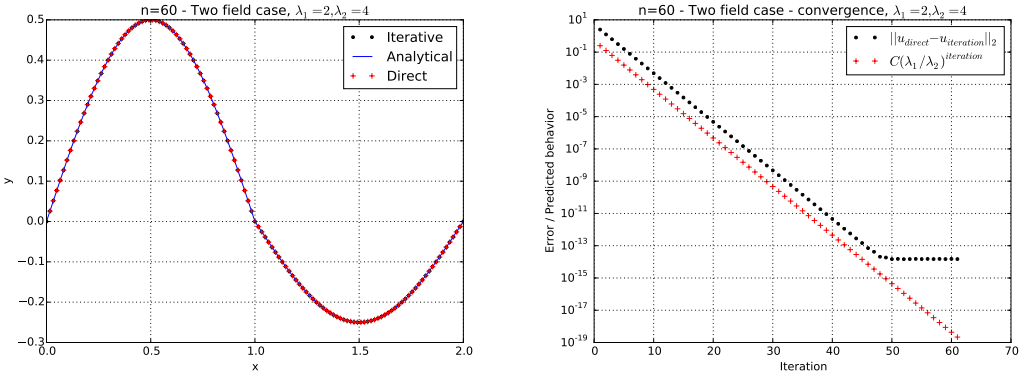

FIGURE 5.2: Two-field case – $\lambda_1 = 2, \lambda_2 = 4$

As expected, we see a solution which matches the complete domain solver and the convergence behavior is as predicted. We also note that the error reaches its numerical lower limit around iteration 49 after which it levels out. We now look at an example where we predict convergence but change the relation between λ_1 and λ_2 as well as the discretization Δx to convince us that the prediction still holds true. Different discretizations, both larger and smaller were tried, without influencing the rate of convergence, an example for $n = 20$ is pictured in Fig. 5.3, where $\lambda_1 = 3$ and $\lambda_3 = 8$.

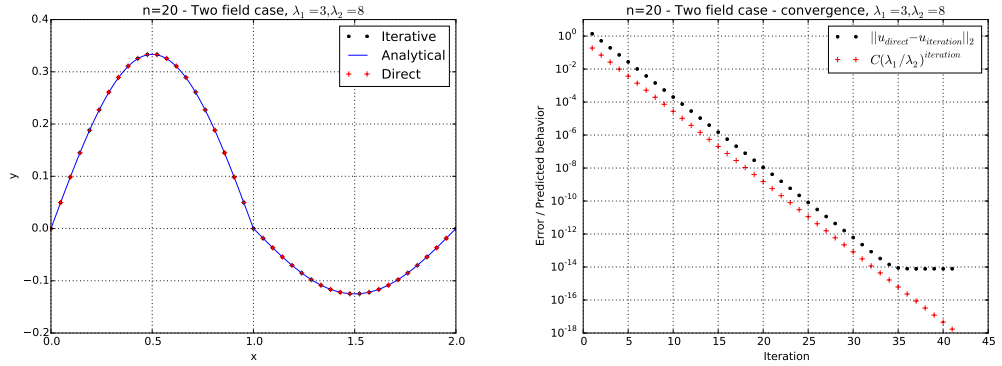

FIGURE 5.3: Two-field case – $\lambda_1 = 3, \lambda_2 = 8$

As in the previous case, all solutions and predictions match. We see that the error reaches its numerical lower limit, this time around iteration 34. Finally we consider a case where we do not predict convergence at all, namely $\lambda_1 = 1.1, \lambda_2 = 1$, Fig. 5.4:

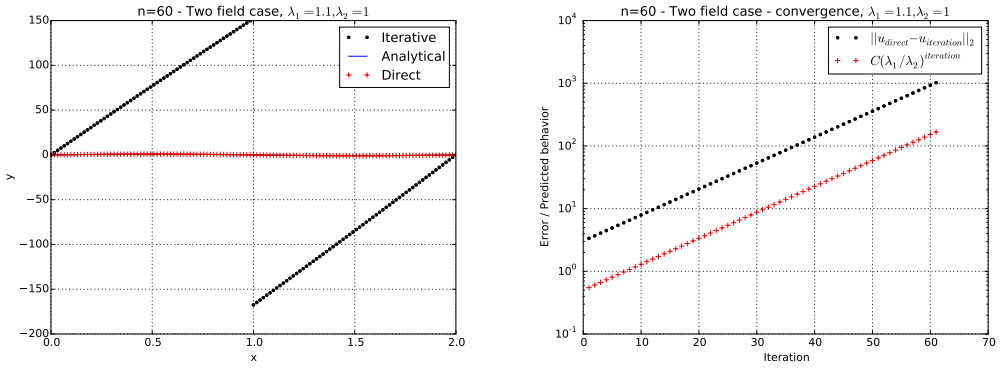

FIGURE 5.4: Two-field case – $\lambda_1 = 1.1, \lambda_2 = 1$

In the left frame we see that the solution has "exploded". We also note that the rate by which the error grows is accurately predicted by the spectral radius.

5.2 Three field domain

Similar to the two field case, we test if the spectral radii given by (4.51), (4.88) and (4.104) provide accurate estimators for the convergence rates.

5.2.1 Test problem and analytical solution

As before we consider:

$$-\Delta u = \frac{\pi^2}{\lambda}\sin(\pi x). \tag{5.11}$$

This has the general solution:

$$u(x) = \frac{\sin(\pi x)}{\lambda} + C_2 x + C_1 \tag{5.12}$$

with:

$$u'(x) = \frac{\pi \cos(\pi x)}{\lambda} + C_2. \tag{5.13}$$

We now extend the function from two to three domains (as depicted in Fig.4.1), $\Omega_1 = [0, 1), \Omega_2 = [1, 2], \Omega_3 = (2, 3]$ with different values for λ, we have to fulfill boundary conditions as well as continuity in u_Γ and the coupling condition of the first derivate in u_{Γ_1} and u_{Γ_2}.

$$u_1(0) = 0$$

$$u_1(1) = u_2(1)$$

$$\lambda_1 u_1'(1) = \lambda_2 u_2'(1)$$

$$u_2(2) = u_3(2)$$

$$\lambda_2 u_2'(2) = \lambda_3 u_3'(2)$$

$$u_3(3) = 0.$$

Solving the resulting linear system gives us the general solution

$$u_1(x) = \frac{\sin(\pi x)}{\lambda_1} \tag{5.14}$$

$$u_2(x) = \frac{\sin(\pi x)}{\lambda_2} \tag{5.15}$$

$$u_3(x) = \frac{\sin(\pi x)}{\lambda_3}. \tag{5.16}$$

This gives us an exact form which we can compare against numerical results for different values of λ_1, λ_2 and λ_3.

