# DOMAIN DECOMPOSITION APPROACHES FOR THE GENERATION OF EQUIDISTRIBUTING GRIDS

ALEXANDER HOWSE

# Domain Decomposition Approaches for the Generation of Equidistributing Grids

by

© Alexander Howse

*A thesis submitted to the*

*School of Graduate Studies*

*in partial fulfillment of the*

*requirements for the degree of*

*Master of Science*

Department of Mathematics and Statistics

Memorial University of Newfoundland

April 2013

St John's Newfoundland & Labrador

# Table of Contents

# Abstract

To solve boundary value problems whose solutions contain moving fronts or sharp variations, moving mesh methods can be used to achieve reasonable solution resolution with a fixed, moderate number of mesh points. Such meshes are obtained by solving nonlinear elliptic differential equations which are governed by an equidistribution principle. In this thesis we combine the moving mesh technique with several Schwarz domain decomposition methods, which allow elliptic boundary value problems to be solved by parallel computation. Convergence results are established for both parallel and alternating iterations using classical, optimal, or optimized Schwarz transmission conditions. Results for multidomain and time-dependent variations are also presented. Four potential sets of optimized transmission conditions are proposed for a 2D mesh generation algorithm. Numerical results are provided to illustrate typical behavior of the proposed algorithms.

# List of Tables

# List of Figures

# Acknowledgments

I would like to thank the Natural Resources and Engineering Council of Canada (NSERC) for financially supporting my studies.

I would like to thank the faculty and staff of the Department of Mathematics and Statistics for their instruction and assistance throughout my time at Memorial University of Newfoundland.

I would like to thank the examiners of my master's thesis, Dr. Felix Kwok and Dr. Chun-Hua Ou, for their valuable comments and suggestions. I accept responsibility for all remaining errors.

I would like to thank my supervisor, Dr. Ronald Haynes, for his guidance, encouragement and patience during our time working together.

Finally, I would like to thank my parents, Joan and Maxwell Howse, and my fiancée, Samantha Bastone, for their love and support.

# Chapter 1

# Introduction

The efficient solution of partial differential equations (PDEs) which vary over disparate space and time scales often benefit from the use of non-uniform meshes chosen to adapt to the local solution behavior. There are many different adaptive methods which generate meshes for such PDEs, most falling into one, or more, of three general categories [22, 32]:

- $h$-refinement: change the number of points to locally coarsen or refine the mesh according to an a posteriori error estimate,
- $p$-refinement: varying the order of the numerical method used over the domain — achieved by varying the order of basis polynomials in finite element methods, or

- $r$-refinement: relocating mesh points to best resolve the solution while keeping the number of mesh points and mesh topology fixed.

It is the last of these categories, also known as *moving mesh methods*, which we consider here.

A standard way to perform spatial mesh adaptation is the equidistribution principle of de Boor [12]. Given some positive measure $M(t, x, u)$ of the error or difficulty in representing the solution $u(t, x)$ over the physical domain $\Omega$ at time $t$, typically referred to as a mesh density function, equidistribution requires the integral of $M$ to be equally distributed over each mesh element. $M$ is typically chosen so that it is large where we expect proportionally large error in the computed solution. Enforcing equidistribution concentrates mesh points in regions where the error is large. The mesh is typically determined by solving a nonlinear mesh PDE which is coupled to the physical PDE of interest. Particular mesh PDEs have been developed and analyzed for specific problems: see for example [14, 54] for CFD problems and [56, 57] for flow and magnetohydrodynamics. Thorough recent reviews of grid generation by moving mesh methods can be found in [7, 39].

In addition to mesh equidistribution, we wish to solve both the physical and mesh PDEs by taking advantage of parallel computation. We introduce spatial parallelism by using a domain decomposition approach, in which the domain of a boundary value

problem (BVP) of interest is partitioned into multiple subdomains and the original problem is reformulated as a coupled set of smaller subdomain problems to be solved in a parallel iteration, in the hope of obtaining the solution to the original problem more quickly. The coupling is created through the use of transmission conditions, designed to ensure that solutions on adjacent subdomains agree, and that the original solution is represented as a piecewise combination of subdomain solutions over their respective subdomains.

In this thesis we consider the combination of mesh equidistribution and domain decomposition techniques, resulting in a parallel mesh adaptation method. In Chapter 4 we consider the steady mesh problem, which determines a mesh for a time independent function $u(x)$, presenting convergence results for both parallel and alternating classical Schwarz iterations for two or more subdomains in Section 4.2, and convergence results for optimal Schwarz and optimized Schwarz in Sections 4.3 and 4.4, respectively. In Chapter 5 we discuss the application of DD to the time dependent mesh problem, providing convergence results for parallel and alternating classical Schwarz iterations for two or more subdomains. This combination of mesh equidistribution and domain decomposition has previously been presented in the experimental papers [29, 31, 32] and some results of Chapters 4 and 5 have previously been published by Gander and Haynes in [22], Haynes and Howse in [30], and by all

three authors in [23].

An outline of the remainder of the thesis is as follows. In Chapter 2 we provide an introduction to moving mesh methods as determined by the equidistribution principle. We describe how mesh equations are derived in both steady and time dependent cases for a single spatial dimension, along with details on how these equations can be implemented numerically. We then turn to higher dimensional cases, discussing several general methods used to develop mesh adaptation equations, then present a specific algorithm for 2D mesh generation, along with details for numerical implementation. In Chapter 3 we introduce domain decomposition methods for general elliptic boundary value problems, discussing what are known as classical Schwarz methods, and describing how they can be modified to obtain optimized and optimal Schwarz methods. We also show two different ways these methods can be extended to solving time dependent parabolic problems. In Chapter 6 we discuss how Schwarz methods can be applied to a particular 2D mesh adaptation problem, proposing several possible optimized Schwarz variants. In Chapter 7 we provide details on the numerical implementation of the domain decomposition mesh adaptation iterations proposed in Chapters 4 through 6. We show numerous numerical results to illustrate typical trends and behavior observed for these iterations. Finally, in Chapter 8 we give a summary of the results presented and highlight areas for future work.

# Chapter 2

# Moving Mesh Methods via the Equidistribution Principle

Of the three broad categories of adaptive mesh methods described in Chapter 1, this chapter will focus on some particular $r$-refinement, or moving mesh, methods for one or more spatial dimensions. We begin by looking at the 1D case, describing how many of the standard equations governing mesh movement are based on the equidistribution principle (EP) of de Boor [12] and discussing the set of equations known as moving mesh partial differential equations (MMPDEs). We then move to the setting of higher dimensions, presenting moving mesh methods developed for two and three spatial dimensions.

## 2.1 1D Moving Mesh Methods

The main goal of moving mesh methods is to use a fixed number of mesh points to best represent, or resolve, a function $u(x)$. This may be a known function or may be the solution to a problem of interest. A commonly used method for function resolution follows from the concept of *equidistribution*, originating in the work of de Boor in the 1970s. In the papers [11] and [12] de Boor addresses a similar problem of how to best approximate functions using polynomial splines of fixed order with varying knots — that is, placing the knots in a perhaps nonlinear fashion to best capture the characteristics of a given function. We use the text [39] as a general reference throughout the following discussion.

For a function $M(x) > 0$, continuous over the bounded interval $[a, b]$, and given an integer $N > 1$, equidistribution involves determining a mesh $x_1 = a < x_2 < \cdots < x_N = b$ such that $M(x)$ is evenly distributed over each sub-interval determined by the mesh points. By "evenly distributed", we mean that the area under $M(x)$ is equal for each sub-interval, that is

$$\int_{x_1}^{x_2} M(x)dx = \cdots = \int_{x_{N-1}}^{x_N} M(x)dx. \tag{2.1}$$

Such a mesh $\{x_1, x_2, \ldots, x_N\}$ is referred to as an equidistributing mesh for $M(x)$.

For the purposes of moving mesh methods, when determining a mesh for $u(x)$ the

function $M(x)$ is in fact typically $M(u(x))$ and has been referred to in past literature as a *monitor function* [7, 32, 35] and more recently as a *mesh density function* [22]. We note that in [39] they instead adopt the term monitor function for $(M(x))^2$ to be consistent with the multi-dimensional case, which may be a source of confusion for some readers. Mesh density functions are chosen to somehow indicate the error or difficulty in resolving the function $u(x)$, such that $M(x)$ is large at values of $x$ for which $u(x)$ is difficult to resolve. The development of appropriate mesh density functions has been the area of much study, and choosing an appropriate function can contribute greatly to the success of the numerical implementation of the moving mesh method. As a detailed summary of possible mesh density functions can be found in [39], we will instead comment on some of the most common choices for $M(x)$.

The trivial mesh density function is $M(x) \equiv 1$, which simply results in a uniformly distributed mesh, that is, no adaptivity. Two commonly used mesh density functions are the arc-length mesh density function, $M(x) = \sqrt{1 + |u_x|^2}$, and the curvature arc-length mesh density function, $M(x) = \sqrt[4]{1 + |u_{xx}|^2}$. Both of these mesh density functions quantitatively measure certain aspects of the function $u(x)$, and agree with an intuitive idea of "difficulty" — we would anticipate that large or rapid variations in function value would require more mesh points to resolve accurately, and this behavior corresponds to large values of the arc-length or curvature mesh density functions. It

has been observed in practice that the arc-length mesh density function can lead to numerical difficulty, concentrating too many points at singularities and making the coupled mesh and physical PDE problem exceedingly stiff [52]. An alternative which approximates the arc-length mesh density function and which may mitigate these difficulties is $M(x) = \sqrt{1 + \frac{1}{\alpha}|u_x|^2}$, which is known as the relaxed arc-length function. For $\alpha \gg 1$, $M(x) \approx 1$, and an approximately uniform grid results. As $\alpha \to 1$, $M(x)$ approaches the original arc-length function.

While mesh density functions involving arc-length or curvature are often used due to their ease of implementation and straightforward interpretation, it is also possible to formulate mesh density functions based on interpolation error. Indeed, in [39], the authors demonstrate how to derive an optimal mesh density function based on interpolation error, obtaining the continuous function

$$M(x) = \left[1 + \frac{1}{\alpha}|u^{(k+1)}(x)|^2\right]^{\frac{1}{1+2(k-m+1)}},$$

where

$$\alpha = \left[\frac{1}{b-a}\int_a^b |u^{(k+1)}|^{\frac{2}{1+2(k-m+1)}} dx\right]^{1+2(k-m+1)},$$

is described as an adaptation intensity parameter, as it controls the impact the derivatives of $u(x)$ on the mesh adaptation. The constant $k$ is the degree of the interpolating piecewise polynomials ($k = 0$ being piecewise constant, $k = 1$ piecewise linear, and

so forth) and $m$ controls whether the error is measured in the $L^2$ norm ($m = 0$) or the $H^1$ semi-norm ($m = 1$). In deriving error bounds for this mesh density function and showing it to be optimal, the authors illustrate the key advantage of adaptive meshes over uniform meshes: the uniform mesh error bound has a solution dependent factor $||u^{(k+1)}||_{L^2(a,b)}$ whereas the optimal mesh error bound has a factor of $||u^{(k+1)}||_{L^{\frac{2}{1+2(k-m+1)}}(a,b)}$. While these quantities are of similar size for a smooth function $u$, in the case of $u$ not being smooth the factor for the adaptive error bound is much smaller, hence in such a case the error bound for an adaptive mesh will be significantly smaller than that for a uniform mesh with the same number of mesh points.

Having presented several different possible choices for the mesh density function $M(x)$, we note that each of these functions is bounded away from zero. This is not unintentional — if there is some $\breve{m}$ such that $M(x) \geq \breve{m} > 0 \; \forall x \in [a, b]$, then we can infer the existence of a unique equidistributing mesh for any integer $N > 1$. We state this result of [39] as Theorem 2.1.

**Theorem 2.1.** *For any strictly positive mesh density function $M(x)$ and any given integer $N > 1$ there exists a unique equidistributing mesh of $N$ points satisfying* (2.1).

*Proof.* As we require the integral of $M(x)$ over each subinterval to be equal, we can

rewrite (2.1) as

$$\int_a^{x_j} M(x)dx = \frac{j-1}{N-1}\sigma, \quad \text{for} \quad j = 1, \ldots, N, \tag{2.2}$$

where we denote

$$\sigma = \int_a^b M(x)dx.$$

By our assumption that there exists $\breve{m}$ such that $M(x) \geq \breve{m} > 0 \; \forall x \in [a, b]$, we know that $\int_a^x M(\bar{x})d\bar{x}$ is a strictly monotone increasing function of $x$, hence each $x_j$ of the equidistributing mesh is uniquely determined. □

In Figure 2.1 we plot the function $u(x) = \frac{1}{2}[1 - \tanh(20x - 10)]$ using 40 mesh points, indicating the position of mesh points, as determined using the arc-length mesh density function, below the plot. The mesh points are spread out in regions of little variation and cluster together where the function value changes rapidly.

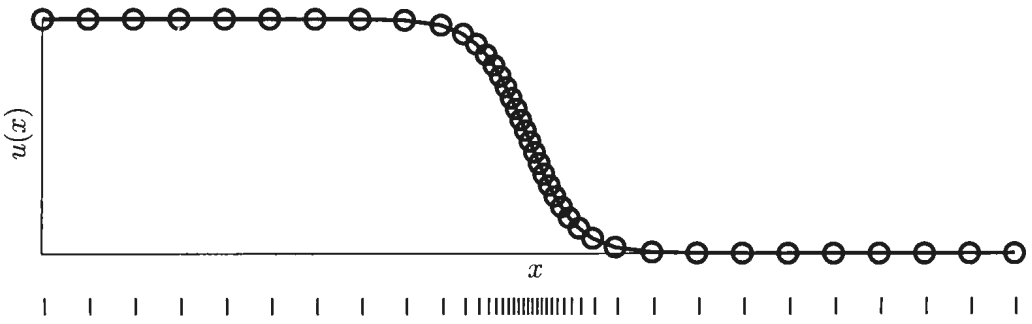


Figure 2.1: A specified function $u(x)$ on an equidistributing mesh determined using the arc-length mesh density function.

Although we have introduced the concept of equidistribution for a discrete mesh, it is convenient for theoretical work to consider a continuous form of the equidistribution condition. To do so, we introduce a computational coordinate $\xi$ over the interval $[0, 1]$ and suppose that the mesh $\{x_1, x_2 \ldots, x_N\}$ is obtained via a coordinate transformation

$$x = x(\xi), \quad \xi \in [0, 1], \quad x(0) = a, \quad x(1) = b,$$

which satisfies $x_i = x(\xi_i)$, $i = 1, \ldots, N$, where

$$\xi_i = \frac{(i-1)}{(N-1)}, \quad i = 1, \ldots, N.$$

Under this transformation (2.2) becomes

$$\int_a^{x(\xi_i)} M(x)dx = \xi_i \sigma, \quad i = 1, \ldots, N.$$

More generally, if a mapping $x = x(\xi)$ satisfies the equation

$$\int_a^{x(\xi)} M(x)dx = \xi\sigma, \quad \forall \xi \in (0, 1), \tag{2.3}$$

then it is called an *equidistributing coordinate transformation* for the function $M(x)$ [39].

While the integral form of (2.3) will be used extensively in proving results, it is the differential version of the EP which will typically be used in subsequent discussion. To obtain this differential equation, we differentiate (2.3) once and twice with respect

to $\xi$ to obtain

$$M(x(\xi))\frac{dx(\xi)}{d\xi} = \sigma,$$

and

$$\frac{d}{d\xi}\left(M(x(\xi))\frac{dx(\xi)}{d\xi}\right) = 0, \tag{2.4}$$

respectively. When combined with the boundary conditions $x(0) = a$ and $x(1) = b$, (2.4) is the boundary value problem which governs mesh equidistribution.

## 2.2 Moving Mesh Partial Differential Equations

Up to this point we have focused on how to find an equidistributing mesh $x(\xi)$ for some $u(x)$ which varies only in space. In reality, it is likely the function of interest will be of the form $u(x,t)$, and in such cases we wish to find a mesh that also varies in time. Our problem is thus to determine a mesh $\{x_1(t), x_2(t), \ldots x_N(t)\}$ such that, for $t \geq 0$

$$\int_a^{x_j(t)} M(x(t),t)\,dx = \frac{j-1}{N-1}\int_a^b M(x(t),t)dx, \quad \text{for} \quad j = 1,\ldots,N.$$

This is the same condition as for the steady case, except we now allow $M(x(t),t)$ to vary in time, resulting in a mesh which also changes as time passes. The reason for calling these adaptive techniques "moving mesh" methods is apparent in the time dependent case: the fixed number of discrete mesh points will move throughout the

interval $[a, b]$ to best resolve the function $u(x, t)$ as it evolves. For example, in Figure 2.2 we plot the mesh trajectory for the function

$$u(x,t) = \frac{1}{2}[1 - \tanh(c(t)[x - t - 0.4]), \quad c(t) = 1 + \frac{199}{2}[1 + \tanh(50t - 2.5)],$$

a similar problem appearing in [22], which illustrates how the mesh moves as time passes. As in the steady case, mesh points cluster where there is rapid change in function value, and as the location of this region of rapid change varies as time passes, the mesh moves in response.

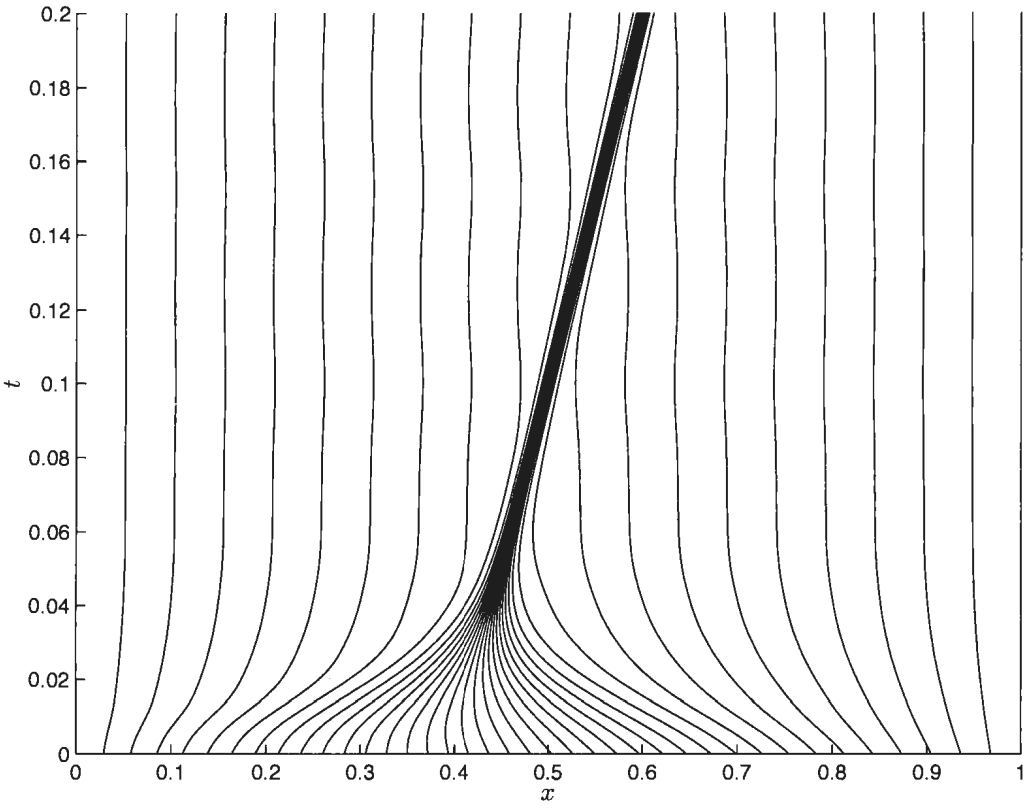


Figure 2.2: The mesh trajectory associated with a function $u(x, t)$.

To obtain the continuous formulation of this moving mesh we proceed as before, introducing a computational coordinate $\xi$ over $[0,1]$ such that

$$x = x(\xi, t), \quad \xi \in [0,1], \quad x(0,t) = a, \quad x(1,t) = b,$$

to arrive at the condition

$$\int_a^{x(\xi,t)} M(x,t)dx = \xi\sigma(t), \quad \forall \xi \in (0,1), \tag{2.5}$$

where

$$\sigma(t) = \int_a^b M(x(t),t)dx.$$

By differentiating (2.5) twice with respect to $\xi$, it is clear that our time dependent mesh transformation $x(\xi,t)$ satisfies

$$\frac{\partial}{\partial \xi}\left(M(x(\xi,t),t)\frac{\partial x(\xi,t)}{\partial \xi}\right) = 0, \quad \forall t \geq 0. \tag{2.6}$$

As this equation does not explicitly involve the rate of change of the mesh function, $\dot{x}(\xi,t) := \frac{\partial x(\xi,t)}{\partial t}$, it is known as a quasi-static EP (QSEP).