5.2.2 Method 1

The initial guess is constant across all examples with $(u^0_{\Gamma_1}, u^0_{\Gamma_2}) = (1, 2)$. We begin by looking at the case where all $\lambda = 1$, where we predict neither convergence nor divergence, Fig. 5.5:

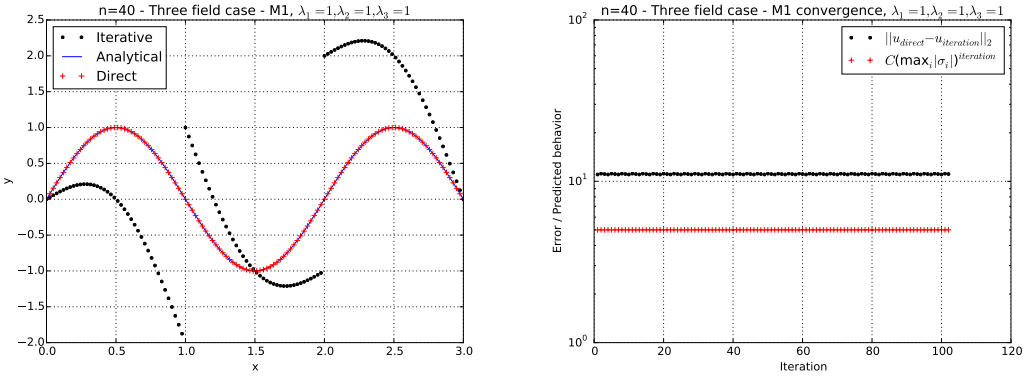

FIGURE 5.5: Three-field case, Method 1 – $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1, |\sigma_{max}| = 1$

On the left side we see that the iterative solution does not align with the direct solution. On the right side we see a flat behavior of the error, consistent with a spectral radius of one. We now move on to a case where we expect convergence.

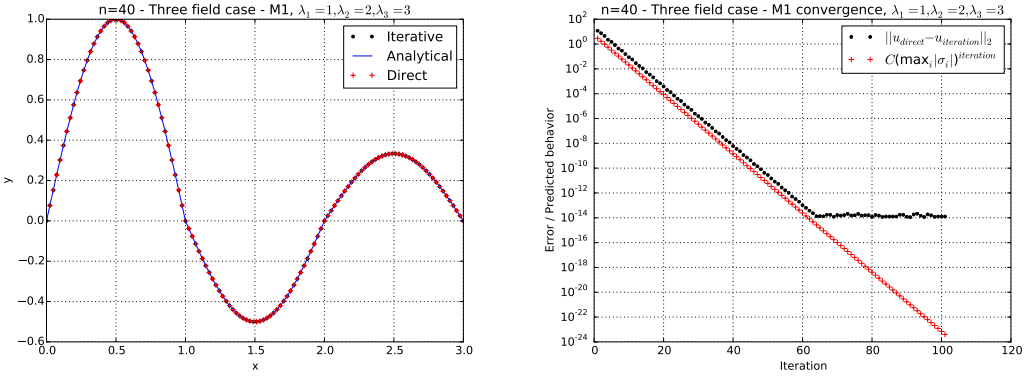

FIGURE 5.6: Three-field case, Method 1 – $\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 3, |\sigma_{max}| \approx 0.5771$

On the left side of Fig. 5.6 (where $\lambda_1 = 1, \lambda_2 = 2$, and $\lambda_3 = 3$) we see a complete solution matching the direct solver. On the right we see that the error follows the predicted behavior until iteration 64 were it levels out at around 10^{-14}. We now look at another example where we predict convergence but at a significantly faster rate.

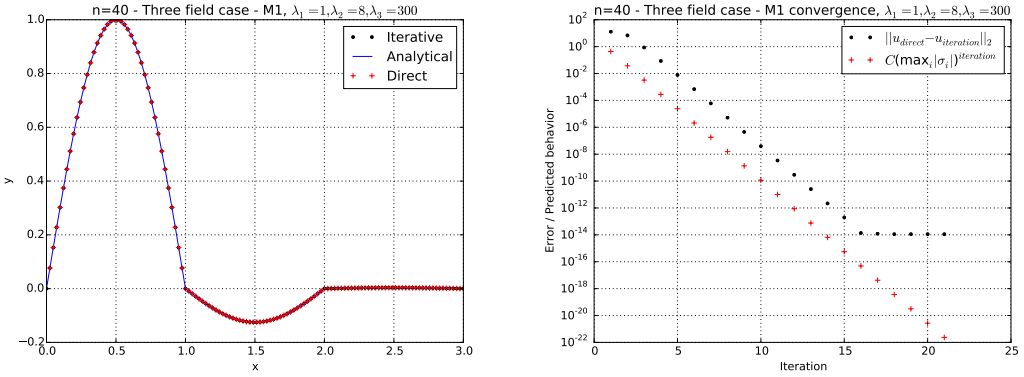

FIGURE 5.7: Three-field case, Method 1 – $\lambda_1 = 1, \lambda_2 = 8, \lambda_3 = 300, |\sigma_{max}| \approx 0.0864$

Fig. 5.7 (where $\lambda_1 = 1, \lambda_2 = 8$, and $\lambda_3 = 300$) we see a complete solution matching the whole domain solver with the iteration reaching numerical limit around iteration 16. The behavior of the error is accurately predicted by the spectral radius. Finally we consider a case where we predict divergence.

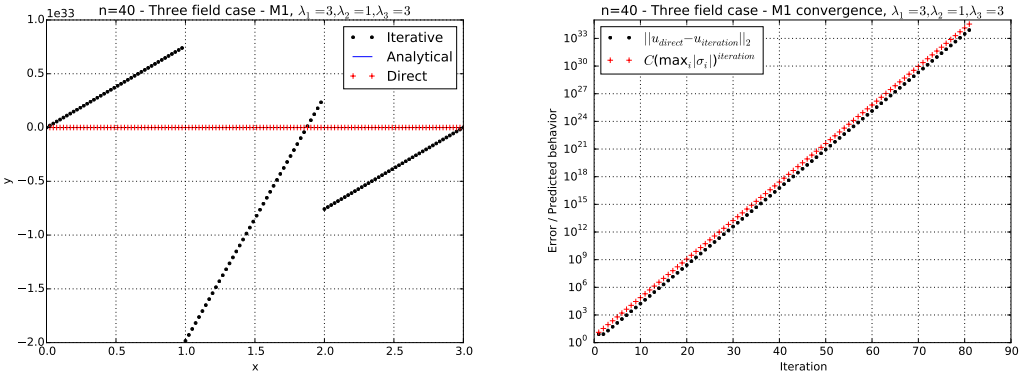

FIGURE 5.8: Three-field case, Method 1 – $\lambda_1 = 3, \lambda_2 = 1, \lambda_3 = 3, |\sigma_{max}| \approx 2.6180$

On the left side of Fig. 5.8 (where $\lambda_1 = 3, \lambda_2 = 1$, and $\lambda_3 = 3$) we see that solution has "exploded" and on the right that the rate by which the error grows is accurately predicted by the spectral radius.

5.2.3 Method 2

The initial guess is constant across all examples with $(u^0_{\Gamma_1}, u^0_{\Gamma_2}) = (1, -1)$. We begin by looking at the case where all $\lambda = 1$ and we have a spectral radius of 1, Fig. 5.9:

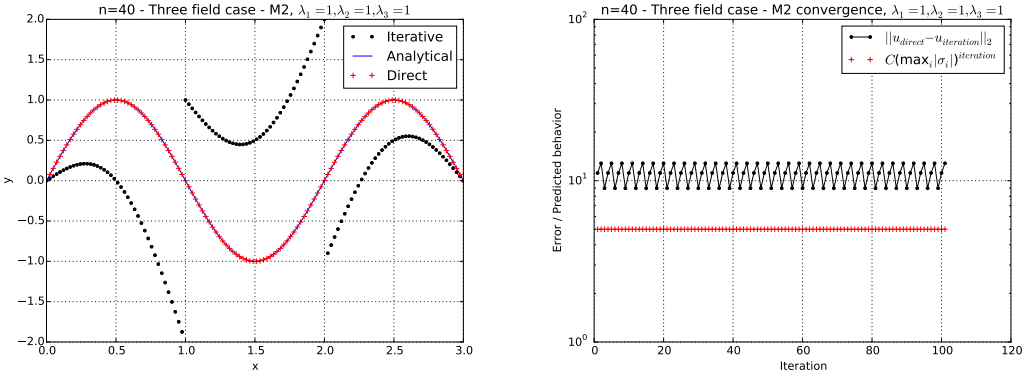

FIGURE 5.9: Three-field case, Method 2 – $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1, |\sigma_{max}| = 1$

We see that the iterative solution does not match that of the direct solver and that the error behavior is flat. We do observe a "sawtooth" pattern consisting of three repeating points but the trend is accurately predicted by the spectral radius. We now look at a case where we predict convergence. Fig. 5.10 illustrates the case where $\lambda_1 = 1, \lambda_2 = 2$ and $\lambda_3 = 3$.

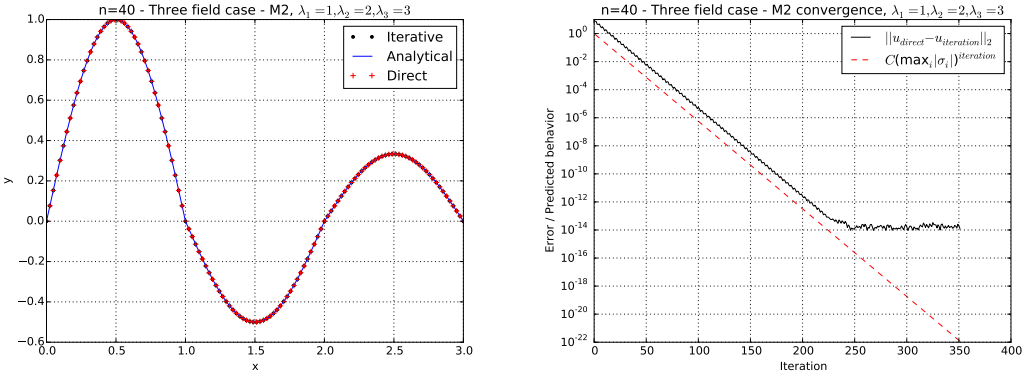

FIGURE 5.10: Three-field case, Method 2 – $\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 3, |\sigma_{max}| \approx 0.8660$

We see that the solution matches the whole domain solver and that the error behavior is accurately predicted by the spectral radius. We also note that we still see a "sawtooth" pattern in the error. Next we look at another case where convergence is predicted but at a faster rate. Fig. 5.11 shows the case with $\lambda_1 = 1, \lambda_2 = 50$ and $\lambda_3 = 60$.

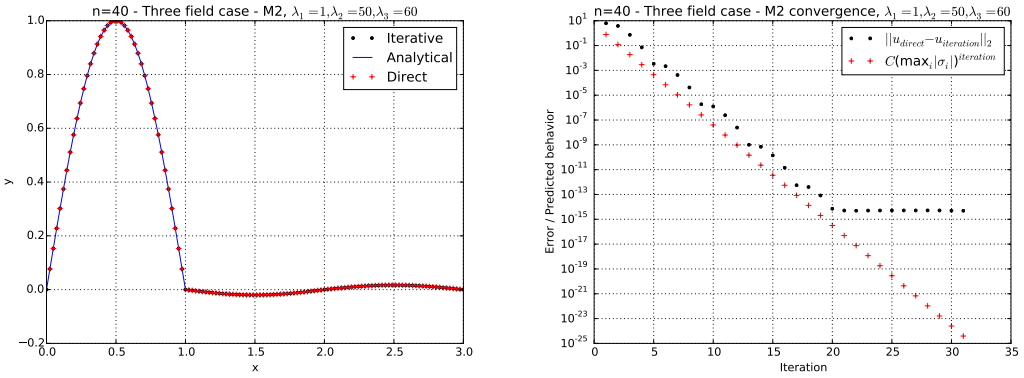

FIGURE 5.11: Three-field case, Method 2 – $\lambda_1 = 1, \lambda_2 = 50, \lambda_3 = 60, |\sigma_{max}| \approx 0.1549$