While the desired mesh transformation $x(\xi,t)$ will satisfy (2.6) exactly, there are many situations when an alternate method of determining the mesh would be beneficial. In practice the function $u(x,t)$ will often not be known a priori — it being the solution of some underlying physical PDE of interest. In such a case (2.6) will be coupled to the PDE through the monitor function resulting in both problems being

solved together. If we semi-discretize (2.6) in space, we obtain a system of algebraic equations to be solved. Applying the method of lines to solve the physical PDE will result in a system of differential algebraic equations (DAEs). Such systems are typically difficult to solve, being stiff and ill-conditioned [1, 39].

To overcome this problem we instead may use a PDE related to (2.6) which explicitly involves the mesh speed, $\dot{x}(\xi, t)$. The remainder of this section follows the work of [39] and [35], the former discussing adaptive mesh generation in great detail and the latter being one of the original references for the methods which follow. If we semi-discretize such an equation in space we will obtain a system of ordinary differential equations (ODEs) rather than algebraic equations, thus when coupled with the numerical solution of the physical PDE we will have a system of ODEs, which are typically easier to solve than a system of DAEs. In addition to this, introducing the mesh speed term provides a measure of temporal smoothing, which is beneficial for the accurate solution of certain physical PDEs.

An equation for determining a mesh transformation which involves the mesh speed is known as a *moving mesh PDE* (MMPDE). There are many different MMPDEs which may be used, and often multiple ways to derive a given MMPDE, see [7, 33, 34, 38, 52]. One possibility is to obtain an MMPDE by time differentiation of QSEPs.

For instance, differentiating (2.5) once with respect to time gives

$$M(x(\xi,t),t)\dot{x}(\xi,t) + \int_a^{x(\xi,t)} \frac{\partial}{\partial t} M(\tilde{x},t)d\tilde{x} = \xi\dot{\sigma}(t),$$

and then by differentiating twice with respect to $\xi$, we obtain

$$\frac{\partial^2}{\partial\xi^2}(M\dot{x}) = -\frac{\partial}{\partial\xi}\left(\frac{\partial M}{\partial t}\frac{\partial x}{\partial\xi}\right). \tag{MMPDE1}$$

The process by which (MMPDE1) was derived implicitly assumed that the mesh $x(\xi,t)$ will satisfy the QSEP (2.5) at any given time, which is typically not the case in actual numerical computations. As such, it is desirable to derive MMPDEs which are relaxed, such that the difference between the computed and equidistributing meshes will play a role. To do so, we require the mesh to satisfy (2.6) at some time $t+\tau$ $(0 \leq \tau \ll 1)$, that is

$$\frac{\partial}{\partial\xi}\left(M(x(\xi,t+\tau),t+\tau)\frac{\partial x(\xi,t+\tau)}{\partial\xi}\right) = 0. \tag{2.7}$$

By expanding the functions of (2.7) using Taylor series and dropping terms of $O(\tau^2)$ and higher, we obtain

$$\frac{\partial^2}{\partial\xi^2}(M\dot{x}) = -\frac{\partial}{\partial\xi}\left(\frac{\partial M}{\partial t}\frac{\partial x}{\partial\xi}\right) - \frac{1}{\tau}\frac{\partial}{\partial\xi}\left(M\frac{\partial x}{\partial\xi}\right). \tag{MMPDE2}$$

Compared to (MMPDE1), we see that the effect of relaxing the equidistribution condition has introduced an additional term in (MMPDE2), which serves as a correction:

if $x(\xi, t)$ is not equidistributed, the additional term

$$-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right),$$

will move the mesh towards equidistribution, even in cases where $M(x,t)$ is time-independent, avoiding a potential pitfall of (MMPDE1). Arguing that the term involving $\frac{\partial M}{\partial t}$ may be dropped without any significant loss of effectiveness leads to

$$\frac{\partial^2}{\partial \xi^2}(M\dot{x}) = -\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right). \qquad \text{(MMPDE3)}$$

Expanding the derivative on the left hand side of (MMPDE3) and making the further simplifying assumption that the term involving $\dot{x}\frac{\partial M}{\partial \xi}$ is negligible we obtain

$$\frac{\partial}{\partial \xi}\left(M\frac{\partial \dot{x}}{\partial \xi}\right) = -\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right). \qquad \text{(MMPDE4)}$$

Further MMPDEs may be derived by making use of the concept of attraction and repulsion pseudoforces between mesh points [35]. A mesh point will attract other points if a measure of the truncation error is larger than the average measure, and vice versa. The error measure, denoted by $W$, is typically related to a given monitor function, often expressed as

$$W_i = \int_{x_i}^{x_{i+1}} M(\tilde{x}, t)d\tilde{x},$$

which can be interpreted as a discrete form of

$$W = M\frac{\partial x}{\partial \xi},$$

as both $W$ and $W_i$ can be approximated at point $x_i$ by using the midpoint rule: $W_i \approx M_{i+\frac{1}{2}}(x_{i+1} - x_i)$. Finally, the mesh speed is proportional to the rate of change of the error measure in the computational coordinate, giving

$$\frac{\partial x}{\partial t} = \frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M(x(\xi,t),t,u)\frac{\partial x}{\partial \xi}\right), \qquad \text{(MMPDE5)}$$

for some positive constant $\tau$. Examining this equation, we see that mesh points will move toward regions where the error is large, and will remain unmoving if $W$ is unchanging, that is, when the mesh is equidistributed.

Comparing the MMPDEs derived thus far, we see that both (MMPDE1) and (MMPDE2) involve the function $\frac{\partial M}{\partial t}$, causing them to be significantly more difficult to implement, as this term is often not easy to calculate. It has also been observed that the correction term in (MMPDE2) – (MMPDE5) not only forces the mesh toward equidistribution, but also prevents the mesh from crossing, that is, individual lines in a mesh trajectory cannot cross each other [34]. The parameter $\tau$ present in (MMPDE2) – (MMPDE5) is a timescale for the mesh to reach equidistribution [32]. This relaxation prevents oscillations from occurring in the time integration, hence producing smoother mesh trajectories. The appropriate value of $\tau$ for numerical calculations is generally problem dependent, but values of $10^{-k}$, for some small non-negative integer $k$, are commonly used [34]. The goal is to choose such a value of $\tau$ so that the mesh evolves at a rate proportional to that of the solution $u(x,t)$ [33].

However, it has been noted that maintaining a small constant value of $\tau$ may make the MMPDE unnecessarily stiff. To avoid this problem, after any transient rapid mesh movement due to significant changes in the behavior of $u(x,t)$, one can increase the value of $\tau$ without adversely affecting the quality of the solution [52]. One may instead use a variable $\tau(t)$, where $\tau(t) \in [\tau_{\min}, \tau_{\max}]$ is determined at each time by considering the size of $M(x,t)$ at this step. For further discussion, refer to [39, 52]. At this point we note that it is (MMPDE5) that will be used for discussion of time dependent problems in subsequent chapters.

## 2.3 Numerical Implementation

As mentioned previously, the mesh equation is coupled to the physical PDE via the monitor function. For example, given a PDE of the form $u_t = \mathcal{L}(u)$, where $\mathcal{L}$ is a spatial differential operator, we consider the coupled system

$$\frac{\partial u}{\partial t} = \mathcal{L}(u), \quad \frac{\partial x}{\partial t} = \frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M(x(\xi,t),t)\frac{\partial x}{\partial \xi}\right), \tag{2.8}$$

for $x \in [0,1]$ and $t \in (0,T]$. A review of various finite difference, finite element, collocation, and spectral methods previously used to solve this problem can be found in [7]. Of particular interest is the solver MOVCOL, described in detail in [36], in which (2.8) is semi-discretized in space using a collocation approach, and the resulting

system of ODEs is integrated in time using the solver DASSL [51]. In MOVCOL both the mesh $x^k$ and the solution $u^k$ at time $t^k$ are determined simultaneously by solving a single nonlinear system. In higher dimensions such an approach may be prohibitively computationally expensive. However, if we do not require the mesh to reach the same level of accuracy as the solution, less costly lower order methods can be used [2, 38]. Another way to reduce the computational cost of this system is to use an alternating approach [3, 7]. To step forward in time, the mesh equation and physical PDE are solved alternately, iterating until they agree within a user specified tolerance. This approach avoids the difficulty which the nonlinear coupling of mesh and solution would normally introduce, and can preserve other desirable features such as ellipticity and sparsity in the individual physical or mesh problems [7].

To discretize (2.4) or (MMPDE5) for numerical solution, we use centered finite difference spatial derivatives and the backward Euler method for time integration due to its simplicity and numerical stability. For (2.4) this leads to

$$\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) \approx \frac{1}{\Delta\xi}\left(M(x_{i+1/2})\left.\frac{dx}{d\xi}\right|_{x_{i+1/2}} - M(x_{i-1/2})\left.\frac{dx}{d\xi}\right|_{x_{i-1/2}}\right).$$

To replace the mesh density function $M(x)$ evaluated at half-nodes, we make the additional approximations

$$M(x_{i+1/2})\left.\frac{dx}{d\xi}\right|_{x_{i+1/2}} \approx \left(\frac{M(x_{i+1}) + M(x_i)}{2}\right)\left(\frac{x_{i+1} - x_i}{\Delta\xi}\right),$$

and

$$M(x_{i-1/2}) \left.\frac{dx}{d\xi}\right|_{x_{i-1/2}} \approx \left(\frac{M(x_i) + M(x_{i-1})}{2}\right)\left(\frac{x_i - x_{i-1}}{\Delta\xi}\right),$$

to obtain

$$\frac{[M(x_{i+1}) + M(x_i)]\,[x_{i+1} - x_i] - [M(x_i) + M(x_{i-1})]\,[x_i - x_{i-1}]}{2\Delta\xi^2} = 0,$$

for $i = 2, \ldots, N-1$, where $N$ is the number of mesh points. The Jacobian for this system is tridiagonal, with non-zero entries

$$J(i,j) = \begin{cases} \dfrac{[M(x_{i-1}) + M(x_i)] - M'(x_{i-1})\,[x_i - x_{i-1}]}{2\Delta\xi^2}, & \text{if } i-1 = j, \\ \dfrac{M'(x_i)\,[x_{i-1} - 2x_i + x_{i+1}] - [M(x_{i-1}) + 2M(x_i) + M(x_{i+1})]}{2\Delta\xi^2}, & \text{if } i = j, \\ \dfrac{[M(x_{i+1}) + M(x_i)] + M'(x_{i+1})\,[x_{i+1} - x_i]}{2\Delta\xi^2}, & \text{if } i+1 = j, \end{cases}$$

where $M' = \dfrac{dM}{dx}$. For (MMPDE5) we first semi-discretize in space to obtain the system of coupled ODEs

$$\frac{dx_i}{dt} = \frac{[M(x_{i+1}) + M(x_i)]\,[x_{i+1} - x_i] - [M(x_i) + M(x_{i-1})]\,[x_i - x_{i-1}]}{\tau \cdot 2\Delta\xi^2},$$

for $i = 2, \ldots, N-1$. Using backward Euler to handle the time derivatives and rearranging, we obtain

$$G_i^{n+1} := x_i^{n+1} - x_i^n - \frac{\Delta t}{\tau \cdot 2\Delta\xi^2}\left(\left[M(x_{i+1}^{n+1}) + M(x_i^{n+1})\right]\left[x_{i+1}^{n+1} - x_i^{n+1}\right] - \left[M(x_i^{n+1}) + M(x_{i-1}^{n+1})\right]\left[x_i^{n+1} - x_{i-1}^{n+1}\right]\right), \quad \text{for} \quad i = 2, \ldots, N-1.$$

The Jacobian of $G$ is once again tridiagonal, with nonzero entries for

$$\frac{\partial G_i^{n+1}}{\partial x_{i-1}^{n+1}}, \quad \frac{\partial G_i^{n+1}}{\partial x_i^{n+1}}, \quad \text{and} \quad \frac{\partial G_i^{n+1}}{\partial x_{i+1}^{n+1}}, \quad \text{for} \quad i = 2, \ldots, N-1.$$

These nonzero entries of the Jacobian are:

$$J(i,j) = \begin{cases} \dfrac{-\Delta t}{2\tau\Delta\xi^2}\left(\left[M(x_{i-1}^{n+1}) + M(x_i^{n+1})\right] - M'(x_{i-1}^{n+1})\left[x_i^{n+1} - x_{i-1}^{n+1}\right]\right), & \text{if } i-1=j, \\ 1 - \dfrac{\Delta t}{2\tau\Delta\xi^2}\left(M'(x_i^{n+1})\left[x_{i-1}^{n+1} - 2x_i^{n+1} + x_{i+1}^{n+1}\right] - \right. & \\ \quad \left.\left[M(x_{i-1}^{n+1}) + 2M(x_i^{n+1}) + M(x_{i+1}^{n+1})\right]\right), & \text{if } i=j, \\ \dfrac{-\Delta t}{2\tau\Delta\xi^2}\left(\left[M(x_{i+1}^{n+1}) + M(x_i^{n+1})\right] + M'(x_{i+1}^{n+1})\left[x_{i+1}^{n+1} - x_i^{n+1}\right]\right), & \text{if } i+1=j. \end{cases}$$

In practice, it has been observed that a lack of smoothness in the mesh transformation can affect the convergence of the iteration for the equidistributing mesh or prevent the physical solution from being accurately represented — see [39] as a general reference for the following discussion. To prevent this problem a common strategy is to use some type of spatial smoothing on the monitor function $M(x)$. As a result, a smoothed mesh density function $\widetilde{M}(x)$ will be used in place of the originally calculated function $M(x)$. In the continuous case, a smoothed mesh density function can be formulated as the solution to a boundary value problem (BVP). For a given mesh density function, $M$, a smoothed mesh density function, $\widetilde{M}$, can be obtained

by solving

$$\begin{aligned} \left(I - \beta^{-2}\frac{d^2}{d\xi^2}\right)\widetilde{M} &= M, \quad \xi \in (0,1), \\ \frac{d\widetilde{M}}{d\xi}(0) &= \frac{d\widetilde{M}}{d\xi}(1) = 0, \end{aligned} \tag{2.9}$$

where $\beta > 0$ is a parameter and $I$ the identity operator. If we discretize this BVP using centered finite differences, we obtain the system of algebraic equations

$$\begin{aligned} \widetilde{M}_j - \frac{1}{\beta^2\Delta\xi^2}(\widetilde{M}_{j+1} - 2\widetilde{M}_j + \widetilde{M}_{j-1}) &= M_j, \quad \text{for} \quad j = 2,\ldots,N-1, \\ \widetilde{M}_1 = \widetilde{M}_2, \quad \widetilde{M}_{N-1} &= \widetilde{M}_N. \end{aligned} \tag{2.10}$$

In both (2.9) and (2.10) the smoothing is a global process, as either a BVP or a linear system must be solved to determine $\widetilde{M}$. A local smoothing method may often be sufficient for a given problem — such methods can be obtained by expanding $\left(I - \beta^{-2}\frac{d^2}{d\xi^2}\right)\widetilde{M}$ as a series, truncating, and using appropriate finite difference approximations. More generally, when working with discretized equations, one can compute $\widetilde{M}$ as the weighted average of $M$ at nearby mesh points [7,34,39]. There are many possible ways to carry out this weighting, a reasonably representative example is given by

$$\widetilde{M}_i = \frac{\sum_{k=\min(1,i-p)}^{\max(N,i+p)} \gamma^{|j-k|} M_k}{\sum_{k=\min(1,i-p)}^{\max(N,i+p)} \gamma^{|j-k|}}, \quad \text{for} \quad i = 1,\ldots,N. \tag{2.11}$$

In (2.11) the parameter $p$ controls the number of terms in the sum, $p = 1$ or 2 being commonly used, and $\gamma \in (0, 1)$ controls the weight corresponding to each term in the sum.

It is also possible to incorporate the smoothing of the mesh density function directly into the EP, then derive smoothed versions of (2.4) and the MMPDEs. This eliminates the need to compute the smoothed mesh density function as an individual step in the solution process. For instance, the mesh transformation $x(\xi)$ for $\widetilde{M}$ will satisfy

$$\frac{d}{d\xi}\left(\widetilde{M}(x(\xi))\frac{d}{d\xi}x(\xi)\right) = 0,$$

where $\widetilde{M}$ is the solution to (2.9). That is,

$$\frac{d}{d\xi}\left(\left(I - \beta^{-2}\frac{d^2}{d\xi^2}\right)^{-1} M\frac{dx}{d\xi}\right) = 0.$$

Integrating both sides, isolating the inverse operator and then solving for $M$ we obtain

$$M = \theta\left(I - \beta^{-2}\frac{d^2}{d\xi^2}\right)\left(\frac{1}{\frac{dx}{d\xi}}\right),$$

where $\theta$ is the constant of integration. Solving for $\theta^{-1}$ and then differentiating with respect to $\xi$ leads to the smoothed equidistribution equation

$$\frac{d}{d\xi}\left[\frac{1}{M}\left(I - \beta^{-2}\frac{d^2}{d\xi^2}\right)\left(\frac{1}{\frac{dx}{d\xi}}\right)\right] = 0.$$

A similar approach can be used to obtain smoothed MMPDEs.

## 2.4 Moving Meshes in Higher Dimensions

The concept of moving mesh methods has been extended to two or more dimensions in a variety of ways: the review article [7] and the textbook [39] describe multiple ways to implement adaptivity in higher dimensions. As in 1D, the goal is to find a mesh transformation $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}) : \Omega_c \to \Omega$ which maps from a computational domain, $\Omega_c$, to best resolve some function $u(\boldsymbol{x})$ throughout the physical domain, $\Omega$.

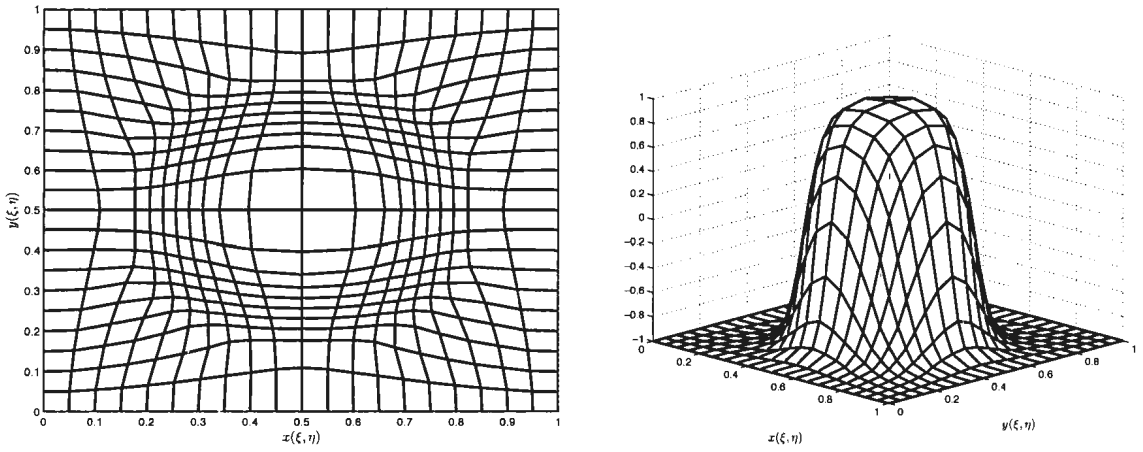


Figure 2.3: A 2D adaptive mesh (left) and the function $u(x, y)$ being equidistributed plotted using this mesh (right).

An example of 2D mesh equidistribution is given in Figure 2.3, where we determine an equidistributing mesh for the function

$$u(x, y) = \tanh\left[35\left(\frac{1}{16} - \left(x - \frac{1}{2}\right)^2 - \left(y - \frac{1}{2}\right)^2\right)\right].$$

Mesh points are concentrated in an annulus containing the circle $(x-\frac{1}{2})^2+(y-\frac{1}{2})^2 = \frac{1}{4}^2$ where $u(x,y)$ undergoes rapid variations.