Again we see that the iteration solution matches the whole domain solver and that the error follows the power function of the spectral radius. We still see some "sawtooth" behavior of the error function but the trend follows the prediction. Finally we look at a case where we predict divergence; Fig. 5.12 ($\lambda_1 = 2, \lambda_2 = 3, \lambda_3 = 10$):

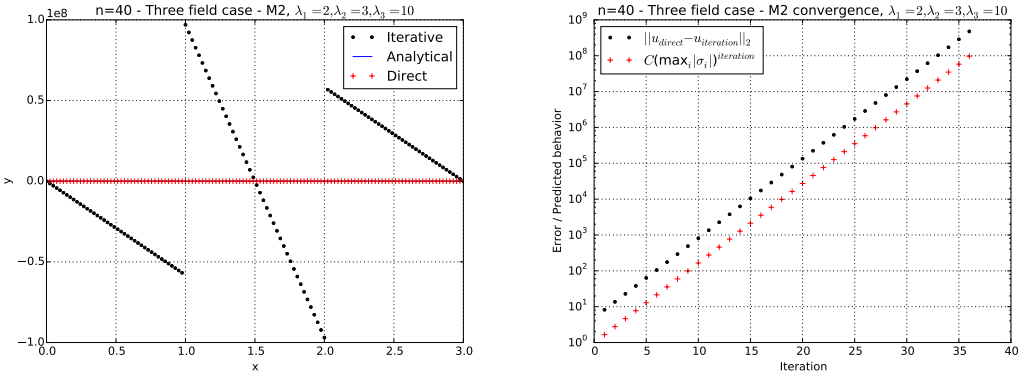

FIGURE 5.12: Three-field case, Method 2 – $\lambda_1 = 2, \lambda_2 = 3, \lambda_3 = 10, |\sigma_{max}| \approx 1.6667$

We that the solution has "exploded" and that the growth of the error is accurately predicted by the spectral radius.

5.2.4 Method 3

The initial guess is constant across all examples with $(u^0_{\Gamma_1}, u^0_{\Gamma_2}) = (1, -1)$. Again we begin by looking at the case where all $\lambda = 1$ and the spectral radius equals 1, Fig. 5.13:

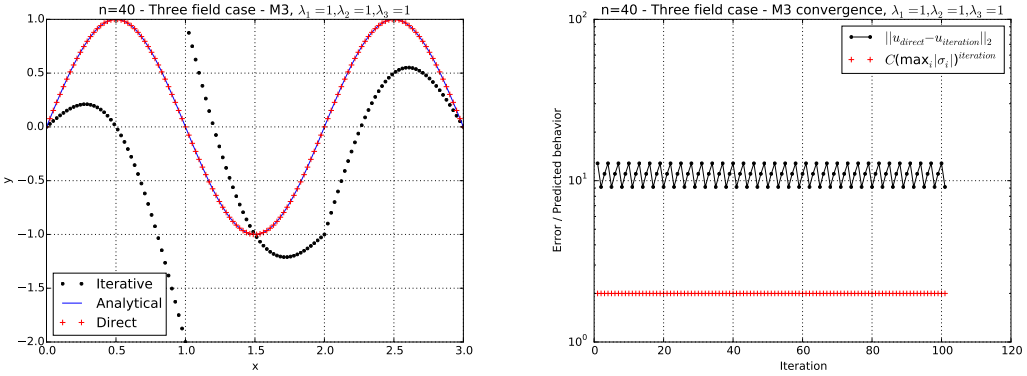

FIGURE 5.13: Three-field case, Method 3 – $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1, |\sigma_{max}| = 1$

As predicted, we see a flat behavior of the error and non-matching iterative and direct solutions. We also note that while this behavior is identical to that observed in the first example of Method 2, the plot is not. That is to say, we can see that the methods are different by how they approach the initial guess.

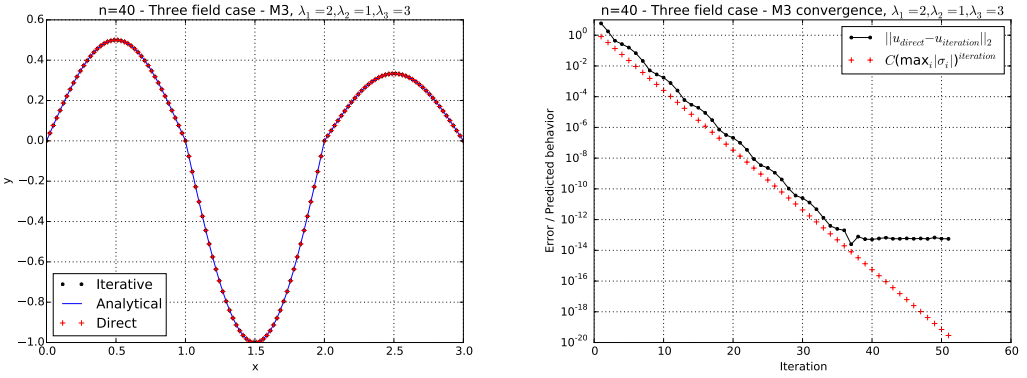

FIGURE 5.14: Three-field case, Method 3 – $\lambda_1 = 2, \lambda_2 = 1, \lambda_3 = 3, |\sigma_{max}| \approx 0.4082$

Fig. 5.14 illustrates the case where $\lambda_1 = 2, \lambda_2 = 1$ and $\lambda_3 = 3$. We see the predicted convergence with a saw-tooth behavior similar to that seen in Method 2. The solution matches the direct solver as expected.

Next we consider another case where we predict convergence; Fig. 5.14 ($\lambda_1 = 50, \lambda_2 = 1, \lambda_3 = 50$):

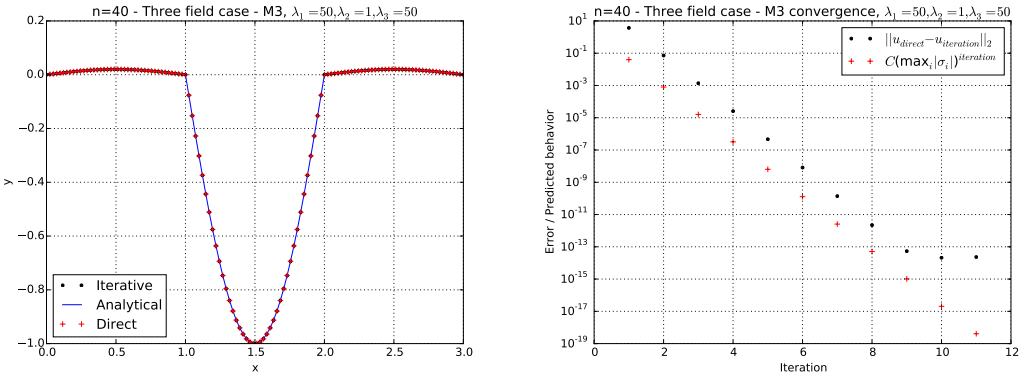

FIGURE 5.15: Three-field case, Method 3 – $\lambda_1 = 50, \lambda_2 = 1, \lambda_3 = 50, |\sigma_{max}| = 0.02$

Indeed we see convergence up to machine limits after 10 iterations. Again the spectral radius accurately predicts the rate of convergence. Finally we look at a case where we predict divergence; Fig. 5.16 ($\lambda_1 = 1, \lambda_2 = 3, \lambda_3 = 2$):

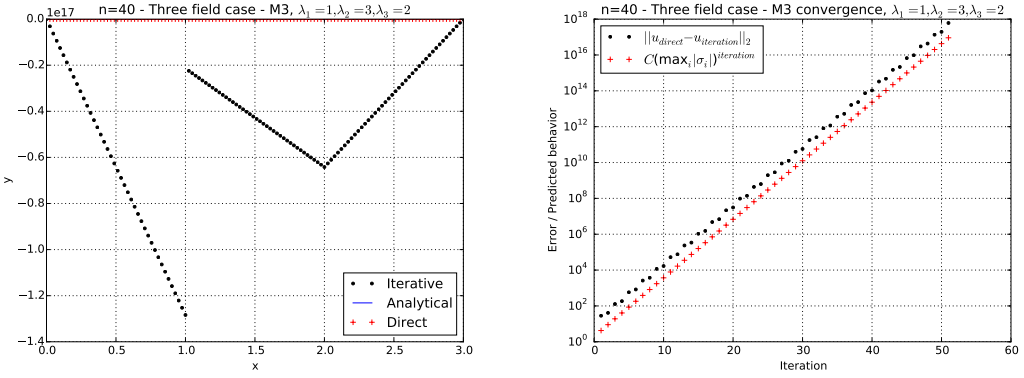

FIGURE 5.16: Three-field case, Method 3 – $\lambda_1 = 1, \lambda_2 = 3, \lambda_3 = 2, |\sigma_{max}| \approx 2.1213$

As predicted we see divergence and the solution has "exploded". The error grows at the rate predicted by the spectral radius.

5.2.5 Final overview

Building on the analysis in 4.4, we can gain a better understanding of the convergence properties of the three methods by visualizing their respective spectral radii using a heat map. As λ_1 and λ_3 are interchangeable in the three field case as they only determine from which direction the problem is solved, we hold λ_2 constant.

Note that the y-axis on the two plots represent different values as taking smaller values for λ_3 in Method 1 quickly results in spectral radii far larger than 1 .

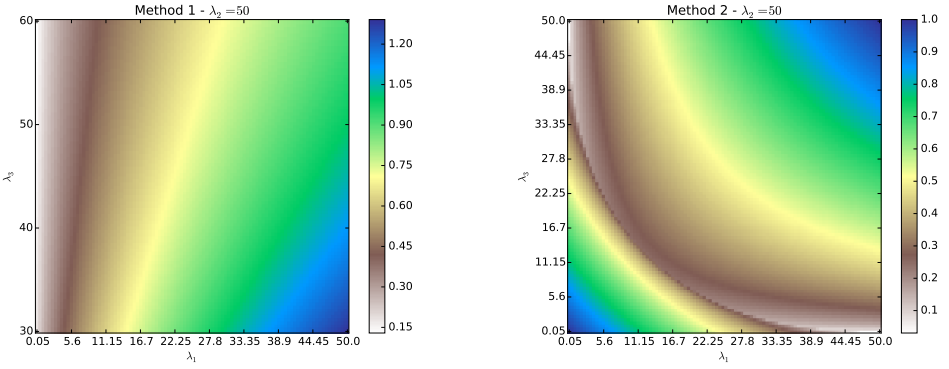

FIGURE 5.17: Visualization of spectral radius of selected values for Method 1 and Method 2, holding $\lambda_2 = 50$ constant.

But if we consider that λ_1 and λ_3 are interchangeable in applications of Method 1, then a fair comparison between Methods 1 and 2 could be done by assuming that we always choose the order such that λ_1 and λ_3 are placed optimally. Doing this allows us to compare the Methods with the same axes without Method 1 "exploding". Method 3 can't be represented in the same area without its spectral radius becoming extremely large and thus it is shown for a different set of λ-values.