In multiple dimensions the EP, which will determine the volume of mesh elements, is no longer sufficient to fully determine the mesh. An additional alignment condition, which ensures the mesh is aligned with the behavior of the physical solution, is used to specify shape and orientation of the mesh elements. A matrix valued function $M(\boldsymbol{x})$ called the *monitor function* is used to define a metric on $\Omega$, and $\rho = \sqrt{\det(M(\boldsymbol{x}))}$ is the corresponding mesh density function. As in the 1D case the appropriate choice of monitor function $M$, and hence mesh density function $\rho$, is key to the success of any adaptive method — the details are presented in [39]. If $K$ is a mesh element and $\mathcal{J}_h$ is the mesh over $\Omega$, then equidistribution requires that

$$\int_K \rho(\boldsymbol{x})d\boldsymbol{x} = \frac{\sigma}{N}, \quad \forall K \in \mathcal{J}_h,$$

where there are $N$ elements in $\mathcal{J}_h$ and $\sigma = \int_\Omega \rho(\boldsymbol{x})d\boldsymbol{x}$, that is, each mesh element will have the same volume in the metric. The other condition is that the mesh elements must be equilateral in the metric $M$. If $\gamma_1, \ldots, \gamma_{d(d+1)/2}$ are the edges of an element $K$ and $|\gamma_i|_M$ is the length of edge $\gamma_i$ in the metric $M$, then this condition requires

$$|\gamma_1|_M = \cdots = |\gamma_{d(d+1)/2}|_M, \quad \forall K \in \mathcal{J}_h.$$

These conditions can be simplified to obtain approximations used for numerical im-

plementation, and can also be described in terms of a continuous mesh transformation for theoretical analysis.

A second common approach is based on the calculus of variations, in which $\boldsymbol{\xi}(\boldsymbol{x})$, the inverse mapping of $\boldsymbol{x}(\boldsymbol{\xi})$, is determined to be the function which minimizes the functional

$$I[\boldsymbol{\xi}] = \int_{\Omega} F(\nabla\boldsymbol{\xi}, \boldsymbol{\xi}, \boldsymbol{x}) d\boldsymbol{x},$$

where $F$ can take on numerous different forms, see [3, 7, 33, 37–39] for examples of this. The mesh transformation in the steady case can be obtained by solving the Euler-Lagrange equations

$$-\nabla \cdot \left[ \frac{\partial F}{\partial \boldsymbol{a}^i} - J \frac{\partial F}{\partial J} \boldsymbol{a}_i \right] = 0, \quad \text{for} \quad i = 1, 2, 3, \tag{2.12}$$

where $\boldsymbol{a}^i = \nabla\xi_i$, $\boldsymbol{a}_i = \frac{\partial \boldsymbol{x}}{\partial \xi_i}$ and $J$ is the Jacobian, $\nabla_{\boldsymbol{\xi}}\boldsymbol{x}$. MMPDEs can be obtained by modifying (2.12) in a similar fashion to the 1D EP:

$$\frac{\partial \xi_i}{\partial t} = \frac{1}{\tau p(\boldsymbol{x}, t)} \nabla \cdot \left[ \frac{\partial F}{\partial \boldsymbol{a}^i} - J \frac{\partial F}{\partial J} \boldsymbol{a}_i \right], \quad \text{for} \quad i = 1, 2, 3, \tag{2.13}$$

where $\tau > 0$ is a parameter controlling the time scale of mesh movement and $p(\boldsymbol{x}, t)$ is described as a balancing function which will hopefully allow all mesh points to move on the same time scale; the appropriate choice of the balancing function vary with the problem and the formulation of the mesh equations. Finally, the mesh equation

for the computational coordinate (2.13) can be transformed to obtain a MMPDE for the physical mesh,

$$\dot{\boldsymbol{x}} = -\frac{1}{\tau p(\boldsymbol{x},t)} \sum_i \boldsymbol{a}_i \nabla \cdot \left[ \frac{\partial F}{\partial \boldsymbol{a}^i} - J \frac{\partial F}{\partial J} \boldsymbol{a}_i \right].$$

Another method of mesh adaptation based on the concept of local equidistribution in two dimensions has been proposed in [40]. As before, we let $\boldsymbol{x} = [x, y]^T$ be the spatial coordinates of a mesh in a 2D physical domain, $\Omega$. We introduce the coordinate transformation $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$, where $\boldsymbol{\xi} = [\xi, \eta]^T$ denotes the spatial coordinates on the computational domain, $\Omega_c = [0,1] \times [0,1]$. In contrast to other methods where the quantity being equidistributed can be changed, we focus solely on how to equidistribute the arc-length measure of a function $u(x,y)$ over $\Omega$. Specifically, we consider a scaled arc-length measurement of variation of $u$ along the arc element from $\boldsymbol{x}$ to $\boldsymbol{x} + d\boldsymbol{x}$, which can be expressed as

$$ds = [a^2(du)^2 + d\boldsymbol{x}^T d\boldsymbol{x}]^{1/2} = [d\boldsymbol{x}^T M d\boldsymbol{x}]^{1/2}, \tag{2.14}$$

where $M = a^2 \nabla u \cdot \nabla u^T + I$, and $I$ is the identity matrix. Making use of the mesh transformation $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$, (2.14) can be expressed as

$$ds = [d\boldsymbol{\xi}^T J^T M J d\boldsymbol{\xi}]^{1/2}, \tag{2.15}$$

where $J$ is the Jacobian of the transformation.

The EP follows from (2.15): if $u(\boldsymbol{x}(\boldsymbol{\xi}))$ is to be the constant value $ds$ along any arc element in the computational domain of fixed length $[d\boldsymbol{\xi}^T d\boldsymbol{\xi}]^{1/2}$, then (2.15) must be independent of $\boldsymbol{\xi}$. This implies that $J^T M J$ should be independent of $\boldsymbol{\xi}$, or

$$[d\boldsymbol{\xi}^T\, J^T\, M\, J\, d\boldsymbol{\xi}]^{1/2} = [d\boldsymbol{\xi}^T\, \widetilde{M}\, d\boldsymbol{\xi}]^{1/2}, \tag{2.16}$$

where $\widetilde{M}$ is a constant, $\boldsymbol{\xi}$-independent, and symmetric positive definite matrix. If a coordinate transformation can be found which satisfies (2.16), $u$ will have the same variation at any point in $\Omega$ along any arc of length

$$\left[\left(\tfrac{\partial \boldsymbol{x}}{\partial \xi} d\xi + \tfrac{\partial \boldsymbol{x}}{\partial \eta} d\eta\right)^T \left(\tfrac{\partial \boldsymbol{x}}{\partial \xi} d\xi + \tfrac{\partial \boldsymbol{x}}{\partial \eta} d\eta\right)\right]^{1/2}.$$

Usually (2.16) cannot be satisfied by the coordinate transformation on the whole computational domain when the number of nodes along a given coordinate line is fixed, which is the case for moving mesh methods [40]. However, if we weaken this condition and only require the transformation to satisfy (2.16) locally; that is, only require $\widetilde{M}$ to be constant along a given coordinate line; it is possible to find a local equidistribution on $\Omega$. This leads to the system of equations

$$\begin{aligned}
\left[\begin{pmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \end{pmatrix}^T M \begin{pmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \end{pmatrix}\right]^{1/2} &= c_1(\eta), \\
\left[\begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \end{pmatrix}^T M \begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \end{pmatrix}\right]^{1/2} &= c_2(\xi).
\end{aligned} \tag{2.17}$$

The unknown $c_1(\eta)$ is a constant for each $\xi$ and $c_2(\xi)$ is constant for each $\eta$. They can remain unspecified, as they will be eliminated in the numerical implementation of this system. In practice, instead of the scaled arc-length matrix $M$, we use

$$M = \frac{a^2 \nabla u \cdot \nabla u^T}{1 + b \nabla u^T \nabla u} + I,$$

where the parameter $b \geq 0$ is used to prevent situations where extremely small mesh spacing or mesh tangling could occur, that is, when $|\nabla u|$ is very large.

In [40] a combination of Dirichlet and Neumann conditions are used along $\partial\Omega_c$

$$x(0,\eta) = y(\xi,0) = 0, \quad x(1,\eta) = y(\xi,1) = 1, \tag{2.18}$$

$$\frac{\partial x}{\partial \eta}(\xi,0) = \frac{\partial x}{\partial \eta}(\xi,1) = \frac{\partial y}{\partial \xi}(0,\eta) = \frac{\partial y}{\partial \xi}(1,\eta) = 0, \tag{2.19}$$

where $\xi, \eta \in [0,1]$. The Dirichlet conditions are consistent with the requirement that there are mesh points on the boundary of the domain. The Neumann orthogonality conditions are arbitrary, and in fact can cause smoothness issues near the domain boundaries. As an alternative, we follow [46] and apply the 1D EP, (2.4), to determine $x(\xi,0)$, $x(\xi,1)$, $y(0,\eta)$ and $y(1,\eta)$.

### 2.4.1 Numerical Implementation

To implement system (2.17) numerically, we consider a uniform computational mesh over the $\Omega_c = [0,1] \times [0,1]$, with points $(\xi_i, \eta_j)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$,

such that $\xi_1 = \eta_1 = 0$ and $\xi_n = \eta_m = m$. Discretizing the equations of (2.17) at the half-nodes, $(i + \frac{1}{2}, j)$ and $(i, j + \frac{1}{2})$, we obtain

$$\left[ \begin{pmatrix} x_{i+1,j} - x_{i,j} \\ y_{i+1,j} - y_{i,j} \end{pmatrix}^T M_{i+\frac{1}{2},j} \begin{pmatrix} x_{i+1,j} - x_{i,j} \\ y_{i+1,j} - y_{i,j} \end{pmatrix} \right]^{1/2} = c_1(\eta_j),$$

for $i = 1, \ldots, n-1$ and $j = 2, \ldots, m-1$; and

$$\left[ \begin{pmatrix} x_{i,j+1} - x_{i,j} \\ y_{i,j+1} - y_{i,j} \end{pmatrix}^T M_{i,j+\frac{1}{2}} \begin{pmatrix} x_{i,j+1} - x_{i,j} \\ y_{i,j+1} - y_{i,j} \end{pmatrix} \right]^{1/2} = c_2(\xi_i),$$

for $i = 2, \ldots, n-1$ and $j = 1, \ldots, m-1$, where $M_{i,j} = M(\xi_i, \eta_j)$. The unknown constants $c_1$ and $c_2$ are eliminated by computing differences of these equations for adjacent indices, leading to

$$\begin{aligned} &\left[ \begin{pmatrix} x_{i,j} - x_{i-1,j} \\ y_{i,j} - y_{i-1,j} \end{pmatrix}^T M_{i-\frac{1}{2},j} \begin{pmatrix} x_{i,j} - x_{i-1,j} \\ y_{i,j} - y_{i-1,j} \end{pmatrix} \right]^{1/2} \\ &- \left[ \begin{pmatrix} x_{i+1,j} - x_{i,j} \\ y_{i+1,j} - y_{i,j} \end{pmatrix}^T M_{i+\frac{1}{2},j} \begin{pmatrix} x_{i+1,j} - x_{i,j} \\ y_{i+1,j} - y_{i,j} \end{pmatrix} \right]^{1/2} = 0, \end{aligned} \tag{2.20}$$

and

$$\left[\begin{pmatrix} x_{i,j}-x_{i,j-1} \\ y_{i,j}-y_{i,j-1} \end{pmatrix}^T M_{i,j-\frac{1}{2}} \begin{pmatrix} x_{i,j}-x_{i,j-1} \\ y_{i,j}-y_{i,j-1} \end{pmatrix}\right]^{1/2} - \left[\begin{pmatrix} x_{i,j+1}-x_{i,j} \\ y_{i,j+1}-y_{i,j} \end{pmatrix}^T M_{i,j+\frac{1}{2}} \begin{pmatrix} x_{i,j+1}-x_{i,j} \\ y_{i,j+1}-y_{i,j} \end{pmatrix}\right]^{1/2} = 0, \tag{2.21}$$

both for $i = 2, \ldots, n-1$ and $j = 2, \ldots, m-1$. These equations enforce the equidistribution on the interior mesh points of the computational domain.

To discretize the matrix $M$ at half nodes, we note that

$$\begin{bmatrix} u_\xi \\ u_\eta \end{bmatrix} = \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix},$$

and by computing the inverse of this matrix, we obtain

$$\nabla u = \begin{bmatrix} \dfrac{u_\xi y_\eta - u_\eta y_\xi}{x_\xi y_\eta - x_\eta y_\xi}, & \dfrac{-u_\xi x_\eta + u_\eta x_\xi}{x_\xi y_\eta - x_\eta y_\xi} \end{bmatrix}^T. \tag{2.22}$$

To discretize (2.22) at the points $(\xi_{i+1/2}, \eta_{j+1/2})$, discretizations for derivatives of $x$, $y$, and $u$ with respect to $\xi$ and $\eta$ are required. These are obtained by central differences — for example

$$x_\xi(\xi_{i+1/2}, \eta_{i+1/2}) = \frac{1}{2}\left(x_{i+1,j} - x_{i,j} + x_{i+1,j+1} - x_{i,j+1}\right),$$
$$x_\eta(\xi_{i+1/2}, \eta_{i+1/2}) = \frac{1}{2}\left(x_{i,j+1} - x_{i,j} + x_{i+1,j+1} - x_{i+1,j}\right),$$

with similar expressions for $y$ and $u$. Expressions for $M$ at the $(i+\frac{1}{2}, j)$ and $(i, j+\frac{1}{2})$ nodes are the averages

$$M_{i+\frac{1}{2},j} = \frac{1}{2}\left(M_{i+\frac{1}{2},j+\frac{1}{2}} + M_{i+\frac{1}{2},j-\frac{1}{2}}\right) \quad \text{and} \quad M_{i,j+\frac{1}{2}} = \frac{1}{2}\left(M_{i+\frac{1}{2},j+\frac{1}{2}} + M_{i-\frac{1}{2},j+\frac{1}{2}}\right).$$

As in the 1D case, it has been noted that smoothing the matrix $M$ can significantly improve the quality of the mesh determined by the system of discretized equations [40]. This is carried out by replacing the matrices $M$ in (2.20) and (2.21) by the weighted averages

$$\widetilde{M}_{i+\frac{1}{2},j} = \sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1} M_{k+\frac{1}{2},l}\left(\frac{\gamma}{1+\gamma}\right)^{|k-i|+|l-j|},$$

$$\widetilde{M}_{i,j+\frac{1}{2}} = \sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1} M_{k,l+\frac{1}{2}}\left(\frac{\gamma}{1+\gamma}\right)^{|k-i|+|l-j|}.$$

Here $\gamma$ is a positive constant, the scaling parameter, and the summations are understood to only include values which are well defined.

Having provided an introduction to mesh equidistribution, we next turn to the other method of interest, domain decomposition, in Chapter 3.

# Chapter 3

# Domain Decomposition

In this chapter we introduce the concept of domain decomposition methods for BVPs. We begin by considering the historical origin of what are known as Schwarz methods, describing how such methods are implemented for continuous problems as well as linear systems. We then discuss two common approaches used to extend DD methods for elliptic problems to time dependent parabolic problems.

## 3.1 Domain Decomposition for Time Independent Problems

Domain decomposition (DD) is a method which is used to facilitate the solution of PDEs by introducing spatial parallelism. In the DD approach, the spatial domain $\Omega$ of a given BVP is partitioned into overlapping or non-overlapping subdomains, and we solve the original PDE on each of the subdomains, resulting in a system of coupled BVPs. To complete these new BVPs, transmission conditions are introduced along the artificial boundaries to ensure that solutions on adjacent subdomains agree with each other, and that they reproduce the solution to the original problem in a piecewise manner. Through DD, the large systems of equations which result when solving high dimensional PDEs can be replaced by multiple systems of smaller size, being less costly to solve. Furthermore, by appropriately choosing the transmission conditions the subdomain problems can be solved in parallel, thus DD lends itself to implementation on parallel computer architecture.

### 3.1.1 Classical Schwarz

The origin of DD can be traced to the work of H. A. Schwarz in 1869, who sought to provide a rigorous proof of the Dirichlet principle — that if a function $u(x)$ is

a solution to Laplace's equation on a bounded domain $\Omega$ with boundary conditions $u = g$ on $\partial\Omega$, then $u$ is the infimum of the Dirichlet integral $\int_\Omega |\nabla v|^2 dx$ over all functions $v$ satisfying $v = g$ on $\partial\Omega$ — for an arbitrary domain $\Omega$ [18].

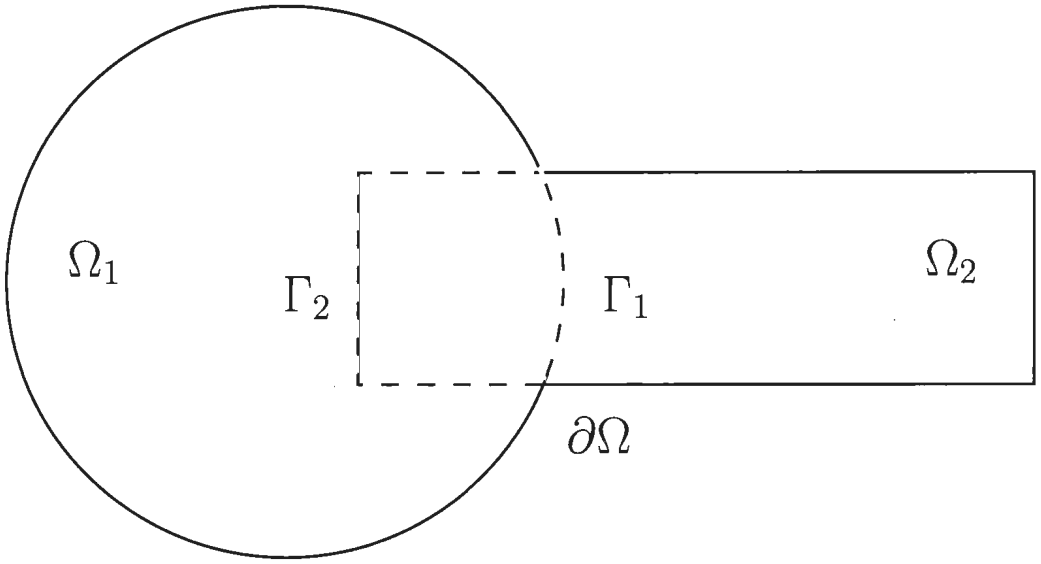


Figure 3.1: The original "complicated" domain considered by Schwarz, the combination of a circle and a rectangle.

The original domain $\Omega$ considered by Schwarz, pictured in Figure 3.1, is a combination of two simple subdomains, a circle $\Omega_1$ and a rectangle $\Omega_2$. The artificial boundaries introduced are $\Gamma_1 = \partial\Omega_1 \cap \Omega_2$ and $\Gamma_2 = \partial\Omega_2 \cap \Omega_1$. The iteration proposed by Schwarz required an initial guess along $u_2^0$ along $\Gamma_1$, where superscript denotes

iteration number, and was then calculated according to the iteration: for $n = 0, 1, \ldots$

$$\Delta u_1^{n+1} = 0 \text{ in } \Omega_1, \qquad \Delta u_2^{n+1} = 0 \text{ in } \Omega_2,$$

$$u_1^{n+1} = u_2^n \text{ on } \Gamma_1, \qquad u_2^{n+1} = u_1^{n+1} \text{ on } \Gamma_2.$$

We note that $u_1^n$ and $u_2^n$ must also satisfy the posed Dirichlet condition along the outer boundaries of $\Omega_1$ and $\Omega_2$ [18]. This iteration is the origin of what is now called the *alternating Schwarz method*, so called because the subdomain problems are solved in an alternating fashion: first over $\Omega_1$, then $\Omega_2$, $\Omega_1$, $\Omega_2$, and so on. The alternating Schwarz method can be extended to any number of subdomains, so long that care is taken to use the most recent solution information is used along the artificial boundaries.