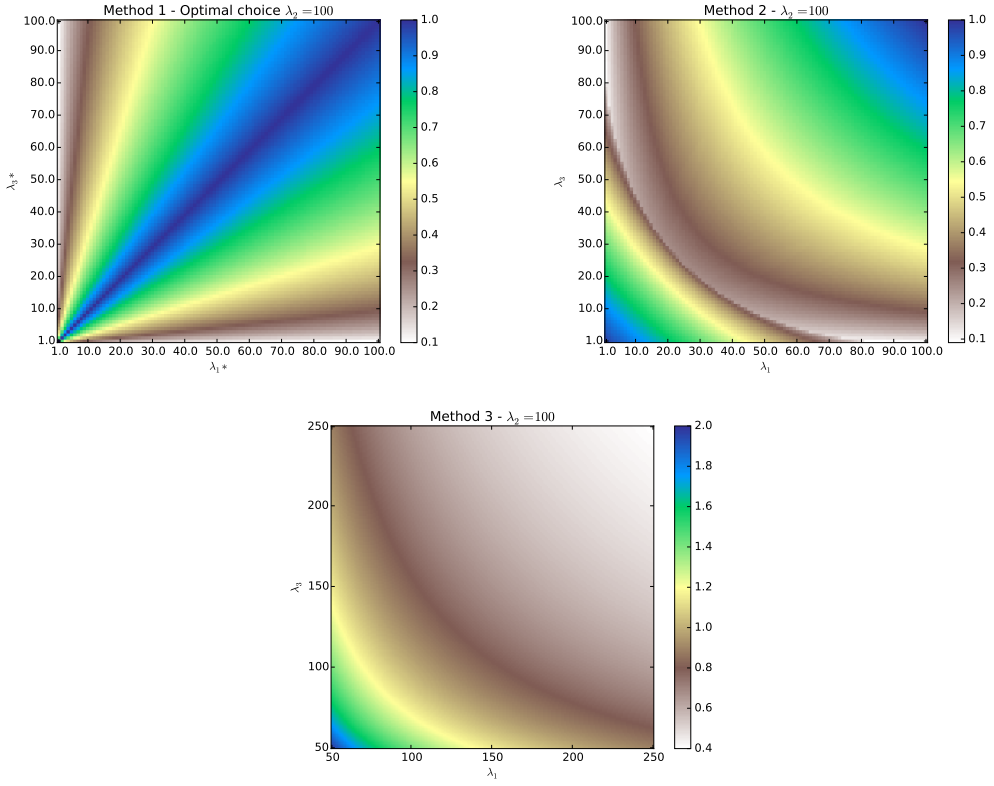

FIGURE 5.18: Visualization of spectral radius of selected values for Method 1, Method 2 and Method 3, holding $\lambda_2 = 100$ constant.

Now we see the same diagonal symmetry axis emerge for Method 1 as is also seen for Methods 2 and 3.

We see that the three methods have distinctly different and to a large extent complementary areas of fast convergence. The final step in assessing the strength of our combined toolbox is of course to choose the optimal method in addition to checking the optimal directional order for Method 1.

One way to do this is to begin with Method 1 and then gradually add the other methods in areas where they perform better. We begin by combining the optimal implementation of Method 1 with Method 2.

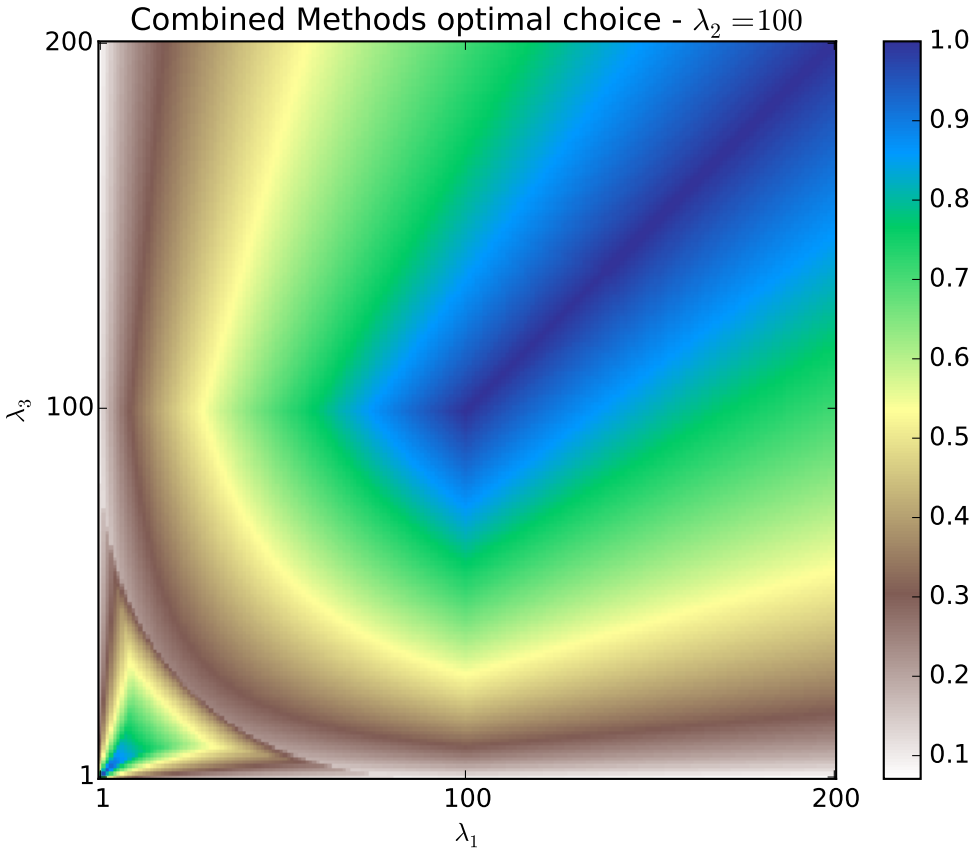

FIGURE 5.19: Combined Method 1 and Method 2 – Visualization of the smallest spectral radius for selected values, holding $\lambda_2 = 100$ constant.

We see that Method 2 offers benefits in convergence when both $\lambda_1 \leq \lambda_2$ and $\lambda_3 \leq \lambda_2$ in that it does offer more efficient solutions within parts of this area when compared to Method 1. Even inside the square where this holds true Method 1, is better along the axes close to the origin. We now add Method 3 to our visualization.

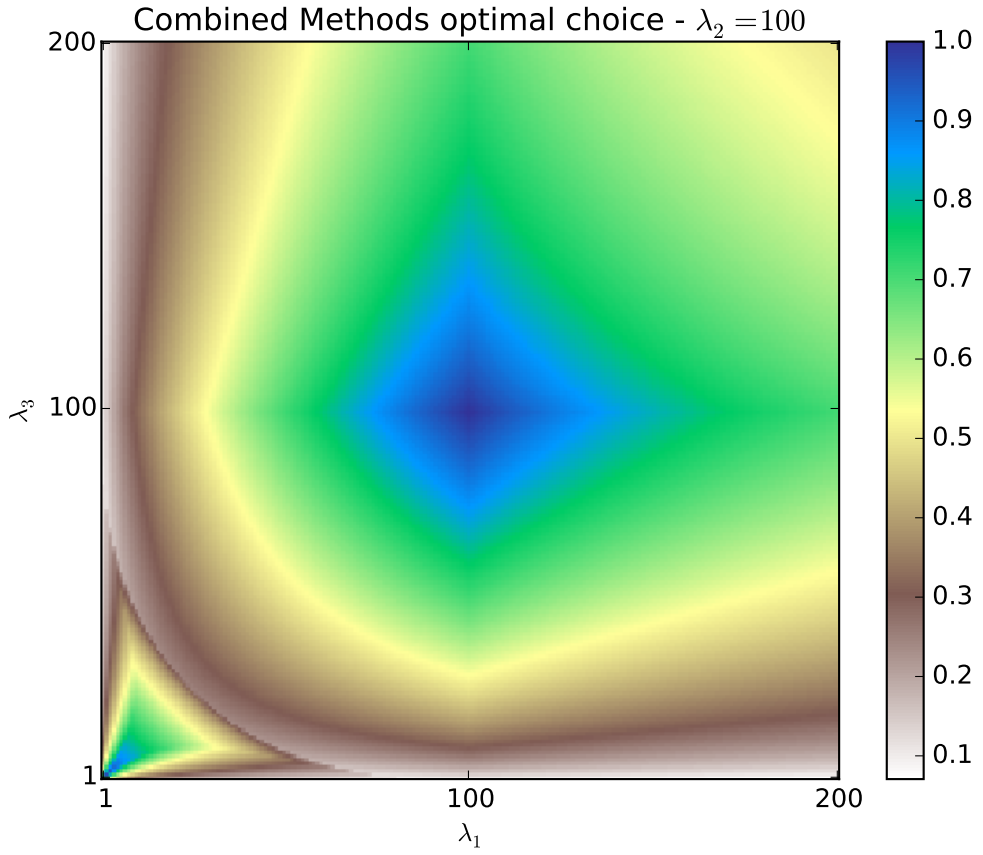

FIGURE 5.20: Combined Method 1, Method 2 and Method 3 – Visualization of the smallest spectral radius for selected values, holding $\lambda_2 = 100$ constant.

We see that Method 3 significantly improves the performance where both $\lambda_2 \leq \lambda_1$ and $\lambda_2 \leq \lambda_1$. Now that we have all three methods combined we can consider much wider intervals for λ_1 and λ_3 without ever exceeding a spectral radius of one.

This gives us our grand overview:

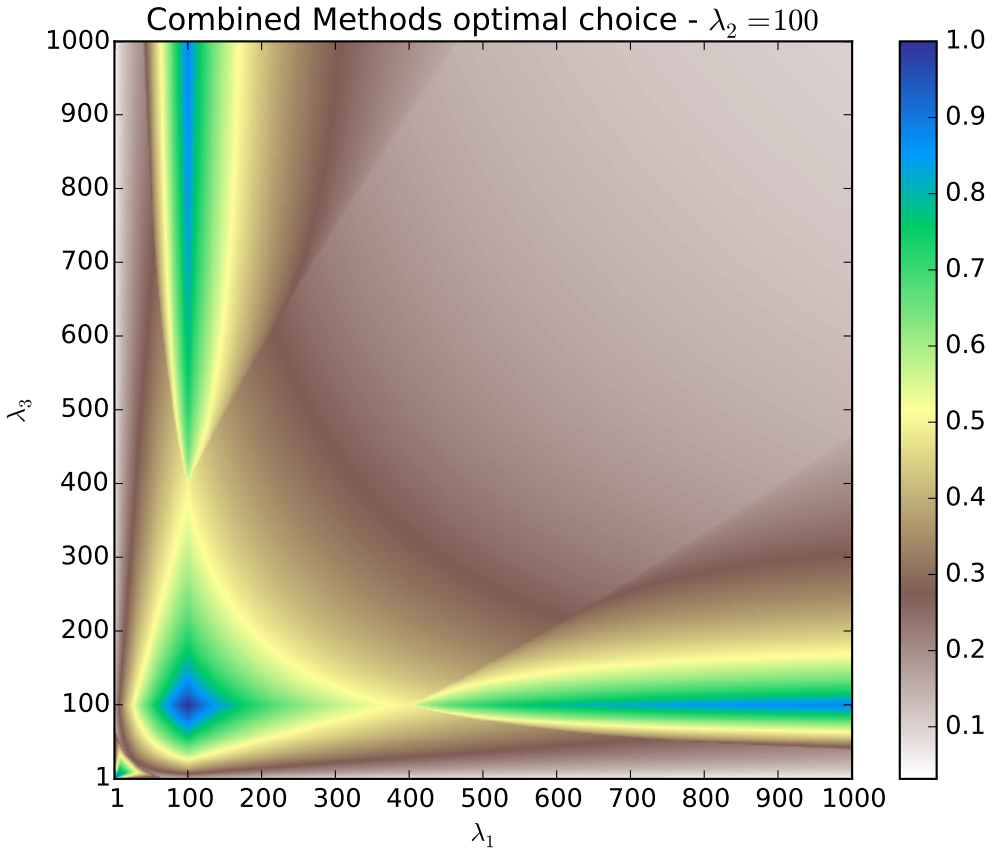

FIGURE 5.21: Visualization of the smallest spectral radius for selected values, holding $\lambda_2 = 100$ constant.

What we see is a collection of methods that thrive on differences. In this example $\lambda_2 = 100$ and we see that along the axes where either $\lambda_1 = 100$ or $\lambda_3 = 100$, we have larger spectral radii, at some points approaching one. Yet by combining our three methods we readily reach the areas where the coefficients are different, in many cases with a very quick convergence.

5.2.6 Real world examples

Until now we have treated the λ's as arbitrarily chosen numbers for the purpose of convergence analysis. The λ in the equation (2.2) represents the thermal conductivity of a material. We now introduce some examples on how the algorithms would perform for some possible real world applications [14][15][1]. All units are $W/(mK)$. The spectral radius of the method best suited for a particular combination is written in bold.

λ_1	λ_2	λ_3	M1$\lvert\sigma_{max}\rvert$	M2$\lvert\sigma_{max}\rvert$	M3$\lvert\sigma_{max}\rvert$
Steel(54)	Coolant (0.063)	Copper(401)	857	6364	**0.0012**
Water(0.58)	Glass(1.05)	Water(0.58)	1	**0.5524**	1.81
Steel(54)	Iron(80)	Copper(401)	**0.3669**	3.79	1.12
Oak (0.17)	Air(0.024)	Steel(54)	7.01	2249	**0.1411**
Air(0.024)	Copper(401)	Coolant (0.063)	**0.6172**	0.9998	106326793
Air(0.024)	Water(0.58)	Steel(54)	**0.0211**	92	24
Brass(109)	Copper(401)	Brass(109)	1	**0.2718**	3.68
Steel(54)	Engine oil(0.15)	Steel(54)	360	360	**0.0028**
Air(0.024)	Copper(401)	Air(0.024)	1	**0.9999**	279134985

These examples illustrate how the three methods complement each other. For each example we get at the very least a mathematically convergent method and in several cases we get very fast convergence. The last example is useful to illustrate that while the best method gives a nominally convergent algorithm with M2$\lvert\sigma_{max}\rvert = 0.9999$, it would be unacceptably slow for practical applications.