To implement the alternating Schwarz method for a general time independent problem

$$\mathcal{L}u = f, \quad x \in \Omega, \tag{3.1}$$

where $\mathcal{L}$ is a spatial differential operator, we decompose $\Omega$ into subdomains $\{\Omega_i\}_1^N$ such that

$$\bigcup \Omega_i = \Omega,$$

and implement the iteration: for $n = 0, 1, 2, \ldots$

$$\mathcal{L}u_i^{n+1} = f, \quad x \in \Omega_i, \tag{3.2}$$

$$u_i^{n+1} = u_j^{n+1_{ij}}, \quad x \in \Gamma_{ij}, \tag{3.3}$$

where in the superscript $1_{ij} = 1$ for $i > j$, zero otherwise, and $\Gamma_{ij} = \partial\Omega_i \cap \Omega_j$. To avoid complications arising from multiple choices of boundary conditions in subsequent discussion, we only consider partitionings of $\Omega$ such that at most two subdomains contain any given point of the original domain. For example, in one dimension $\Omega$ is an interval $[a, b]$, and subdomains are intervals satisfying

$$\Omega_i = [\alpha_i, \beta_i], \quad \text{for} \quad i = 1, \ldots, N,$$

where $\alpha_1 = a$, $\beta_N = b$, and

$$\alpha_i < \beta_{i-1} < \alpha_{i+1}, \quad \text{for} \quad i = 2, \ldots, N-1.$$

An important extension to the alternating Schwarz method is the *parallel Schwarz method*, first proposed by Lions [42], in which each subdomain problem will use information from the previous iteration along the artificial boundaries, and so all subdomain problems can be solved simultaneously. An obvious analogy to make is to compare the alternating and parallel Schwarz methods to the Gauss-Seidel and Jacobi iterations for linear systems, where each linear equation can either be solved in sequence or in parallel depending on whether the most recent information is used. If $\Omega$ is

decomposed into $\Omega_1$ and $\Omega_2$ and we denote the nonempty intersections $\partial\Omega_1 \cap \Omega_2 = \Gamma_1$ and $\partial\Omega_2 \cap \Omega_1 = \Gamma_2$, the parallel iteration is: for $n = 0, 1, \ldots$

$$\mathcal{L}u_1^{n+1} = f, \quad x \in \Omega_1, \qquad \mathcal{L}u_2^{n+1} = f, \quad x \in \Omega_2,$$

$$u_1^{n+1} = u_2^n, \quad x \in \Gamma_1, \qquad u_2^{n+1} = u_1^n, \quad x \in \Gamma_2.$$

Similarly, if we decompose $\Omega$ into arbitrarily many subdomains we obtain the iteration: for $i = 1, \ldots, N$ and for $n = 0, 1, \ldots$

$$\mathcal{L}u_i^{n+1} = f, \quad x \in \Omega_i, \tag{3.4}$$

$$u_i^{n+1} = u_j^n, \quad x \in \Gamma_{ij}. \tag{3.5}$$

Taken together, the alternating and parallel Schwarz methods comprise what are known as the *classical Schwarz methods.* They are characterized by their use of Dirichlet transmission conditions between subdomains and the fact that subdomains must overlap, that is, adjacent subdomains must share more points than the intersection of their respective boundaries. If subdomains do not overlap, the Dirichlet conditions will only impose continuity at the artificial boundaries, not smoothness, hence the DD iteration will typically not generate a smooth solution, thus it will not agree with the single domain solution. It is also possible for classical iterations to fail to converge for certain equations, even when overlap is used — the example of the indefinite Helmholtz equation is given in [18]. Another common failing of the classical

Schwarz methods are their typically slow rates of convergence for smaller amounts of overlap between subdomains. However, there is a way to overcome all of the above failings, which is to modify the transmission conditions used between subdomains, leading to what are known as *optimized Schwarz methods.*

### 3.1.2 Optimized and Optimal Schwarz

The origin of optimized Schwarz methods can be traced to the work [43] by Lions, in which he proposed the use of Robin transmission conditions, linear combinations of function value and derivative value, to overcome the requirement of overlap for the classical Schwarz DD iteration to converge. As before, $\Omega$ is decomposed into $\Omega_1$ and $\Omega_2$; but we now allow the possibility that the subdomains only share a common boundary — that is, $\Gamma_1 = \Gamma_2 = \Gamma$. The parallel optimized Schwarz iteration is of the form

$$\begin{aligned} \mathcal{L}u_1^{n+1} &= f, \quad x \in \Omega_1, \qquad (\partial_{n_1} + p_1)u_1^{n+1} = (\partial_{n_1} + p_1)u_2^n, \quad x \in \Gamma_1, \\ \mathcal{L}u_2^{n+1} &= f, \quad x \in \Omega_2, \qquad (\partial_{n_2} + p_2)u_2^{n+1} = (\partial_{n_2} + p_2)u_1^n, \quad x \in \Gamma_2, \end{aligned} \tag{3.6}$$

where $\partial_{n_i}$ denotes the partial derivative normal to the boundary of domain $i$, and $p_i$ are constant parameters [18,43]. While the rate of convergence for optimized Schwarz iterations depend on the choice of parameters $p_i$, optimized Schwarz methods for elliptic problems have been shown to converge for an arbitrary number of subdomains

which only share common boundaries [18]. Removing this need for overlap allows optimized Schwarz methods to be applied to problems which contain discontinuous coefficients, interfaces between different media, and other cases where overlap would not be part of a natural decomposition. Furthermore, optimized Schwarz methods converge significantly faster than classical Schwarz iterations for little to no extra computational cost, and for many types of problems, there may be simple optimization procedures or closed form solutions for the best choice of transmission condition parameters [17].

The convergence of the Schwarz DD approach can be further improved by using more general transmission conditions, the best possible choice of conditions for a given problem produce what are known as *optimal Schwarz methods.* By doing so, the parallel Schwarz DD iteration will take the form

$$\mathcal{L}u_1^{n+1} = f, \quad x \in \Omega_1, \qquad \mathcal{L}u_2^{n+1} = f, \quad x \in \Omega_2,$$

$$\mathcal{B}_1 u_1^{n+1} = \mathcal{B}_1 u_2^n, \quad x \in \Gamma_1, \qquad \mathcal{B}_2 u_2^{n+1} = \mathcal{B}_2 u_1^n, \quad x \in \Gamma_2,$$

where $\mathcal{B}_i$ is an operator acting along the interface between subdomains. The optimal choice of $\mathcal{B}_i$ is dependent on the problem being solved. For a large class of second-order problems, with domains decomposed into strips, the optimal choice of transmission condition $\mathcal{B}_i$ is of the form $\partial_{n_j} + DtN_j$, where $DtN_j$ denotes a non-local Dirichlet to Neumann or Steklov-Poincaré operator associated with the elliptic operator $\mathcal{L}$ [18,48].

These optimal operators are often non-local, hence more costly to evaluate and more difficult to implement. To avoid this extra difficulty, it is common to approximate optimal transmission conditions by local operators of the form $\mathcal{B}_i = \partial_{n_j} + p_j + r_j\partial_\tau + q_j\partial_{\tau\tau}$, where $\partial_\tau$ is the tangential derivative at the interface between subdomains [18]. The original optimized Schwarz transmission conditions of (3.6) correspond to taking $r_j = q_j = 0$, and can be thought of as a zeroth order approximation, with higher order approximations being obtained with each subsequent term kept. Further discussion on optimized and optimal Schwarz methods for various problems can be found in [17] and many subsequent papers by Gander.

### 3.1.3 DD For Linear Systems

The alternating and parallel Schwarz iterations of (3.2–3.3) and (3.4–3.5) are formulated for continuous problems. After discretizing an elliptic PDE such as (3.1) we obtain a linear system of the form $A\boldsymbol{u} = \boldsymbol{f}$, so a natural extension of the DD methods already discussed are Schwarz methods applied directly to linear systems [18, 55]. To decompose the vector $\boldsymbol{u}$ into subsets one can use restriction operators. If $A$ is an $n \times n$ matrix, then a restriction matrix $R$ is a $k \times n$ matrix, identically zero except for exactly $k$ ones, one per row, with $R_{ij} = 1$ indicating that $u_j$ is the $i^{\text{th}}$ variable of the

resulting subset. For instance, to decompose a vector into two subsets one can take

$$R_1 = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}_{k\times n} \quad \text{and} \quad R_2 = \begin{bmatrix} & 1 & \\ & & \ddots \\ & & & 1 \end{bmatrix}_{\ell\times n},$$

which take the first $k$ and last $\ell$ elements of $\boldsymbol{u}$ as subsets, respectively. The restriction of the matrix $A$ to a given subset can be similarly obtained using these restriction matrices, defining

$$A_j = R_j A R_j^T, \quad \text{for} \quad j = 1, 2.$$

We can now define the *multiplicative Schwarz method* for linear systems, given by:

$$\boldsymbol{u}^{n+\frac{1}{2}} = \boldsymbol{u}^n + R_1^T A_1^1 R_1(\boldsymbol{f} - A\boldsymbol{u}^n),$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n+\frac{1}{2}} + R_2^T A_2^1 R_2(\boldsymbol{f} - A\boldsymbol{u}^{n+\frac{1}{2}}).$$

Similar to the alternating Schwarz method, we first solve a system using the matrix $A_1$ associated with the first subset of variables, then a second system using $A_2$, which is associated with the second set of variables. Indeed, if one considers the case where the $R_j$ are non-overlapping, that is, $R_1^T R_1 + R_2^T R_2 = I$, then it can be shown that the multiplicative Schwarz algorithm is equivalent to the systems obtained by discretizing the alternating Schwarz algorithm [18]. More generally, if we decompose into $J$ subdomains, we obtain the iteration

$$\boldsymbol{u}^{n+\frac{j}{J}} = \boldsymbol{u}^{n+\frac{j-1}{J}} + R_j^T A_j^{-1} R_j(\boldsymbol{f} - A\boldsymbol{u}^{n+\frac{j-1}{J}}), \quad \text{for} \quad j = 1, 2, \ldots, J.$$

As the multiplicative Schwarz method can be thought of as an analog to the alternating Schwarz method, an obvious question is to ask if there is a Schwarz method for linear systems which can be implemented in parallel. One answer to this question is the *additive Schwarz method*, in which the matrix $A$ is preconditioned by a matrix $A_J^{-1}$, where $A_J$ is a direct sum of diagonal blocks of $A$ [55]. For reference, the direct sum of matrices $A$ and $B$ is denoted by $A \oplus B$ and is given by

$$A \oplus B = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}.$$

In the case of two subdomains, the preconditioning matrix $A_J^{-1}$ is given by

$$A_J^{-1} = A_1^{-1} \oplus A_2^{-1} = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix},$$

or equivalently $R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2$, which leads to the additive Schwarz fixed point iteration [18]

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + (R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2)(\boldsymbol{f} - A\boldsymbol{u}^n).$$

Clearly the linear solves on the right hand side can be done simultaneously, allowing parallel implementation. Indeed, in the case of non-overlapping $R_j$, this iteration is equivalent to the systems obtained by discretizing the overlapping parallel Schwarz algorithm. While this two subdomain iteration can be extended in an obvious fashion to

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \sum_{j=1}^{J} R_j^T A_j^{-1} R_j (\boldsymbol{f} - A\boldsymbol{u}^n), \tag{3.7}$$

it is interesting to note that a non-convergence result has been established, stating that in general the iteration (3.7) cannot converge (See Theorem 3.3 of [18]). There are several ways to overcome this limitation, and furthermore, it has been observed that the preconditioned system obtained by applying $\sum_{j=1}^{J} R_j^T A_j^{-1} R_j$ lends itself to solution by Krylov methods. Further details on various discrete DD methods can be found in [18, 55].

## 3.2 DD for Time Dependent Problems

In the previous section we considered Schwarz DD methods for the solution of steady PDEs. Indeed, there is a vast body of literature of DD methods for the solution of elliptic problems, see for example [47, 55]. DD has also been applied to the solution of time dependent problems of both parabolic and hyperbolic varieties. In particular, Schwarz methods have been applied to time dependent problems using two different approaches, both of which will be discussed in this section. To illustrate these methods we use the sample parabolic problem

$$u_t = \mathcal{L}u, \quad (x,t) \in \Omega \times [0,T], \tag{3.8}$$

subject to appropriate initial and boundary conditions.

### 3.2.1 Solving a Sequence of Elliptic Problems

The classical method for solving time dependent problems using DD is due to Cai [8,9], in which existing alternating or parallel Schwarz methods are applied to the sequence of elliptic problems which result from a time semi-discretization of the PDE. For (3.8), discretizing by the backward Euler method leads to the sequence of elliptic problems

$$u(t_{k+1}) = u(t_k) + \Delta t \mathcal{L} u(t_{k+1}), \quad x \in \Omega, \quad \text{for} \quad k = 1, 2, \ldots, N, \tag{3.9}$$

where $u(t_k)$ denotes the solution at time step $k$, setting $t_0 = 0$ and $t_N = T$. At time step $k$, (3.9) is a simple inhomogeneous elliptic problem, with a time independent source term $u(t_k)$. By applying DD for each $k$, suppressing the argument $t_{k+1}$, a general parallel Schwarz iteration is: for $n = 0, 1, \ldots$

$$u_1^{n+1} = u(t_k) + \Delta t \mathcal{L} u_1^{n+1}, \quad x \in \Omega_1, \qquad \mathcal{B}_1 u_1^{n+1} = \mathcal{B}_1 u_2^n, \quad x \in \Gamma_1,$$

$$u_2^{n+1} = u(t_k) + \Delta t \mathcal{L} u_2^{n+1}, \quad x \in \Omega_2, \qquad \mathcal{B}_2 u_2^{n+1} = \mathcal{B}_2 u_1^n, \quad x \in \Gamma_2.$$

For this method the DD iteration is an "inner loop": at each time step in the solution of (3.9), we perform parallel Schwarz iterations until convergence is achieved, then proceed to the next time step, where the process is repeated. We note that optimized transmission conditions $\mathcal{B}_i$ for the corresponding elliptic problem can be applied for this iteration, see [17] and the references therein.

### 3.2.2 Schwarz Waveform Relaxation

In the more recent method, known as *Schwarz waveform relaxation* (SWR), the space-time domain is decomposed and the time dependent PDE is solved on each spatial subdomain for the entire time interval [5, 25, 26]. The two subdomain DD iteration for (3.9) is: for $n = 0, 1, \ldots$

$$u_{1,t}^{n+1} = \mathcal{L}u_1^{n+1}, \quad (x,t) \in \Omega_1 \times [0,T],$$

$$\mathcal{B}_1 u_1^{n+1} = \mathcal{B}_1 u_2^n, \quad (x,t) \in \Gamma_1 \times [0,T],$$

$$u_{2,t}^{n+1} = \mathcal{L}u_2^{n+1}, \quad (x,t) \in \Omega_2 \times [0,T],$$

$$\mathcal{B}_2 u_2^{n+1} = \mathcal{B}_2 u_1^n, \quad (x,t) \in \Gamma_2 \times [0,T].$$

For this pair of sub-problems the DD is now an "outer loop" in the iterative solution of the PDE. For each DD iteration, both subdomain problems are solved over their entire domain, $\Omega_i \times [0,T]$. Transmission conditions along the interface between subdomains are updated for all $t \in [0,T]$, then the process repeats. A key advantage of the SWR iteration over solving a sequence of elliptic problems is that the user is no longer required to maintain a uniform time step, or even use the same integration technique, for different subdomains, allowing much more flexibility to adjust to each individual sub-problem [21, 32]. Another advantage is in terms of parallel implementation, as fewer transmissions of larger amounts of data can be more efficient than many transmissions of smaller amounts of data by avoiding the "startup" cost each

communication incurs, regardless of transmission size.

The SWR method can be extended to an arbitrary number of subdomains, and convergence results have been established for the heat equation [5, 25], the wave equation [21] and for various other classes of problems, in many cases with optimized transmission conditions proposed [4, 19–21, 24].

Having presented the basics of DD in this Chapter, we will now discuss how DD can be applied to the solution of mesh equidistribution problems in Chapter 4.

# Chapter 4

# DD Methods for the Steady 1D Mesh Equation

After introducing the concepts of moving mesh methods in Chapter 2 and DD in Chapter 3, we now turn to the main focus of this thesis: the coupling of these methods together, establishing the convergence of Schwarz methods for the 1D equidistribution principle (2.4). This work follows from the the experimental papers [29, 31, 32] and builds upon the theoretical work of [22,23], with some of the original results previously included in [30]. Due to the nonlinearity of (2.4) much of the analysis presented will differ from typical strategies used in DD literature. Indeed, much less has been written about DD for nonlinear problems, see [6, 10, 15, 16, 24, 42, 44, 45, 53].

As in the papers [22, 23, 29, 31], we are primarily concerned with the mesh generation problem, less the solution of the coupled problem (2.8), and so we assume the physical solution $u$ is known. This is a reasonable assumption, given the usual approach of decoupling the physical equation and mesh equation mentioned in Chapter 2. If $u$ is a time independent function on $\Omega = [0, 1]$, then from (2.4) an equidistributing mesh transformation $x(\xi) : \Omega_c \to \Omega$ is determined by solving the BVP

$$\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) = 0, \quad x(0) = 0, \quad x(1) = 1. \tag{4.1}$$

The following are new results presented in this Chapter. Theorems 4.16 and 4.19, previously submitted in [30], discuss a multidomain alternating classical Schwarz iteration. Theorems 4.22 – 4.24 consider alternating classical Schwarz iterations with groupings of subdomains to allow parallel computation. Theorems 4.22–4.23 have been submitted to [30], and Theorem 4.24 is previously unpublished.

Theorems 4.26 – 4.28 cover optimal Schwarz iterations for two subdomains, Theorem 4.29 presents a parallel optimal Schwarz iteration for three non-overlapping subdomains, and Theorem 4.31 gives an optimal result for four or more non-overlapping subdomains. Of the five, Theorems 4.27 and 4.28 have previously been submitted in [30].

Finally, Theorem 4.34, previously submitted in [30], discusses convergence of an alternating optimized Schwarz iteration on two non-overlapping subdomains.

Before presenting convergence results for various alternating and parallel methods for (4.1), we provide some preliminary results which will be used throughout the chapter.

## 4.1 Preliminaries

The DD methods in the following sections require the solution of (4.1) with a specified function $M(x)$, subject to multiple types of boundary conditions. We begin by noting the existence and uniqueness of a solution to this differential equation subject to Dirichlet boundary conditions. These results originally appear in [22].

**Lemma 4.1.** *Consider the following BVP on an arbitrary subdomain* $(\alpha, \beta) \subset \Omega = (0, 1)$,

$$\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) = 0, \quad x(\alpha) = \gamma_\alpha, \quad x(\beta) = \gamma_\beta. \tag{4.2}$$

*If* $M$ *is differentiable and bounded away from zero and infinity, i.e. there exists* $\check{m}$ *and* $\hat{m}$ *such that* $0 < \check{m} \leq M(x) \leq \hat{m} < \infty$ *for all* $x$*, then this BVP has a unique solution given implicitly by*

$$\int_{\gamma_\alpha}^{x(\xi)} M(\tilde{x})d\tilde{x} = \frac{\xi - \alpha}{\beta - \alpha}\int_{\gamma_\alpha}^{\gamma_\beta} M(\tilde{x})d\tilde{x}, \quad \xi \in (\alpha, \beta). \tag{4.3}$$

*Proof.* The differential equation and boundary condition at $\xi = \alpha$ is satisfied by

$$\int_{\gamma_\alpha}^{x(\xi)} M(\tilde{x})\, d\tilde{x} = \mathcal{C}(\xi - \alpha),$$

where we choose $\mathcal{C}$ to satisfy the Dirichlet condition at $\xi = \beta$. From direct calculation we obtain

$$\mathcal{C} = \frac{1}{\beta - \alpha} \int_{\gamma_\alpha}^{\gamma_\beta} M(\tilde{x})\, d\tilde{x},$$

arriving at the implicit representation (4.3).

To establish the existence and uniqueness of $x(\xi)$ which satisfies (4.2), we note that $x(\xi)$ is the solution $\theta$ of

$$G(\theta) = \frac{\xi - \alpha}{\beta - \alpha} \int_{\gamma_\alpha}^{\gamma_\beta} M(\tilde{x})\, d\tilde{x}, \tag{4.4}$$

where $G(\theta) \equiv \int_{\gamma_\alpha}^{\theta} M(\tilde{x})\, d\tilde{x}$. Under the assumptions of Lemma 4.1, $G$ is continuous and uniformly monotonic, that is,

$$\frac{dG}{d\theta} = M(\theta) \geq \check{m} > 0.$$

Hence, by the inverse function theorem, there is a unique, continuously differentiable solution to (4.4) and (4.2). □

A simple consequence of Lemma 4.1 which will be used several times is stated in the following Corollary.

**Corollary 4.2.** *For any $\tilde{\xi} \in (0,1)$, the solution $x(\tilde{\xi})$ which solves (4.2) satisfies the equation*

$$\int_0^{x(\tilde{\xi})} M(\tilde{x})\, d\tilde{x} = \tilde{\xi} \int_0^1 M(\tilde{x})\, d\tilde{x}.$$

Similarly, the following expressions also easily follow from Lemma 4.1.

**Corollary 4.3.** *Suppose the domain $[0,1]$ is decomposed into subdomains $[0,\beta]$ and $[\alpha,1]$, with $\alpha \leq \beta$. Then the following hold:*

*(i) The function $x(\xi)$ solving*

$$\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) = 0, \quad \xi \in [0,\beta],$$

*with $x(0) = 0$ and $x(\beta)$ a known value, satisfies*

$$\int_0^{x(\xi)} M(\tilde{x})\, d\tilde{x} = \frac{\xi}{\xi_u} \int_0^{x(\xi_u)} M(\tilde{x})\, d\tilde{x}, \quad \xi \in [0,\xi_u], \tag{4.5}$$

*where $\xi_u \leq \beta$.*

*(ii) The function $x(\xi)$ solving*

$$\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) = 0, \quad \xi \in [\alpha,1],$$

*with $x(1) = 1$ and $x(\alpha)$ a known value, satisfies*

$$\int_{x(\xi)}^1 M(\tilde{x})\, d\tilde{x} = \frac{1-\xi}{1-\xi_l} \int_{x(\xi_l)}^1 M(\tilde{x})\, d\tilde{x}, \quad \xi \in [\xi_l,1], \tag{4.6}$$

*where $\xi_l \geq \alpha$.*

The analysis of optimized Schwarz methods in Section 4.4 will require the solution of boundary value problems of the form

$$\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) = 0, \quad x(0) = 0, \quad M(x)x_\xi + px\Big|_\beta = \gamma_\beta, \tag{4.7}$$

and

$$\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) = 0, \quad M(x)x_\xi - px\Big|_\beta = \gamma_\beta, \quad x(1) = 1, \tag{4.8}$$

where $p$ and $\gamma_\beta$ are constants and $\beta \in (0,1)$ is fixed. Note the change of sign in the boundary condition at $\xi = \beta$ from (4.7) to (4.8). The existence of unique solutions to these BVPs, as well as their implicit solutions, are provided in the following two Lemmas.

**Lemma 4.4.** *Under the assumptions of Lemma 4.1, the BVP* (4.7) *has a unique solution for all* $p > 0$ *given implicitly by*

$$\int_0^{x(\xi)} M(\tilde{x})\,d\tilde{x} = (\gamma_\beta - px(\beta))\xi, \quad \xi \in (0,\beta). \tag{4.9}$$

*Proof.* The differential equation and boundary condition at $\xi = 0$ is satisfied by

$$\int_0^{x(\xi)} M(\tilde{x})\,d\tilde{x} = C\xi,$$

where $C$ is chosen to satisfy the Robin condition at $\xi = \beta$. Direct calculation gives $C = \gamma_\beta - px(b)$, from which (4.9) follows.

To establish the existence and uniqueness of $x(\xi)$ satisfying (4.9), we note that if $\xi = \beta$ then the boundary value $x(\beta)$ is the solution $\theta$ of

$$G(\theta) = \beta\gamma_\beta, \tag{4.10}$$

where

$$G(\theta) \equiv \int_0^\theta M(\tilde{x})\, d\tilde{x} + \beta p\theta.$$

Under the assumptions of Lemma 4.1, $G$ is continuous and uniformly monotonic, that is, there exists a constant $G_p > 0$ such that

$$\frac{dG}{d\theta} = M(\theta) + \beta p \geq G_p > 0,$$

hence (4.10) has a unique solution $x(\beta)$. The existence of a unique, continuously differentiable solution $x(\xi)$ for $\xi \in (0, \beta)$ follows from considering (4.9) with the now specified $x(\beta)$. Noting that

$$\tilde{G}(\theta) = \int_0^\theta M(\tilde{x})\, d\tilde{x},$$

a continuous and uniformly monotonic function, it follows that it has a continuously differentiable inverse. □

**Lemma 4.5.** *Under the assumptions of Lemma 4.1, the BVP* (4.8) *has a unique solution for all* $p > 0$, *given implicitly as*

$$\int_{x(\xi)}^1 M(\tilde{x})\, d\tilde{x} = (\gamma_\beta + px(\beta))(1 - \xi), \quad \xi \in (\beta, 1). \tag{4.11}$$

*Proof.* Equation (4.11) follows from direct calculation. To establish existence and uniqueness of a function $x(\xi)$ satisfying (4.11), we evaluate this equation for $\xi = \beta$ to obtain

$$\int_{x(\beta)}^{1} M(\tilde{x})\, d\tilde{x} = (1-\beta)(\gamma_\beta + px(\beta)),$$

or, after rearranging,

$$\int_{x(\beta)}^{1} M(\tilde{x})\, d\tilde{x} - (1-\beta)px(\beta) = (1-\beta)\gamma_\beta.$$

If we define

$$G(\theta) = \int_{\theta}^{1} M(\tilde{x})\, d\tilde{x} - (1-\beta)p\theta,$$

under the assumptions of Lemma 4.1, $G$ is a continuous and uniformly monotone decreasing, hence is invertible. We conclude there is a unique $x(\beta)$ given by $G^{-1}((1-\beta)\gamma_\beta)$. Having determined $x(\beta)$, the unique solution $x(\xi)$ for $\xi \in (\beta, 1)$ is given by

$$\tilde{G}^{-1}((\gamma_\beta + px(\beta))(1-\xi)), \quad \text{where} \quad \tilde{G}(\theta) = \int_{\theta}^{1} M(\tilde{x})\, d\tilde{x},$$

is clearly a continuous and uniformly monotonic decreasing function under the stated assumptions. □

## 4.2 Classical Schwarz Methods

In this section, we present methods to solve (4.1) via classical Schwarz iterations. We begin with the two subdomain parallel Schwarz method due to Gander and Haynes [22] and the alternating Schwarz method we previously described in [23]. We then discuss classical Schwarz methods on an arbitrary number of subdomains, again in both parallel and alternating forms. It will be seen that there are alternating iterations which can be implemented in parallel to take advantage of parallel computation without sacrificing the improved convergence of alternating methods.

### 4.2.1 Two Subdomain Methods

We decompose the domain $\Omega_c = [0,1]$ into two overlapping subdomains $\Omega_1 = [0,\beta]$ and $\Omega_2 = [\alpha,1]$ with $\alpha < \beta$, and first consider the parallel iteration: for $n = 1,2,\ldots$

$$\begin{aligned} (M(x_1^n)x_{1,\xi}^n)_\xi = 0, \ \ \xi \in \Omega_1, \qquad & (M(x_2^n)x_{2,\xi}^n)_\xi = 0, \ \ \xi \in \Omega_2, \\ x_1^n(0) = 0, \qquad & x_2^n(\alpha) = x_1^{n-1}(\alpha), \\ x_1^n(\beta) = x_2^{n-1}(\beta), \qquad & x_2^n(1) = 1. \end{aligned} \tag{4.12}$$

We begin by constructing implicit solutions on the subdomains using Lemma 4.1.

**Lemma 4.6.** *Under the assumptions of Lemma 4.1, the subdomain solutions on* $\Omega_1$

*and* $\Omega_2$ *of* (4.12) *are given implicitly by*

$$\int_0^{x_1^n(\xi)} M(\tilde{x})\, d\tilde{x} = \frac{\xi}{\beta} \int_0^{x_2^{n-1}(\beta)} M(\tilde{x})\, d\tilde{x}, \tag{4.13}$$

*and*

$$\int_{x_2^n(\xi)}^1 M(\tilde{x})\, d\tilde{x} = \frac{1-\xi}{1-\alpha} \int_{x_1^{n-1}(\alpha)}^1 M(\tilde{x})\, d\tilde{x}. \tag{4.14}$$

*Proof.* Simply compare the subdomain problems in (4.12) with (4.2) and use the implicit representation of the solution in (4.3). □

Using these representations of the subdomain solutions we can prove a convergence estimate for the parallel Schwarz iteration (4.12). This result first appeared in [22]. We will use the $L^\infty$ norm, defined for any function $f : (a,b) \to \mathbb{R}$ by $||f||_\infty := \sup_{x\in(a,b)} |f(x)|$.

**Theorem 4.7.** *Under the assumptions of Lemma 4.1, the parallel classical Schwarz iteration* (4.12) *converges for any initial values* $x_1^0(\alpha)$ *and* $x_2^0(\beta)$. *Furthermore, we have the convergence estimates*

$$||x - x_1^{2n+1}||_\infty \le \rho^n \frac{\hat{m}}{\check{m}} |x(\beta) - x_2^0(\beta)|, \quad ||x - x_2^{2n+1}||_\infty \le \rho^n \frac{\hat{m}}{\check{m}} |x(\alpha) - x_1^0(\alpha)|, \tag{4.15}$$

*where*

$$\rho := \frac{\alpha}{\beta} \frac{1-\beta}{1-\alpha} < 1.$$

*Proof.* Using Lemma 4.6 and defining $C := \int_0^1 M(\tilde{x})\,d\tilde{x}$, the sequence $x_1^n(\alpha)$ satisfies

$$\begin{aligned}\int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x} &= \frac{\alpha}{\beta}\int_0^{x_2^{n-1}(\beta)} M(\tilde{x})\,d\tilde{x} = \frac{\alpha}{\beta}\left(C - \int_{x_2^{n-1}(\beta)}^1 M(\tilde{x})\,d\tilde{x}\right)\\ &= \frac{\alpha}{\beta}\left(C - \frac{1-\beta}{1-\alpha}\int_{x_1^{n-2}(\alpha)}^1 M(\tilde{x})\,d\tilde{x}\right)\\ &= \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha}\int_0^{x_1^{n-2}(\alpha)} M(\tilde{x})\,d\tilde{x} + \frac{\alpha}{\beta}\frac{\beta-\alpha}{1-\alpha}C,\end{aligned} \tag{4.16}$$

where the second and fourth equalities above follow from

$$\int_0^x M(\tilde{x})\,d\tilde{x} = C - \int_x^1 M(\tilde{x})\,d\tilde{x},$$

and the third follows from (4.14) evaluated at $\xi = \beta$ with $n$ replaced by $n-1$. Defining

$$K_1^n = \int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x},$$

the relation (4.16) is equivalent to the *linear* fixed point iteration

$$K_1^n = \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha}K_1^{n-2} + \frac{\alpha}{\beta}\frac{\beta-\alpha}{1-\alpha}C. \tag{4.17}$$

The contraction factor of this iteration,

$$\rho := \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha},$$

is strictly less than one, hence the iteration converges to a limit $K_1^*$ satisfying

$$K_1^* = \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha}K_1^* + \frac{\alpha}{\beta}\frac{\beta-\alpha}{1-\alpha}C, \quad \text{or} \quad K_1^* = \alpha C. \tag{4.18}$$

Similarly, we can obtain a fixed point iteration for the second subdomain,

$$K_2^n = \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha}K_2^{n-2} + \frac{\beta-\alpha}{1-\alpha}C, \tag{4.19}$$

where $K_2^n$ is defined by

$$K_2^n = \int_0^{x_2^n(\beta)} M(\tilde{x})\,d\tilde{x}.$$

Possessing the same contraction factor $\rho$, this iteration also converges, the limit $K_2^*$ satisfying

$$K_2^* = \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha}K_2^* + \frac{\beta-\alpha}{1-\alpha}C, \quad \text{or} \quad K_2^* = \beta C. \tag{4.20}$$

We have thus shown

$$\lim_{n\to\infty}\int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x} = \alpha\int_0^1 M(\tilde{x})\,d\tilde{x} \quad \text{and} \quad \lim_{n\to\infty}\int_0^{x_2^n(\beta)} M(\tilde{x})\,d\tilde{x} = \beta\int_0^1 M(\tilde{x})\,d\tilde{x}.$$

From Corollary 4.2, it follows that

$$\lim_{n\to\infty}\int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x} = \int_0^{x(\alpha)} M(\tilde{x})\,d\tilde{x} \quad \text{and} \quad \lim_{n\to\infty}\int_0^{x_2^n(\beta)} M(\tilde{x})\,d\tilde{x} = \int_0^{x(\beta)} M(\tilde{x})\,d\tilde{x}.$$

We now establish the $L^\infty$ norm convergence estimates. By subtracting (4.13) and (4.14) from the corresponding equivalent expressions for $x(\xi)$, we obtain

$$\int_{x_1^{2n+1}(\xi)}^{x(\xi)} M(\tilde{x})\,d\tilde{x} = \frac{\xi}{\beta}\int_{x_2^{2n}(\beta)}^{x(\beta)} M(\tilde{x})\,d\tilde{x}. \tag{4.21}$$

and

$$\int_{x_2^{2n+1}(\xi)}^{x(\xi)} M(\tilde{x})\,d\tilde{x} = \frac{1-\xi}{1-\alpha}\int_{x_1^{2n}(\alpha)}^{x(\alpha)} M(\tilde{x})\,d\tilde{x}. \tag{4.22}$$

Subtracting (4.17) from (4.18) and likewise subtracting (4.19) from (4.20), then proceeding by induction, we obtain

$$\int_{x_1^{2n}(\alpha)}^{x(\alpha)} M(\tilde{x})\,d\tilde{x} = \rho^n \int_{x_1^0(\alpha)}^{x(\alpha)} M(\tilde{x})\,d\tilde{x}, \tag{4.23}$$

and

$$\int_{x_2^{2n}(\beta)}^{x(\beta)} M(\tilde{x})\,d\tilde{x} = \rho^n \int_{x_2^0(\beta)}^{x(\beta)} M(\tilde{x})\,d\tilde{x}. \tag{4.24}$$

Combining (4.24) with (4.21) and (4.23) with (4.22), we have

$$\int_{x_1^{2n+1}(\xi)}^{x(\xi)} M(\tilde{x})\,d\tilde{x} = \frac{\xi}{\beta}\rho^n \int_{x_2^0(\beta)}^{x(\beta)} M(\tilde{x})\,d\tilde{x}, \tag{4.25}$$

and

$$\int_{x_2^{2n+1}(\xi)}^{x(\xi)} M(\tilde{x})\,d\tilde{x} = \frac{1-\xi}{1-\alpha}\rho^n \int_{x_1^0(\alpha)}^{x(\alpha)} M(\tilde{x})\,d\tilde{x}. \tag{4.26}$$

Finally, by the bounds on $M$ we have, for any $a, b \in \mathbb{R}$,

$$\check{m}|a-b| \le \left|\int_a^b M(\tilde{x})\,d\tilde{x}\right| \le \hat{m}|a-b|. \tag{4.27}$$

Taking absolute values in (4.25) and (4.26), then applying (4.27), we obtain the estimates

$$|x(\xi) - x_1^{2n+1}(\xi)| \le \frac{\xi}{\beta}\rho^n \frac{\hat{m}}{\check{m}}|x(\beta) - x_2^0(\beta)|, \quad \xi \in [0,\beta],$$

and

$$|x(\xi) - x_2^{2n+1}(\xi)| \le \frac{1-\xi}{1-\alpha}\rho^n \frac{\hat{m}}{\check{m}}|x(\alpha) - x_1^0(\alpha)|, \quad \xi \in [\alpha,1].$$

We obtain the estimates in (4.15) by taking the supremum of each inequality. □

The need for overlapping subdomains for the convergence of classical Schwarz methods can be seen from the statement of Theorem 4.7: the contraction factor $\rho$ is less than one only if $\alpha < \beta$.

In a similar fashion we describe the alternating Schwarz iteration: for $n = 1, 2, \ldots$

$$\begin{aligned} &(M(x_1^n)x_{1,\xi}^n)_\xi = 0, \ \ \xi \in \Omega_1, && (M(x_2^n)x_{2,\xi}^n)_\xi = 0, \ \ \xi \in \Omega_2, \\ &x_1^n(0) = 0, && x_2^n(\alpha) = x_1^n(\alpha), \\ &x_1^n(\beta) = x_2^{n-1}(\beta), && x_2^n(1) = 1. \end{aligned} \tag{4.28}$$

For each iteration we first solve the subdomain problem over $\Omega_1$, then use the resulting solution to provide updated boundary conditions for the subdomain problem over $\Omega_2$. This can accelerate the convergence of the iteration, with the drawback being that subdomain problems must be solved sequentially.

Before establishing convergence of (4.28), we use direct integration to represent the subdomain solutions implicitly, recording them in Lemma 4.8.

**Lemma 4.8.** *The subdomain solutions on $\Omega_1$ and $\Omega_2$ of (4.28) are given implicitly as*

$$\int_0^{x_1^n(\xi)} M(\tilde{x})\, d\tilde{x} = \frac{\xi}{\beta} \int_0^{x_2^{n-1}(\beta)} M(\tilde{x})\, d\tilde{x}, \tag{4.29}$$

*and*

$$\int_0^{x_2^n(\xi)} M(\tilde{x})\, d\tilde{x} = \frac{1-\xi}{1-\alpha} \int_0^{x_1^n(\alpha)} M(\tilde{x})\, d\tilde{x} + \frac{\xi-\alpha}{1-\alpha} \int_0^1 M(\tilde{x})\, d\tilde{x}. \tag{4.30}$$

We now prove Theorem 4.9, previously presented in [23].

**Theorem 4.9.** *Under the assumptions of Lemma 4.1, the alternating Schwarz iteration (4.28) converges for any initial guess $x_2^0(\beta)$ and we have the error estimates*

$$||x - x_1^{n+1}||_\infty \leq \rho^n \frac{\hat{m}}{\check{m}} |x(\beta) - x_2^0(\beta)|, \quad ||x - x_2^{n+1}||_\infty \leq \rho^n \frac{\hat{m}}{\check{m}} |x(\alpha) - x_1^0(\alpha)|, \quad (4.31)$$

*with contraction factor* $\rho := \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha} < 1$.

*Proof.* Evaluating (4.29) at $\xi = \alpha$ and using the expression for $x_2^{n-1}(\beta)$ from (4.30), we have

$$\int_0^{x_1^n(\alpha)} M\, d\tilde{x} = \frac{\alpha}{\beta}\left\{\frac{\beta-1}{\alpha-1}\int_0^{x_1^{n-1}(\alpha)} M\, d\tilde{x} + \frac{\beta-\alpha}{1-\alpha}\int_0^1 M\, d\tilde{x}\right\}.$$

Defining the quantities

$$K_1^n = \int_0^{x_1^n(\alpha)} M(\tilde{x})\, d\tilde{x} \qquad \text{and} \qquad C = \int_0^1 M(\tilde{x})\, d\tilde{x},$$

we obtain the iteration

$$K_1^n = \frac{\alpha}{\beta}\frac{\beta-1}{\alpha-1}K_1^{n-1} + \frac{\alpha}{\beta}\frac{\beta-\alpha}{1-\alpha}C. \quad (4.32)$$

This iteration converges with rate

$$\rho := \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha} < 1,$$

and has limit

$$K_1^* = \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha}K_1^* + \frac{\alpha}{\beta}\frac{\beta-\alpha}{1-\alpha}C, \quad \text{or} \quad K_1^* = \alpha C. \quad (4.33)$$

As the single domain solution satisfies

$$\int_0^{x(\alpha)} M(\tilde{x})\, d\tilde{x} = \alpha C,$$

and $M(x) \geq \check{m} > 0$, we have convergence at the interface to the correct limit.

Taking the difference of (4.32) and (4.33), we have

$$\int_{x_1^n(\alpha)}^{x(\alpha)} M(\tilde{x})\, d\tilde{x} = \rho^n \int_{x_1^0(\alpha)}^{x(\alpha)} M(\tilde{x})\, d\tilde{x}. \tag{4.34}$$

Subtracting (4.30) from the equivalent expression for the single domain solution and using (4.34), we obtain

$$\int_{x_2^{n+1}(\xi)}^{x(\xi)} M(\tilde{x})\, d\tilde{x} = \frac{1-\xi}{1-\alpha} \int_{x_1^n(\alpha)}^{x(\alpha)} M(\tilde{x})\, d\tilde{x} = \frac{1-\xi}{1-\alpha} \rho^n \int_{x_1^0(\alpha)}^{x(\alpha)} M(\tilde{x})\, d\tilde{x}.$$

Taking absolute values and using the bounds on $M$, we obtain, for all $\xi \in [\alpha, 1]$,

$$|x(\xi) - x_2^{n+1}(\xi)| \leq \frac{1-\xi}{1-\alpha} \rho^n \frac{\hat{m}}{\check{m}} |x(\alpha) - x_1^0(\alpha)|.$$

Taking the supremum gives the second estimate in (4.31). The subdomain one estimate follows similarly. □

### 4.2.2 A Parallel Multidomain Method

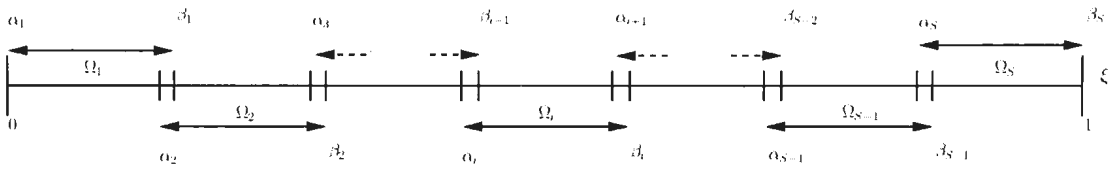


Figure 4.1: Decomposing the unit interval into $S$ subdomains.