Summarizing; this collection of methods provides a potentially very strong tool for iterative solutions of the three field case but it is highly sensitive to the problem parameters.

[1] Coolant values are specifically Fluorinert FC-77

Chapter 6

Conclusions and comments

We have confirmed earlier results on the convergence behavior of the two-field case. Further we have created three algorithms for solving the 1D three-field case and analyzed their convergence by rewriting them as fixed-point iterations acting on the boundary points. The spectral radii of these fixed-point iteration-matrices depend only on material properties and they have been shown to give reliable estimates for the convergence of the whole iterative solution.

We have also shown that different choices of sequencing the Dirichlet-Neumann iteration can result in quantitatively different and in many cases complementary convergence properties. Summing up:

- The spectral radii of the fixed-point iterations acting on the boundaries are accurate predictors of the total solution error in all cases considered.
- The rate of convergence in the 1D three-field case depends only on the material properties, not on the discretization.
- By choosing where to impose either Neumann or Dirichlet boundary conditions and by what sequence they are carried out, we can construct algorithms with different convergence properties.
- Choosing the method based on problem parameters is crucial to be able to achieve the fastest convergence rates.
- The choice of best method does not guarantee fast convergence, i.e., there are combinations where neither of the three methods delivers fast convergence.

6.1 Recommendations for further study

We have shown for the 1D three field case, that by choosing different order and different combinations of Dirichlet-Neumann conditions, we can create algorithms with quantitatively different convergence properties. The dream would of course be having a suitable algorithm for any combination of values. We improve our toolbox incrementally every time we develop a new method which better reaches some areas when compared to the methods we had before. A natural continuation of the work would thus be examining what other combinations, if any, are possible.

As noted in the beginning of this work, there is reason to believe that the 1D convergence rates could match the asymptotic convergence behavior of the 2D case. It would be highly desirable if we could use the results of this work to predict convergence behavior in higher dimensions, studying if this is indeed the case is another natural continuation.

Our algorithms were constructed specifically for the three field case. Their method of construction could be extended to any n-field case. It could be of interest to see what behavior we see as we add fields, perhaps one could even find systematic changes and thus gain a general formula for generating the best possible method for any n-field problem.

A final project with an end-user in mind, provided this approach proves its extended usefulness as speculated above, would be to collect all known algorithms into a mother program. It would take material constants as its input and generate the most suitable program for the specific problem given, together with an estimator for the rate of convergence.

[1] SpaceX. *Falcon 9 Launch Vehicle - Payload user's guide - Rev. 2*, October 21, 2016 (p. 10)

[2] A. Neys. *In-Vehicle Brake System Temperature Model*, Master's Thesis, Chalmers University 2012

[3] A. Monge. *The Dirichlet-Neumann Iteration for Unsteady Thermal Fluid Structure Interaction*, Licentiate Theses in Mathematical Sciences 2016:3, Lund University

[4] A. Monge, P. Birken. *Convergence Analysis of the Dirichlet-Neumann Iteration for Finite-element Discretizations* PAMM Proc. Appl. Math. Mech. 16, 2016 (pp. 733 - 734)

[5] A. Toselli, O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, Springer, Berlin 2004 (pp. 1-10)

[6] J. Côté, M. J. Gander, L. Laayouni, S. Loisel. *Comparison of the Dirichlet-Neumann and Optimal Schwarz Method on the Sphere*, Domain Decomposition Methods in Science and Engineering, Springer, Berlin, 2005 (pp. 235-242)

[7] H. Berninger, R. Kornhuber, O. Sander. *Convergence Behaviour of Dirichlet–Neumann and Robin Methods for a Nonlinear Transmission Problem*, Domain Decomposition Methods in Science and Engineering XIX, Springer, Berlin 2011 (pp. 87-98)

[8] I. Maiera, B. Haasdonk. *A Dirichlet-Neumann reduced basis method for homogeneous domain decomposition problems*, Appl. Num. Math. 78, 2014 (pp. 31-48)

[9] M. H. Gutknecht. *Iterative Methods - Part II of "Software for Numerical Linear Algebra", Lecture notes*, ETH Zürich, Spring semester 2008 (pp. 6-9)

[10] E. W. Weisstein. *Boundary Conditions.* From MathWorld – A Wolfram Web Resource, http://mathworld.wolfram.com/BoundaryConditions.html (retrieved 02-10-2017)

[11] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra.* SIAM, Philadelphia, PA, 2000 (p. 514)

[12] L. Papula. *Mathematische Formelsammlung, 10. Auflage*, Vieweg+Teubner, Wiesbaden 2009 (p. 96)

[13] A. M. Bruckner, J. B. Bruckner, B. S. Thomson. *Real Analysis: Second Edition*, 2008 (p. 620)

[14] Engineering toolbox. *Resource of thermal conductivities.* http://www.engineeringtoolbox.com/thermal-conductivity-d_429.html (retrieved 08-10-2017)

[15] 3M. *Fluorinert Electronic Liquid FC-77 Product Information* http://multimedia.3m.com/mws/media/648930/fluorinert-electronic-liquid-fc-77.pdf (retrieved 08-10-2017)

Images

[F1] SpaceX. https://upload.wikimedia.org/wikipedia/commons/4/44/SpaceX_Testing_Merlin_1D_Engine_In_Texas.jpg (retrieved 13-12-2017) licensed under Creative Commons CC0 1.0 Universal Public Domain Dedication.

[F2] Clément Bucco-Lechat. https://commons.wikimedia.org/wiki/File:Geneva_MotorShow_2013_-_Koenigsegg_brake_disc.jpg (retrieved 12-12-2017) licensed under Creative Commons Attribution-Share Alike 3.0 Unported.

[F3] S. J. Ling, J. Sanny, W. Moebs. University Physics Volume 2, Openstax. https://openstax.org/details/books/university-physics-volume-2 (retrieved 05-10-2017) licensed under Creative Commons Attribution License v4.0