In practice, we would like to take advantage of multicore, multiprocessor scientific computing environments for grid generation. A typical multidomain decomposition of the interval $[0, 1]$ into $S$ subdomains is illustrated in Figure 4.1. Towards this goal, we now review the extension of the classical Schwarz methods for (4.1) to $S > 2$ subdomains, originally presented by Gander and Haynes in [22]. The subdomain solution, $x_i(\xi)$, on the $i^{\text{th}}$ subdomain, $\Omega_i = (\alpha_i, \beta_i)$, for $i = 1, 2, \ldots, S$, is found by solving

$$(M(x_i)x_{i,\xi})_\xi = 0, \quad x_i(\alpha_i) = x_{i-1}(\alpha_i), \quad x_i(\beta_i) = x_{i+1}(\beta_i), \tag{4.35}$$

where $\alpha_1 = 0$, $x_1(\alpha_1) = 0$, $\beta_S = 1$ and $x_S(\beta_S) = 1$. In addition to requiring that adjacent subdomains overlap ($\alpha_{i+1} < \beta_i$, for $i = 1, \ldots, S-1$), we assume that $\beta_i \leq \alpha_{i+2}$ for $i = 1, \ldots, S-2$, so that there is no overlap between non-adjacent subdomains. We obtain the solution on the whole domain by composing the subdomain solutions $x_i(\xi)$. We begin by presenting the classical parallel Schwarz iteration: for $n = 1, 2, \ldots$

$$(M(x_i^n)x_{i,\xi}^n)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n-1}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n-1}(\beta_i), \tag{4.36}$$

for $i = 1, \ldots, S$, where we have defined for convenience $x_0^n(\alpha_1) \equiv 0$ and $x_{S+1}^n(\beta_S) \equiv 1$. The analysis which follows is from [22]. We define the error on the $i^{\text{th}}$ subdomain at iteration $n$, as

$$e_i^n(\xi) = \int_{x_i(\xi)}^{x_i^n(\xi)} M(\tilde{x})\, d\tilde{x}, \quad \text{for} \quad i = 1, \ldots, S. \tag{4.37}$$

We demonstrate convergence by showing this measure contracts to zero for each subdomain. As $M$ is bounded away from zero we know $\lim_{n\to\infty} e_i^n(\xi) = 0$ implies $\lim_{n\to\infty} |x_i(\xi) - x_i^n(\xi)| = 0$.

The error on each subdomain is given explicitly in the following Lemma. For convenience we denote $e_0^n(\alpha_1) \equiv 0$ and $e_{S+1}^n(\beta_S) \equiv 0$.

**Lemma 4.10.** *The error on each subdomain satisfies*

$$e_i^{n+2}(\xi) = \frac{1}{\beta_i - \alpha_i}\left[(\xi - \alpha_i)e_{i+1}^{n+1}(\beta_i) + (\beta_i - \xi)e_{i-1}^{n+1}(\alpha_i)\right], \tag{4.38}$$

*for* $\xi \in [\alpha_i, \beta_i]$, $i = 1, \ldots, S$.

*Proof.* Computing the difference of (4.35) and (4.36) and differentiating (4.37) twice, we see the error $e_i^{n+2}(\xi)$ satisfies the linear BVP

$$\frac{d^2 e_i^{n+2}}{d\xi^2} = 0, \quad e_i^{n+2}(\alpha_i) = e_{i-1}^{n+1}(\alpha_i), \quad e_i^{n+2}(\beta_i) = e_{i+1}^{n+1}(\beta_i),$$

for $i = 1, \ldots, S$. Integrating and applying the boundary conditions gives the stated result. □

We now use (4.38) to relate the error on subdomain $i$ at iteration $n+2$ to the error on subdomain $i$ and its neighbors at iteration $n$. For simplicity, we introduce the quantities

$$r_i = \frac{\beta_{i-1} - \alpha_i}{\beta_i - \alpha_i}, \quad p_i = \frac{\beta_i - \beta_{i-1}}{\beta_i - \alpha_i}, \quad q_i = \frac{\alpha_{i+1} - \alpha_i}{\beta_i - \alpha_i}, \quad s_i = \frac{\beta_i - \alpha_{i+1}}{\beta_i - \alpha_i}. \tag{4.39}$$

**Lemma 4.11.** *The error at the interface $\xi = \beta_{i-1},\ i = 2, \ldots, S$ satisfies*

$$\begin{aligned} |e_i^{n+2}(\beta_{i-1})| \leq & r_i r_{i+1} |e_{i+2}^n(\beta_{i+1})| + r_i p_{i+1} |e_i^n(\alpha_{i+1})| \\ & + p_i q_{i-1} |e_i^n(\beta_{i-1})| + p_i s_{i-1} |e_{i-2}^n(\alpha_{i-1})|, \end{aligned} \tag{4.40}$$

*while at $\xi = \alpha_{i+1},\ i = 1, \ldots, S-1$ we have*

$$\begin{aligned} |e_i^{n+2}(\alpha_{i+1})| \leq & q_i r_{i+1} |e_{i+2}^n(\beta_{i+1})| + q_i p_{i+1} |e_i^n(\alpha_{i+1})| \\ & + s_i q_{i-1} |e_i^n(\beta_{i-1})| + s_i s_{i-1} |e_{i-2}^n(\alpha_{i-1})|. \end{aligned} \tag{4.41}$$

*Proof.* Inequality (4.40) is obtained by evaluating (4.38) at $\beta_{i-1}$ and then using (4.38) to write $e_{i+1}^{n+1}(\beta_i)$ in terms of $e_{i+2}^n$ and $e_i^n$ and $e_{i-1}^{n+1}(\alpha_i)$ in terms of $e_i^n$ and $e_{i-2}^n$. Taking absolute values, applying the extended triangle inequality and noting each of the quantities in (4.39) is non-negative gives the result. Inequality (4.41) is obtained similarly. □

If $S$ is even, we write the relations (4.40–4.41) as the matrix inequalities

$$\boldsymbol{e}^{n+2} \leq M_{\boldsymbol{e}} \boldsymbol{e}^n \quad \text{and} \quad \hat{\boldsymbol{e}}^{n+2} \leq M_{\hat{\boldsymbol{e}}} \hat{\boldsymbol{e}}^n,$$

where

$$\boldsymbol{e}^n = (|e_1^n(\alpha_2)|, |e_3^n(\beta_2)|, |e_3^n(\alpha_4)|, \ldots, |e_{S-1}^n(\beta_{S-2})|, |e_{S-1}^n(\alpha_S)|)^T,$$

$$\hat{\boldsymbol{e}}^n = (|e_2(\beta_1)|, |e_2^n(\alpha_3)|, |e_4^n(\beta_3)|, \ldots, |e_{S-2}^n(\alpha_{S-1})|, |e_S^n(\beta_{S-1})|)^T,$$

with $(S-1)\times(S-1)$ matrices $M_e$ and $M_{\hat{e}}$ given by

$$M_e = \begin{pmatrix} p_2q_1 & r_2q_1 & & & & & & \\ p_3s_2 & p_3q_2 & p_4r_3 & r_4r_3 & & & & \\ s_3s_2 & s_3q_2 & p_4q_3 & r_4q_3 & & & & \\ & & p_5s_4 & p_5q_4 & p_6r_5 & r_6r_5 & & \\ & & s_5s_4 & s_5q_4 & p_6q_5 & r_6q_5 & & \\ & & & & \ddots & \ddots & & \\ & & & & & p_{S-1}s_{S-2} & p_{S-1}q_{S-2} & p_Sr_{S-1} \\ & & & & & s_{S-1}s_{S-2} & s_{S-1}q_{S-2} & p_Sq_{S-1} \end{pmatrix},$$

and

$$M_{\hat{e}} = \begin{pmatrix} p_2q_1 & q_3r_2 & r_3r_2 & & & & & \\ s_2q_1 & p_3q_2 & r_3q_2 & & & & & \\ & p_4s_3 & p_4q_3 & p_5r_4 & r_5r_4 & & & \\ & s_4s_3 & s_4q_3 & p_5q_4 & r_5q_4 & & & \\ & & & \ddots & \ddots & & & \\ & & & p_{S-2}s_{S-3} & p_{S-2}q_{S-3} & p_{S-1}r_{S-2} & r_{S-1}r_{S-2} & \\ & & & s_{S-2}s_{S-3} & s_{S-2}q_{S-3} & p_{S-1}q_{S-2} & r_{S-1}q_{S-2} & \\ & & & & & p_Ss_{S-1} & p_Sq_{S-1} & \end{pmatrix}.$$

If instead $S$ is odd, we obtain the matrix inequalities

$$\boldsymbol{f}^{n+2} \le M_{\boldsymbol{f}}\boldsymbol{f}^n \quad \text{and} \quad \hat{\boldsymbol{f}}^{n+2} \le M_{\hat{\boldsymbol{f}}}\hat{\boldsymbol{f}}^n,$$

where

$$\boldsymbol{f}^n = (|e_1^n(\alpha_2)|, |e_3^n(\beta_2)|, |e_3^n(\alpha_4)|, \ldots, |e_S^n(\beta_{S-1})|)^T,$$

$$\hat{\boldsymbol{f}}^n = (|e_2(\beta_1)|, |e_2^n(\alpha_3)|, \ldots, |e_{S-1}^n(\beta_{S-2})|, |e_{S-1}^n(\alpha_S)|)^T,$$

and the $(S-1)\times(S-1)$ matrices $M_{\boldsymbol{f}}$ and $M_{\hat{\boldsymbol{f}}}$ are given by

$$M_{\boldsymbol{f}} = \begin{pmatrix} p_2q_1 & r_2q_1 & & & & & & \\ p_3s_2 & p_3q_2 & p_4r_3 & r_4r_3 & & & & \\ s_3s_2 & s_3q_2 & p_4q_3 & r_4q_3 & & & & \\ & & & \ddots & \ddots & & & \\ & & & p_{S-2}s_{S-3} & p_{S-2}q_{S-3} & p_{S-1}r_{S-2} & r_{S-1}r_{S-2} \\ & & & s_{S-2}s_{S-3} & s_{S-2}q_{S-3} & p_{S-1}q_{S-2} & r_{S-1}q_{S-2} \\ & & & & & p_Ss_{S-1} & p_Sq_{S-1} \end{pmatrix},$$

and

$$M_{\hat{\boldsymbol{f}}} = \begin{pmatrix} p_2q_1 & q_3r_2 & r_3r_2 & & & & \\ s_2q_1 & p_3q_2 & r_3q_2 & & & & \\ & p_4s_3 & p_4q_3 & p_5r_4 & r_5r_4 & & \\ & s_4s_3 & s_4q_3 & p_5q_4 & r_5q_4 & & \\ & & & & \ddots & \ddots & \\ & & & & p_{S-1}s_{S-2} & p_{S-1}q_{S-2} & p_Sr_{S-1} \\ & & & & s_{S-1}s_{S-2} & s_{S-1}q_{S-2} & p_Sq_{S-1} \end{pmatrix}.$$

From (4.39), we see that for each $i$, we have $p_i + r_i = q_i + s_i = 1$. It follows that $\|M_{\boldsymbol{e}}\|_\infty = \|M_{\hat{\boldsymbol{e}}}\|_\infty = \|M_{\boldsymbol{f}}\|_\infty = \|M_{\hat{\boldsymbol{f}}}\|_\infty = 1$, so convergence cannot be argued directly using the infinity norm. In [25] it was shown there exists some $\tilde{n} > 0$ for which $\|M_{\boldsymbol{e}}^{\tilde{n}}\|_\infty < 1$ and $\|M_{\hat{\boldsymbol{e}}}^{\tilde{n}}\|_\infty < 1$ from which convergence follows. An alternative method to show convergence was demonstrated in [22]. For a real $m \times m$ matrix $A$ and $a \in \mathbb{R}$, define $\Psi_a(A)$ to be the matrix obtained by deleting all rows, and corresponding columns, for which $\sum_{j=1}^{m} |a_{ij}| < a$. The following result comes from [58].

**Lemma 4.12.** *Suppose a $m \times m$ real matrix $A$ is element-wise non-negative, then $\rho(A) < \|A\|_\infty$ if and only if $\Psi^m_{\|A\|_\infty}(A) = 0$, where $0$ is the $m \times m$ zero matrix.*

Using this Lemma, we prove the following theorem.

**Theorem 4.13.** *Under the assumptions of Lemma 4.1 and the stated restrictions on the decomposition of $\Omega_c$, the multidomain parallel Schwarz iteration* (4.36) *converges globally on an arbitrary number of subdomains.*

*Proof.* The assumptions on the choice of the subdomains ensure the quantities (4.39) are non-negative, hence $M_{\boldsymbol{e}}$ is non-negative (a matrix being non-negative when all elements are non-negative). The first, second last and last rows of $M_{\boldsymbol{e}}$ each have sums less than one, hence $\Psi_1(M_{\boldsymbol{e}})$ zeroes rows and columns one, $S-2$ and $S-1$ of $M_{\boldsymbol{e}}$. We then see that rows two, three, $S-4$ and $S-3$ of $\Psi_1(M_{\boldsymbol{e}})$ have sums less than one, hence $\Psi_1^2(M_{\boldsymbol{e}})$ would zero rows and columns two, three, $S-4$ and $S-3$. It is clear that $\Psi_1^{\tilde{n}}(M_{\boldsymbol{e}}) = 0$ for some $\tilde{n} < S-1$. By Lemma 4.12, $\rho(M_{\boldsymbol{e}}) < 1$. We can similarly show that $\rho(M_{\hat{\boldsymbol{e}}}) < 1$, $\rho(M_{\boldsymbol{f}}) < 1$, and $\rho(M_{\hat{\boldsymbol{f}}}) < 1$. □

If we make a further assumption that the overlap between each pair of subdomains is the same size, we can give an explicit bound on the contraction rate. This assumption forces $r_i = s_i = r$ and $p_i = q_i = p$, leading to the following explicit error estimate.

**Theorem 4.14.** *The Schwarz iteration* (4.36) *on $S$ subdomains with a common over-*

*lap ratio* $r \in (0, 0.5]$ *converges in the infinity norm with iterates satisfying*

$$\max_{1\le 2i\le S} \|x_{2i}^{2n+1}(\xi) - x(\xi)\|_\infty \le \left(1 - 4r(1-r)\sin^2\frac{\pi}{2(S+1)}\right)^n \frac{1}{\check{m}}\|e^0\|_2,$$
$$\max_{1\le 2i+1\le S} \|x_{2i+1}^{2n+1}(\xi) - x(\xi)\|_\infty \le \left(1 - 4r(1-r)\sin^2\frac{\pi}{2(S+1)}\right)^n \frac{1}{\check{m}}\|\hat{e}^0\|_2.$$

*Proof.* From the boundedness of $M$ and (4.27), we have for each $i$ and all $\xi$ that

$$|x_{2i}^{2n+1}(\xi) - x(\xi)| \le \frac{1}{\check{m}}|e_{2i}^{2n+1}(\xi)|.$$

As $|e_{2i}^{2n+1}|$ is linear it is bounded by the maximum of its boundary values. This leads to the sequence of inequalities

$$\begin{aligned}|x_{2i}^{2n+1}(\xi) - x(\xi)| &\le \frac{1}{\check{m}}|e_{2i}^{2n+1}(\xi)| \le \frac{1}{\check{m}}\max\{|e_{2i+1}^{2n}(\beta_{2i})|, |e_{2i-1}^{2n}(\alpha_{2i})|\}\\ &\le \frac{1}{\check{m}}\|e^{2n}\|_\infty \le \frac{1}{\check{m}}\|e^{2n}\|_2\\ &\le \frac{1}{\check{m}}\left(1 - 4r(1-r)\sin^2\frac{\pi}{2(S+1)}\right)^n \|e^0\|_2.\end{aligned}$$

For odd subdomains we obtain

$$\begin{aligned}|x_{2i+1}^{2n+1}(\xi) - x(\xi)| &\le \frac{1}{\check{m}}|e_{2i+1}^{2n+1}(\xi)| \le \frac{1}{\check{m}}\max\{|e_{2i+2}^{2n}(\beta_{2i+1})|, |e_{2i}^{2n}(\alpha_{2i+1})|\}\\ &\le \frac{1}{\check{m}}\|\hat{e}^{2n}\|_\infty \le \frac{1}{\check{m}}\|\hat{e}^{2n}\}_2\\ &\le \frac{1}{\check{m}}\left(1 - 4r(1-r)\sin^2\frac{\pi}{2(S+1)}\right)^n \|\hat{e}^0\|_2.\end{aligned}$$

In both cases the last inequality follows from a similar result in [25]. □

By examining these bounds, we see that the convergence rate deteriorates as additional subdomains are used. This is not unique to the multidomain iteration for (4.1); similar results can be observed for many multidomain DD methods. A common fix for this problem is the introduction of what is known as a coarse grid correction, for examples of this see [55].

### 4.2.3 An Alternating Multidomain Method

While the forthcoming multidomain alternating Schwarz iteration is straightforward to understand and implement, there are significant differences from both the two subdomain alternating case and the multidomain parallel Schwarz iteration. We previously presented the results in this subsection in [30]. To help illustrate the general case, we begin with the four subdomain case, four subdomains being the minimum number of subdomains required for the general pattern to be apparent.

#### Four Subdomains

We decompose $\Omega = [0,1]$ into subdomains $\Omega_i = [\alpha_i, \beta_i]$ for $i = 1, \ldots, 4$, where $\alpha_{i+1} < \beta_i$ for $i = 1, 2, 3$. Furthermore, we assume that $\beta_i < \alpha_{i+2}$ so that non-adjacent subdomains do not overlap. We denote by $x_i(\xi)$ the solution over $\Omega_i$ which is equal

to the original, single domain solution throughout $\Omega_i$. These solutions will satisfy

$$(M(x_i)x_{i,\xi})_\xi = 0, \quad x_i(\alpha_i) = x_{i-1}(\alpha_i), \quad x_i(\beta_i) = x_{i+1}(\beta_i), \quad \text{for} \quad i = 1, \ldots, 4,$$

where $\alpha_1 = 0$, $\beta_4 = 1$, and we define $x_0(\alpha_1) = 0$ and $x_5(\beta_4) = 1$. To determine the subdomain solutions, we implement the classical alternating Schwarz iteration: for $n = 1, 2, \ldots$

$$\left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^n(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n-1}(\beta_i), \quad \text{for} \quad i = 1, \ldots, 4, \tag{4.42}$$

where we define for convenience $x_0^n(\alpha_1) \equiv 0$ and $x_5^n(\beta_4) \equiv 1$.

The error on the $i^{\text{th}}$ subdomain, at iteration $n$, is defined by

$$e_i^n(\xi) = \int_{x_i(\xi)}^{x_i^n(\xi)} M(\tilde{x})\, d\tilde{x}, \quad \text{for} \quad i = 1, \ldots, 4. \tag{4.43}$$

As in the parallel case, convergence is demonstrated by showing this error measure contracts to zero on all subdomains. By the boundedness of $M$,

$$\lim_{n\to\infty} e_i^n(\xi) = 0 \quad \text{implies} \quad \lim_{n\to\infty} |x_i(\xi) - x_i^n(\xi)| = 0.$$

The error on each subdomain is given in Lemma 4.15, introducing $e_0^n(\alpha_1) \equiv 0$ and $e_5^n(\beta_4) \equiv 0$ for convenience. The proof follows the proof of Lemma 4.10 and is omitted.

**Lemma 4.15.** *The error on each subdomain satisfies*

$$e_i^n(\xi) = \frac{1}{\beta_i - \alpha_i}\left[(\xi - \alpha_i)e_{i+1}^{n-1}(\beta_i) + (\beta_i - \xi)e_{i-1}^n(\alpha_i)\right], \tag{4.44}$$

*for* $\xi \in [\alpha_i, \beta_i]$ *and for* $i = 1, \ldots, 4$.

We introduce the quantities

$$r_i = \frac{\beta_{i-1} - \alpha_i}{\beta_i - \alpha_i}, \quad p_i = \frac{\beta_i - \beta_{i-1}}{\beta_i - \alpha_i}, \quad q_i = \frac{\alpha_{i+1} - \alpha_i}{\beta_i - \alpha_i}, \quad s_i = \frac{\beta_i - \alpha_{i+1}}{\beta_i - \alpha_i}, \tag{4.45}$$

and by evaluating (4.44) at $\beta_{i-1}$ and $\alpha_{i+1}$ we find

$$e_i^n(\beta_{i-1}) = e_{i-1}^n(\alpha_i)p_i + e_{i+1}^{n-1}(\beta_i)r_i, \tag{4.46}$$

$$e_i^n(\alpha_{i+1}) = e_{i-1}^n(\alpha_i)s_i + e_{i+1}^{n-1}(\beta_i)q_i. \tag{4.47}$$

We wish to rewrite the expressions for $e_i^n(\beta_{i-1})$ and $e_i^n(\alpha_{i+1})$ by eliminating the terms involving $e_{i-1}^n(\alpha_i)$ on the right hand side.

Evaluating (4.47) for $i = 1$, we find $e_1^n(\alpha_2) = e_2^{n-1}(\beta_1)q_1$. The right hand side only involves terms from the previous iteration, as $e_0^n(\alpha_1) = 0$. Substituting this expression into (4.46) and (4.47) for $i = 2$, the resulting expressions will again only include terms from the previous iteration. Continuing from left to right, substituting the previously obtained expressions into (4.46) and (4.47), taking absolute values and applying the triangle inequality, we obtain the sequence of inequalities

$$|e_1^n(\alpha_2)| \leq q_1|e_2^{n-1}(\beta_1)|,$$

$$|e_1^n(\alpha_2)| \leq q_1 s_2|e_2^{n-1}(\beta_1)| + q_2|e_3^{n-1}(\beta_2)|,$$

$$|e_1^n(\alpha_2)| \leq q_1 s_2 s_3|e_2^{n-1}(\beta_1)| + q_2 s_3|e_3^{n-1}(\beta_2)| + q_3|e_4^{n-1}(\beta_3)|,$$

and

$$|e_2^n(\beta_1)| \leq p_2 q_1 |e_2^{n-1}(\beta_1)| + r_2 |e_3^{n-1}(\beta_2)|,$$

$$|e_3^n(\beta_2)| \leq p_3 s_2 q_1 |e_2^{n-1}(\beta_1)| + p_3 q_2 |e_3^{n-1}(\beta_2)| + r_3 |e_4^{n-1}(\beta_3|,$$

$$|e_4^n(\beta_3)| \leq p_4 s_3 s_2 q_1 |e_2^{n-1}(\beta_1)| + p_4 s_3 q_2 |e_3^{n-1}(\beta_2)| + p_4 q_3 |e_4^{n-1}(\beta_3)|.$$

All of the above inequalities depend only on the values of $|e_j^{n-1}(\beta_k)|$. Thus, if the error measures contract to zero at each $\beta_k$, then we have the contraction at all interfaces. Writing these inequalities in matrix form gives $\boldsymbol{e}^{n+1} \leq M_{\boldsymbol{e}} \boldsymbol{e}^n$, where $\boldsymbol{e}^n = (|e_2^n(\beta_1)|, |e_3^n(\beta_2)|, |e_4^n(\beta_3)|)^T$ and

$$M_{\boldsymbol{e}} = \begin{bmatrix} p_2 q_1 & r_2 & \\ p_3 s_2 q_1 & p_3 q_2 & r_3 \\ p_4 s_3 s_2 q_1 & p_4 s_3 q_2 & p_4 q_3 \end{bmatrix}. \tag{4.48}$$

The matrices obtained for four or more subdomains will have zero entries everywhere above the first superdiagonal, i.e. they are lower Hessenberg. With (4.48) we can establish convergence.

**Theorem 4.16.** *Under the assumptions of Lemma 4.1 and the restrictions on the subdomains of $\Omega_c$ detailed above, the classical alternating Schwarz iteration* (4.42) *converges globally on* 4 *subdomains.*

*Furthermore, if all subdomains are of equal size and each pair of adjacent subdomains have an equal amount of overlap, then the iterates satisfy*

$$\max_{1 \leq i \leq 4} \|x_i^n(\xi) - x(\xi)\|_\infty \leq (\rho(M_{\boldsymbol{e}}))^n \frac{1}{\check{m}} \|\boldsymbol{e}^0\|_2,$$

*where* $\|e^0\|_2$ *is the 2-norm of vector* $e^0$ *and the contraction rate is bounded by*

$$\rho(M_e) \leq 1 - \left(\frac{\beta_i - \beta_{i-1}}{\beta_i - \alpha_i}\right)\left(\frac{\beta_{i-1} - \alpha_i}{\beta_i - \alpha_i}\right)^2.$$

*Proof.* We show that $\rho(M_e) < 1$ by computing $\|M_e\|_\infty$. As in the parallel case the quantities (4.45) are non-negative, hence the matrix $M_e$ is non-negative. As such, $\|M_e\|_\infty$ will be the maximum row sum of $M_e$. We can express each of these sums using nested products, similar to Horner's method of polynomial evaluation [28]. We have the inequalities

$$p_2 q_1 + r_2 < p_2 + r_2 = 1,$$

$$p_3(s_2 q_1 + q_2) + r_3 < p_3(s_2 + q_2) + r_3 = p_3 + r_3 = 1,$$

$$p_4(s_3(s_2 q_1 + q_2) + q_3) < p_4(s_3(s_2 + q_2) + q_3) = p_4(s_3 + q_3) = p_4 < 1,$$

hence $\|M_e\|_\infty < 1$ and convergence of the iteration follows.

If we assume all subdomains are of equal size and each pair of adjacent subdomains have equal overlap, then

$$r = s \equiv \frac{\beta_{i-1} - \alpha_i}{\beta_i - \alpha_i} \quad \text{and} \quad p = q \equiv \frac{\beta_i - \beta_{i-1}}{\beta_i - \alpha_i},$$

we have

$$M_e = \begin{bmatrix} p^2 & r & \\ p^2 r & p^2 & r \\ p^2 r^2 & p^2 r & p^2 \end{bmatrix}.$$

It follows that

$$\|M_e\|_\infty = \max\left\{p^2(1+r)+r, p^2(1+r+r^2)\right\} = p^2(1+r)+r,$$

as $r > p^2 r$. From simple calculations, we find

$$\|M_e\|_\infty = p^2(1+r)+r = 1 - \left(\frac{\beta_i - \beta_{i-1}}{\beta_i - \alpha_i}\right)\left(\frac{\beta_{i-1} - \alpha_i}{\beta_i - \alpha_i}\right)^2,$$

an upper bound on the contraction rate $\rho(M_e)$.

From the bounds on $M$, we know $|x_i^{n+1}(\xi) - x(\xi)| \leq \frac{1}{\check{m}}|e_i^{n+1}(\xi)|$. Furthermore, $|e_i^{n+1}(\xi)|$ is bounded by the maximum of its boundary values, thus:

$$\begin{aligned}
|x_i^{n+1}(\xi) - x(\xi)| &\leq \frac{1}{\check{m}}|e_i^{n+1}(\xi)| \leq \frac{1}{\check{m}}\max\left\{|e_{i+1}^n(\beta_i)|, |e_{i-1}^{n+1}(\alpha_i)|\right\} \\
&\leq \frac{1}{\check{m}}\max\left\{|e_{i+1}^n(\beta_i)|, |e_{i-1}^n(\alpha_i)|\right\} \leq \frac{1}{\check{m}}\|e^n\|_\infty \\
&\leq \frac{1}{\check{m}}\|e^n\|_2 \leq \frac{1}{\check{m}}(\rho(M_e))^n\|e^0\|_2.
\end{aligned}$$

Taking the supremum gives the bound described in the theorem statement. □

### Arbitrary Number of Subdomains

We now extend the four subdomain case to obtain the general result. We decompose $\Omega = [0,1]$ into $S > 2$ subdomains $\Omega_i = [\alpha_i, \beta_i]$ for $i = 1, \ldots, S$, where $\alpha_{i+1} < \beta_i$ for $i = 1, \ldots, S-1$ and $\beta_i < \alpha_{i+2}$ for $i = 1, \ldots, S-2$. As before $x_i(\xi)$ denotes the solution over $\Omega_i$, which is equal to the single domain solution throughout $\Omega_i$, and

satisfies

$$(M(x_i)x_{i,\xi})_\xi = 0, \quad x_i(\alpha_i) = x_{i-1}(\alpha_i), \quad x_i(\beta_i) = x_{i+1}(\beta_i), \quad \text{for} \quad i = 1, \ldots, S,$$

where $\alpha_1 = 0$, $\beta_S = 1$, and we define $x_0(\alpha_1) = 0$ and $x_{S+1}(\alpha_S) = 1$. The alternating Schwarz DD iteration is: for $n = 1, 2, \ldots$

$$\left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^n(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n-1}(\beta_i), \quad \text{for} \quad i = 1, \ldots, S, \tag{4.49}$$

having defined $x_0^n(\alpha_1) \equiv 0$ and $x_{S+1}^n(\beta_S) \equiv 1$.

The error on the $i^{\text{th}}$ subdomain at iteration $n$ is defined in (4.43), for $i = 1, \ldots, S$. As in the four subdomain case, we find an error expression on each subdomain explicitly, introducing the values $e_0^n(\alpha_1) \equiv 0$ and $e_{S+1}^n(\beta_S) \equiv 0$ for convenience.

**Lemma 4.17.** *The error on each subdomain satisfies*

$$e_i^n(\xi) = \frac{1}{\beta_i - \alpha_i}\left[(\xi - \alpha_i)e_{i+1}^{n-1}(\beta_i) + (\beta_i - \xi)e_{i-1}^n(\alpha_i)\right], \tag{4.50}$$

$\xi \in [\alpha_i, \beta_i]$ *and for* $i = 1, \ldots, S$.

Using the quantities (4.45) in (4.50) and applying the triangle inequality we obtain the error bounds in Lemma 4.18.

**Lemma 4.18.** *The error at the interface* $\xi = \beta_{i-1}$ *for* $i = 2, \ldots, S$, *satisfies*

$$|e_i^n(\beta_{i-1})| \leq p_i \sum_{j=2}^{i} \left( |e_j^{n-1}(\beta_{j-1})| q_{j-1} \prod_{k=j}^{i-1} s_k \right) + r_i |e_{i+1}^{n-1}(\beta_i)|, \tag{4.51}$$

*while at* $\xi = \alpha_{i+1}$ *for* $i = 1, \ldots, S-1$, *we have*

$$|e_i^n(\alpha_{i+1})| \leq \sum_{j=2}^{i+1} \left( |e_j^{n-1}(\beta_{j-1})| q_{j-1} \prod_{k=j}^{i} s_k \right), \tag{4.52}$$

*where we define* $\prod_{k=i}^{i-1} s_k = 1$.

*Proof.* Evaluating (4.50) at $\beta_{i-1}$ and $\alpha_{i+1}$ we find

$$e_i^n(\beta_{i-1}) = e_{i-1}^n(\alpha_i) p_i + e_{i+1}^{n-1}(\beta_i) r_i, \tag{4.53}$$

$$e_i^n(\alpha_{i+1}) = e_{i-1}^n(\alpha_i) s_i + e_{i+1}^{n-1}(\beta_i) q_i. \tag{4.54}$$

We wish to replace the terms containing $e_{i-1}^n(\alpha_i)$ with an expression involving values from the previous iteration. Starting at subdomain 1, we find $e_1^n(\alpha_2) = e_2^{n-1}(\beta_1) q_1$, as $e_0^n(\alpha_1) = 0$. This is used to evaluate (4.53) and (4.54) on subdomain 2, and the process repeats. Working from left to right, we find $e_{i-1}^n(\alpha_i)$ using the previously obtained expression for $e_{i-2}^n(\alpha_{i-2})$, arriving at

$$e_{i-1}^n(\alpha_i) = s_{i-1} s_{i-2} \cdots s_2 q_1 e_2^{n-1}(\beta_1) + s_{i-1} s_{i-2} \cdots s_3 q_2 e_3^{n-1}(\beta_2) + \cdots + q_{i-1} e_i^{n-1}(\beta_{i-1}).$$

Substituting this into (4.53) and (4.54) and applying the extended triangle inequality, noting that each of $p_i$, $q_i$, $r_i$ and $s_i$ is positive, we obtain the desired expressions. □

The error expressions (4.51) and (4.52) depend only on the values of $|e_j^{n-1}(\beta_k)|$. Hence if the error measure contracts to zero at each $\beta_k$, then we have the contraction

at all interfaces. Writing these inequalities in matrix form gives $\boldsymbol{e}^{n+1} \leq M_{\boldsymbol{e}}\boldsymbol{e}^n$, where $\boldsymbol{e}^n = (|e_2^n(\beta_1)|, |e_3^n(\beta_2)|, \ldots, |e_S^n(\beta_{S-1})|)^T$ and $M_{\boldsymbol{e}}$ is the $(S-1)\times(S-1)$ matrix

$$M_{\boldsymbol{e}} = \begin{bmatrix} p_2q_1 & r_2 & & & & \\ p_3s_2q_1 & p_3q_2 & r_3 & & & \\ p_4s_3s_2q_1 & p_4s_3q_2 & p_4q_3 & r_4 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ & & & & p_{S-1}q_{S-2} & r_{S-1} \\ p_Ss_{S-1}\cdots s_2q_1 & \cdots & & & p_Ss_{S-1}q_{S-2} & p_Sq_{S-1} \end{bmatrix}.$$

**Theorem 4.19.** *Under the assumptions of Lemma 4.1 and the restrictions on the decomposition of $\Omega_c$ detailed above, the alternating classical Schwarz iteration* (4.49) *converges globally on $S$ subdomains.*

*Furthermore, if all $S \geq 3$ subdomains are of equal size and each pair of adjacent subdomains have an equal amount of overlap, then the iterates satisfy*

$$\max_{1\leq i\leq S}\|x_i^n(\xi) - x(\xi)\|_\infty \leq (\rho(M_{\boldsymbol{e}}))^n \frac{1}{\check{m}}\|\boldsymbol{e}^0\|_2,$$

*where the contraction rate is bounded by*

$$\rho(M_{\boldsymbol{e}}) \leq 1 - \left(\frac{\beta_i - \beta_{i-1}}{\beta_i - \alpha_i}\right)\left(\frac{\beta_{i-1} - \alpha_i}{\beta_i - \alpha_i}\right)^{S-2}.$$

*Proof.* We show that $\rho(M_{\boldsymbol{e}}) < 1$. As the quantities (4.45) are non-negative, the matrix $M_{\boldsymbol{e}}$ is non-negative. The row sum

$$(p_Ss_{S-1}\cdots s_2q_1) + \cdots + p_Ss_{S-1}q_{S-2} + p_Sq_{S-1}$$

can be expressed using nested products as

$$p_S(s_{S-1}(s_{S-2}(\cdots(s_2q_1 + q_2)\cdots) + q_{S-2}) + q_{S-1}).$$

We know that each $q_i < 1$ and that $q_i + s_i = 1$. Starting at the innermost term, we have $s_2 q_1 + q_2 < s_2 + q_2 = 1$. Moving to the next set of brackets, we have $s_3(s_2 q_1 + q_2) + q_3 < s_3 + q_3 = 1$. Proceeding in this manner, we know that each term contained within brackets will be less than one in magnitude, and as such we have

$$p_S(s_{S-1}(s_{S-2}(\cdots(s_2 q_1 + q_2)\cdots) + q_{S-2}) + q_{S-1}) < p_S < 1.$$

Similarly, as $p_i + r_i = 1$, the row sum is bounded as

$$(p_{S-1}s_{S-2}\cdots s_2 q_1) + \cdots + p_{S-1}s_{S-2}q_{S-3} + p_{S-1}q_{S-2} + r_{S-1} < p_{S-1} + r_{S-1} = 1.$$

We see this holds if we consider any row of the matrix, hence we must have $\|M_e\|_\infty < 1$ and the iteration converges.

If we make the simplifying assumption that all subdomains are of equal size and each pair of adjacent subdomains have equal amounts of overlap, then we have

$$r = s \equiv \frac{\beta_{i-1} - \alpha_i}{\beta_i - \alpha_i} \quad \text{and} \quad p = q \equiv \frac{\beta_i - \beta_{i-1}}{\beta_i - \alpha_i},$$

and corresponding matrix

$$M_e = \begin{bmatrix} p^2 & r & & & & \\ p^2 r & p^2 & r & & & \\ p^2 r^2 & p^2 r & p^2 & r & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ & & & & p^2 & r \\ p^2 r^{S-2} & \cdots & & & p^2 r & p^2 \end{bmatrix}.$$

From simple calculations, we find

$$\|M_e\|_\infty = p^2(1 + \cdots + r^{S-3}) + r = 1 - \left(\frac{\beta_i - \beta_{i-1}}{\beta_i - \alpha_i}\right)\left(\frac{\beta_{i-1} - \alpha_i}{\beta_i - \alpha_i}\right)^{S-2},$$

which is an upper bound on the contraction rate. The $L^\infty$ error estimate follows as in the four subdomain case. □

### 4.2.4 A Red-Black Alternating Multidomain Method

The previous multidomain iteration is similar to the Gauss-Seidel iterative technique for solving linear systems, determining an improved approximation to each component of the solution sequentially. Gauss-Seidel can be implemented in parallel by partitioning the elements of the solution vector into different sets, solving for all members of a set simultaneously, and using these values when solving for the next set of elements. For example, we may alternately color each component either red or black and solve for all of the similarly colored components in parallel [13]. Similarly, if we appropriately partition the subdomains into two sets, we are able to solve all subdomain problems from each set in parallel while still maintaining improved convergence which is characteristic of alternating methods. As before, we decompose $\Omega_c = [0, 1]$ into $S > 2$ overlapping subdomains $\Omega_i = [\alpha_i, \beta_i]$, for $i = 1, \ldots, S$, where $\alpha_{i+1} < \beta_i$ for $i = 1, \ldots, S-1$ and $\beta_i < \alpha_{i+2}$ for $i = 1, \ldots, S-2$. We denote by $x_i(\xi)$ the original, single domain solution restricted to $\Omega_i$.

We modify our classical alternating Schwarz iteration to obtain the following, previously presented by the author in [30]. For $i = 1, \ldots, S$: if $i$ is odd, then

$$\left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n-1}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n-1}(\beta_i), \tag{4.55}$$

and if $i$ is even, then

$$\left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n}(\beta_i), \tag{4.56}$$

where $x_0^n(\alpha_1) \equiv 0$ and $x_{S+1}^n(\beta_S) \equiv 1$. If $S$ is even, then for each iteration we solve two sets of $S/2$ boundary value problems, using the results from the odd subdomains to provide updated boundary conditions for the even subdomains. If $S$ is odd, we first solve the $(S+1)/2$ odd subdomain problems, then the $(S-1)/2$ even subdomain problems. In either case, we solve all odd numbered subdomain problems in parallel, then all even numbered subdomain problems in parallel.

Error on the $i^{\text{th}}$ subdomain at iteration $n$ is defined using (4.43) for $i = 1, \ldots, S$, and we establish convergence by showing this measure contracts to zero on all subdomains. The error on each subdomain is given explicitly in the following Lemma, defining $e_0^n(\alpha_1) \equiv 0$ and $e_{S+1}^n(\beta_S) \equiv 0$.

**Lemma 4.20.** *The error on subdomain $\Omega_i = [\alpha_i, \beta_i]$, $i = 1, \ldots, S$, satisfies*

$$e_i^n(\xi) = \begin{cases} \frac{1}{\beta_i - \alpha_i}\left[(\xi - \alpha_i)e_{i+1}^{n-1}(\beta_i) + (\beta_i - \xi)e_{i-1}^{n-1}(\alpha_i)\right], & \text{if } i \text{ is odd,} \\ \frac{1}{\beta_i - \alpha_i}\left[(\xi - \alpha_i)e_{i+1}^{n}(\beta_i) + (\beta_i - \xi)e_{i-1}^{n}(\alpha_i)\right], & \text{if } i \text{ is even.} \end{cases} \tag{4.57}$$

The proof of Lemma 4.20 follows from the proof of Lemma 4.15. We use (4.57) to relate the error on subdomain $i$ at iteration $n$ to the error on subdomain $i$ and its neighbors at iteration $n-1$. Making use of the quantities (4.45) we find the following recursive relationships.

**Lemma 4.21.** *The error at the interface $\xi = \beta_{i-1}$, for $i = 2, \ldots, N$, satisfies*

$$|e_i^{n+1}(\beta_{i-1})| \leq r_i r_{i+1} |e_{i+2}^n(\beta_{i+1})| + r_i p_{i+1} |e_i^n(\alpha_{i+1})| + p_i q_{i-1} |e_i^n(\beta_{i-1})| + p_i s_{i-1} |e_{i-2}^n(\alpha_{i-1})|, \quad (4.58)$$

*while at $\xi = \alpha_{i+1}$ for $i = 1, \ldots, N-1$, we have*

$$|e_i^{n+1}(\alpha_{i+1})| \leq q_i r_{i+1} |e_{i+2}^n(\beta_{i+1})| + q_i p_{i+1} |e_i^n(\alpha_{i+1})| + s_i q_{i-1} |e_i^n(\beta_{i-1})| + s_i s_{i-1} |e_{i-2}^n(\alpha_{i-1})|. \quad (4.59)$$

*Proof.* Inequality (4.58) is obtained in the same way for the even $i$ and odd $i$ cases. By evaluating (4.57) at $\beta_{i-1}$, the error is expressed in terms of the error on subdomains of the opposite parity — if $i$ is even then $i \pm 1$ is odd, and vice-versa. We use (4.57) twice more to eliminate these terms and we obtain an expression for $e_i^{n+1}(\beta_{i-1})$ in terms of subdomain solutions of the same parity at the previous iteration. Taking absolute values, using the triangle inequality and noting $r_i, p_i, q_i$ and $s_i$ are non-negative gives the result. Inequality (4.59) is obtained in a similar way. □

The right hand sides of these bounds are identical to those obtained in the multidomain classical parallel Schwarz method (4.36), hence convergence follows immediately from the proof of Theorem 4.13, leading to the following result.

**Theorem 4.22.** *Under the assumptions of Lemma 4.1 and the restrictions on the partitioning of $\Omega_c$ detailed above, the classical Schwarz iteration (4.55 – 4.56) converges globally on an arbitrary number of subdomains.*

If we assume the overlaps are all of the same size, we have the following error estimate.

**Theorem 4.23.** *The red-black Schwarz iteration (4.55 – 4.56) on $S$ subdomains with a common overlap ratio $r \in (0, 0.5]$ converges in the infinity norm and the iterates satisfy*

$$\max_{1\le 2i\le S} \|x_{2i}^{n+1}(\xi) - x(\xi)\|_\infty \le \left(1 - 4r(1-r)\sin^2\frac{\pi}{2(S+1)}\right)^n \frac{1}{\check{m}}\|e^0\|_2,$$

$$\max_{1\le 2i+1\le S} \|x_{2i+1}^{n+1}(\xi) - x(\xi)\|_\infty \le \left(1 - 4r(1-r)\sin^2\frac{\pi}{2(S+1)}\right)^n \frac{1}{\check{m}}\|\hat{e}^0\|_2.$$

Comparing the contraction estimates of Theorem 4.23 to Theorem 4.14 for the parallel case, we see that the method (4.55 – 4.56) satisfies the same error bound in $n$ iterations that the method (4.36) will in $2n$ iterations. A single iteration of (4.55 - 4.56) requires solving one set of subdomain problems in parallel, followed by

a second set of subdomain problems solved in parallel, meaning 2 subdomain solves are required per processor for each iteration. As such, a given processor will solve $2n$ subdomain problems for both the red-black alternating algorithm and the original parallel algorithm. However, the red-black method will only require half the number of processors for each set of parallel computations.

### 4.2.5 Alternate Subdomain Groupings

Rather than classifying subdomains as odd or even, it is also possible to sort them into $G$ groups, where $G|S$, and then perform DD, solving all subdomain problems within a group simultaneously. In this sense $G = 1$ results in the parallel iteration, $G = 2$ the parallelized alternating iteration and $G = S$ the original alternating iteration.

The natural next case to consider is $G = 3$, where we classify subdomain $\Omega_i$ according to if $i \equiv 1, 2, \text{or } 3 \mod 3$, for $i = 1, \ldots, S$. Doing so, we'll have the following DD iteration: for $n = 1, 2, \ldots$

$$i \equiv 1 \mod 3: \quad \left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n-1}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n-1}(\beta_i), \tag{4.60}$$

$$i \equiv 2 \mod 3: \quad \left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n-1}(\beta_i), \tag{4.61}$$

$$i \equiv 3 \mod 3: \quad \left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n}(\beta_i). \tag{4.62}$$

**Theorem 4.24.** *Under the assumptions of Lemma 4.1 and the restrictions on the*

*decomposition of $\Omega_c$ of Section 4.2.4, the three group alternating classical Schwarz iteration (4.60 - 4.62) converges globally on S subdomains.*

*Proof.* Proceeding in the same manner as in the proof of Lemma 4.15, we can easily derive the error expressions

$$i \equiv 1 \mod 3: \quad e_i^n(\xi) = \frac{1}{\beta_i - \alpha_i}\left[(\xi - \alpha_i)e_{i+1}^{n-1}(\beta_i) + (\beta_i - \xi)e_{i-1}^{n-1}(\alpha_i)\right],$$

$$i \equiv 2 \mod 3: \quad e_i^n(\xi) = \frac{1}{\beta_i - \alpha_i}\left[(\xi - \alpha_i)e_{i+1}^{n-1}(\beta_i) + (\beta_i - \xi)e_{i-1}^{n}(\alpha_i)\right],$$

$$i \equiv 3 \mod 3: \quad e_i^n(\xi) = \frac{1}{\beta_i - \alpha_i}\left[(\xi - \alpha_i)e_{i+1}^{n}(\beta_i) + (\beta_i - \xi)e_{i-1}^{n}(\alpha_i)\right].$$

Making use of the quantities defined in (4.45) and evaluating each of these error expressions at $\beta_{i-1}$ and $\alpha_{i+1}$, we get the following expressions: for $i \equiv 1 \mod 3$:

$$e_i^n(\beta_{i-1}) = p_i e_{i-1}^{n-1}(\alpha_i) + r_i e_{i+1}^{n-1}(\beta_i),$$

$$e_i^n(\alpha_{i+1}) = s_i e_{i-1}^{n-1}(\alpha_i) + q_i e_{i+1}^{n-1}(\beta_i),$$

for $i \equiv 2 \mod 3$:

$$e_i^n(\beta_{i-1}) = p_i(s_{i-1}e_{i-2}^{n-1}(\alpha_{i-1}) + q_{i-1}e_i^{n-1}(\beta_{i-1})) + r_i e_{i+1}^{n-1}(\beta_i),$$

$$e_i^n(\alpha_{i+1}) = s_i(s_{i-1}e_{i-2}^{n-1}(\alpha_{i-1}) + q_{i-1}e_i^{n-1}(\beta_{i-1})) + q_i e_{i+1}^{n-1}(\beta_i),$$

and for $i \equiv 3 \mod 3$:

$$\begin{aligned}
e_i^n(\beta_{i-1}) &= p_i(s_{i-1}s_{i-2}e_{i-3}^{n-1}(\alpha_{i-2}) + s_{i-1}q_{i-2}e_{i-1}^{n-1}(\beta_{i-2}) \\
&\quad + q_{i-1}e_i^{n-1}(\beta_{i-1})) + r_i(p_{i+1}e_i^{n-1}(\alpha_{i+1}) + r_{i+1}e_{i+2}^{n-1}(\beta_{i+1})), \\
e_i^n(\alpha_{i+1}) &= s_i(s_{i-1}s_{i-2}e_{i-3}^{n-1}(\alpha_{i-2}) + s_{i-1}q_{i-2}e_{i-1}^{n-1}(\beta_{i-2}) \\
&\quad + q_{i-1}e_i^{n-1}(\beta_{i-1})) + q_i(p_{i+1}e_i^{n-1}(\alpha_{i+1}) + r_{i+1}e_{i+2}^{n-1}(\beta_{i+1})).
\end{aligned}$$

To simplify these expressions, we let $e_i^n(\beta_{i-1}) = y_i^{-,n}$ and $e_i^n(\alpha_{i+1}) = y_i^{+,n}$, further defining $y_0^{+,n} = 0$ and $y_{S+1}^{-,n} = 0$. Rewriting these equations, where now $i = 1\ldots, \lfloor S/3 \rfloor$, we have

$$\begin{aligned}
y_{3i-2}^{-,n} &= p_{3i}y_{3i-3}^{+,n-1} + r_{3i}y_{3i-1}^{-,n-1}, \\
y_{3i-2}^{+,n} &= s_{3i}y_{3i-3}^{+,n-1} + q_{3i}y_{3i-1}^{-,n-1}, \\
y_{3i-1}^{-,n} &= p_{3i-1}s_{3i-2}y_{3i-3}^{+,n-1} + p_{3i-1}q_{3i-2}y_{3i-1}^{-,n-1} + r_{3i-1}y_{3i}^{-,n-1}, \\
y_{3i-1}^{+,n} &= s_{3i-1}s_{3i-2}y_{3i-3}^{+,n-1} + s_{3i-1}q_{3i-2}y_{3i-1}^{-,n-1} + q_{3i-1}y_{3i}^{-,n-1}, \\
y_{3i}^{-,n} &= p_{3i}s_{3i-1}s_{3i-2}y_{3i-3}^{+,n-1} + p_{3i}s_{3i-1}q_{3i-2}y_{3i-1}^{-,n-1} \\
&\quad + p_{3i}q_{3i-1}y_{3i}^{-,n-1} + r_{3i}p_{3i+1}y_{3i}^{+,n-1} + r_{3i}r_{3i+1}y_{3i+2}^{-,n-1}, \\
y_{3i}^{+,n} &= s_{3i}s_{3i-1}s_{3i-2}y_{3i-3}^{+,n-1} + s_{3i}s_{3i-1}q_{3i-2}y_{3i-1}^{-,n-1} \\
&\quad + s_{3i}q_{3i-1}y_{3i}^{-,n-1} + q_{3i}p_{3i+1}y_{3i}^{+,n-1} + q_{3i}r_{3i+1}y_{3i+2}^{-,n-1}.
\end{aligned}$$

Inspecting the right hand sides of these equations, we see that the only unknowns

appearing at the previous iteration are $y_{3i-3}^{+,n-1}$, $y_{3i-1}^{-,n-1}$, $y_{3i}^{-,n-1}$, $y_{3i}^{+,n-1}$, and $y_{3i+2}^{-,n-1}$. Furthermore, as the pairs $y_{3i-3}^{+,n}$, $y_{3i}^{+,n}$ and $y_{3i-1}^{-,n}$, $y_{3i+2}^{-,n}$ differ in subscript by 3, they are determined by the equations of the same form. As such, we can restrict ourselves to only considering the equations for $y_{3i-1}^{-,n}$, $y_{3i}^{-,n}$ and $y_{3i}^{+,n}$, and if we can show a contraction for these equations, we will have a contraction for the entire system. Isolating the relevant terms, we have, for $i = 1 \ldots, \lfloor S/3 \rfloor$,

$$
\begin{aligned}
y_{3i-1}^{-,n} &= p_{3i-1}s_{3i-2}y_{3i-3}^{+,n-1} + p_{3i-1}q_{3i-2}y_{3i-1}^{-,n-1} + r_{3i-1}y_{3i}^{-,n-1}, \\
y_{3i}^{-,n} &= p_{3i}s_{3i-1}s_{3i-2}y_{3i-3}^{+,n-1} + p_{3i}s_{3i-1}q_{3i-2}y_{3i-1}^{-,n-1} \\
&\quad + p_{3i}q_{3i-1}y_{3i}^{-,n-1} + r_{3i}p_{3i+1}y_{3i}^{+,n-1} + r_{3i}r_{3i+1}y_{3i+2}^{-,n-1}, \\
y_{3i}^{+,n} &= s_{3i}s_{3i-1}s_{3i-2}y_{3i-3}^{+,n-1} + s_{3i}s_{3i-1}q_{3i-2}y_{3i-1}^{-,n-1} \\
&\quad + s_{3i}q_{3i-1}y_{3i}^{-,n-1} + q_{3i}p_{3i+1}y_{3i}^{+,n-1} + q_{3i}r_{3i+1}y_{3i+2}^{-,n-1}.
\end{aligned}
$$

Applying the extended triangle inequality, this leads to a matrix inequality with matrix

$$
M_e = \begin{pmatrix}
p_2q_1 & r_2 & & & & & & & & \\
p_3s_2q_1 & p_3q_2 & p_4r_3 & r_4r_3 & & & & & & \\
s_3s_2q_1 & s_3q_2 & p_4q_3 & r_4q_3 & & & & & & \\
 & & p_5s_4 & p_5q_4 & r_5 & & & & & \\
 & & p_6s_5s_4 & p_6s_5q_4 & p_6q_5 & p_7r_6 & r_7r_6 & & & \\
 & & p_6s_5s_4 & s_6s_5q_4 & s_6q_5 & p_7q_6 & r_7q_6 & & & \\
 & & & & & p_8s_7 & p_8q_7 & r_8 & & \\
 & & & & & p_9s_8s_7 & p_9s_8q_7 & p_9q_8 & p_{10}r_9 & r_{10}r_9 \\
 & & & & & s_9s_8s_7 & s_9s_8q_7 & s_9q_8 & p_{10}q_9 & r_{10}q_9 \\
 & & & & & & & & & \ddots
\end{pmatrix}.
$$

Recalling the fact that $p_i + r_i = 1$ and $s_i + q_i = 1$ it can be seen that, except for the first three and last three rows, every row sums to one, hence $\|M_e\|_\infty = 1$. As the first three rows and last three rows have row sums less than one in magnitude, we make use of Lemma 4.12 as in the proof of Theorem 4.13. We zero six rows and columns for each subsequent application of $\Psi_1(\cdot)$, hence it follows that $\rho_{M_e} < \|M_e\|_\infty = 1$. Convergence of the iteration follows. □

If we were to consider larger $G$ ($G \geq 4$) we obtain an iteration of the form: for $n = 1, 2, \ldots$

$$
\begin{aligned}
i \equiv 1 \mod G: &\quad \left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n-1}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n-1}(\beta_i), \\
i \equiv 2 \mod G: &\quad \left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n-1}(\beta_i), \\
&\qquad\qquad \vdots \\
i \equiv G \mod G: &\quad \left(M(x_i^n)x_{i,\xi}^n\right)_\xi = 0, \quad x_i^n(\alpha_i) = x_{i-1}^{n}(\alpha_i), \quad x_i^n(\beta_i) = x_{i+1}^{n}(\beta_i).
\end{aligned}
$$

This introduces more cases where the left Dirichlet condition uses information from iteration $n$ and the right condition from $n-1$. The resulting error expressions will involve more unknowns at the previous iteration and a larger system of inequalities must be analyzed, resulting in a corresponding iteration matrix $M_e$ with a filled in lower triangular part. The limiting case of this process will be the initial alternating case discussed in Section 4.2.3.

## 4.3 Optimal Schwarz Methods

As noted in the previous chapter, classical Schwarz methods experience slow convergence and require overlap for the iterations to succeed. By formulating alternative transmission conditions it is possible to fix both of these problems. Indeed, in [22] the authors present optimal parallel Schwarz transmission conditions for solving (4.1) on two non-overlapping subdomains. The "optimal" label is used to indicate that the iteration converges to the single domain solution after two iterations: two BVP solves on each subdomain, for a total of four subdomain problem solves. After reviewing this result, we present a new variation demonstrating optimal convergence on overlapping subdomains, then turn to the case of alternating iterations, previously presented by this author in [30]. By switching to the alternating iteration, we can reduce the number of subdomain solves required from four to three. Finally, we establish convergence for optimal transmission conditions for arbitrarily many subdomains.

### 4.3.1 Parallel Iterations

We decompose $\Omega_c = [0,1]$ into two non-overlapping subdomains $\Omega_1 = [0,\alpha]$ and $\Omega_2 = [\alpha,1]$ and consider the iteration [22]: for $n = 1, 2, \ldots$

$$\begin{aligned} (M(x_1^n)x_{1,\xi}^n)_\xi &= 0, \quad \xi \in \Omega_1, & (M(x_2^n)x_{2,\xi}^n)_\xi &= 0, \quad \xi \in \Omega_2, \\ x_1^n(0) &= 0, & \mathcal{B}_2(x_2^n(\alpha)) &= \mathcal{B}_2(x_1^{n-1}(\alpha)), \\ \mathcal{B}_1(x_1^n(\alpha)) &= \mathcal{B}_1(x_2^{n-1}(\alpha)), & x_2^n(1) &= 1, \end{aligned} \tag{4.63}$$

where the transmission operators are

$$\begin{aligned} \mathcal{B}_1(\cdot) &\equiv M(\cdot)\partial_\xi(\cdot) - \frac{1}{1-\alpha}\int_{(\cdot)}^{1} M(\tilde{x})\,d\tilde{x}, \\ \mathcal{B}_2(\cdot) &\equiv M(\cdot)\partial_\xi(\cdot) - \frac{1}{\alpha}\int_{0}^{(\cdot)} M(\tilde{x})\,d\tilde{x}. \end{aligned} \tag{4.64}$$

The subdomain solutions are given by (4.3), where $x_1^n(\alpha)$ and $x_2^n(\alpha)$ are determined by the transmission conditions.

**Theorem 4.25.** *Under the assumptions of Lemma 4.1, the iteration* (4.63) *with transmission conditions* (4.64) *is optimal: convergence is achieved in two iterations.*

*Proof.* Differentiating (4.5) and (4.6) with respect to $\xi$ and letting $\xi_u = \alpha$ and $\xi_\ell = \alpha$ gives

$$\begin{aligned} M(x_1^n(\xi))\partial_\xi x_1^n(\xi) &= \frac{1}{\alpha}\int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x}, \\ M(x_2^n(\xi))\partial_\xi x_2^n(\xi) &= \frac{1}{1-\alpha}\int_{x_2^n(\alpha)}^{1} M(\tilde{x})\,d\tilde{x}. \end{aligned}$$

Suppose $x_1^{n-1}(\xi)$ and $x_2^{n-1}(\xi)$ are given iterates on subdomains one and two. Enforcing the subdomain one transmission condition shows

$$\frac{1}{\alpha}\int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x} - \frac{1}{1-\alpha}\int_{x_1^n(\alpha)}^1 M(\tilde{x})\,d\tilde{x} = \frac{1}{1-\alpha}\int_{x_2^{n-1}(\alpha)}^1 M(\tilde{x})\,d\tilde{x} - \frac{1}{1-\alpha}\int_{x_2^{n-1}(\alpha)}^1 M(\tilde{x})\,d\tilde{x},$$

whence

$$\frac{1}{\alpha}\int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x} - \frac{1}{1-\alpha}\int_{x_1^n(\alpha)}^1 M(\tilde{x})\,d\tilde{x} = 0.$$

Furthermore, as

$$\int_{x_1^n(\alpha)}^1 M(\tilde{x})\,d\tilde{x} = \int_0^1 M(\tilde{x})\,d\tilde{x} - \int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x},$$

we have

$$\int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x} = \alpha\int_0^1 M(\tilde{x})\,d\tilde{x}.$$

As mentioned in the proof on Theorem 4.7, the exact solution also satisfies

$$\int_0^{x(\alpha)} M(\tilde{x})\,d\tilde{x} = \alpha\int_0^1 M(\tilde{x})\,d\tilde{x}.$$

From the positive lower bound on $M$, it follows $x_1^n(\alpha) = x(\alpha)$, hence we have obtained the correct boundary value in two iterations. Furthermore, as both $x_1^n(\xi)$ and $x(\xi)$ satisfy the BVP

$$\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) = 0, \quad x(0) = 0, \quad x(\alpha) = x(\alpha),$$

and by Lemma 4.1 this problem has a unique solution, we conclude $x_1^n(\xi) = x(\xi)$ for $\xi \in (0, \alpha)$. Convergence on subdomain two follows similarly. □

An overlapping optimal iteration can be developed in a similar manner. We now decompose $\Omega_c$ into $\Omega_1 = [0, \beta]$ and $\Omega_2 = [\alpha, 1]$, with $\alpha < \beta$, and consider the iteration: for $n = 1, 2, \ldots$

$$\begin{aligned} (M(x_1^n)x_{1,\xi}^n)_\xi = 0, \quad \xi \in \Omega_1, \qquad & (M(x_2^n)x_{2,\xi}^n)_\xi = 0, \quad \xi \in \Omega_2, \\ x_1^n(0) = 0, \qquad & B_2(x_2^n(\alpha)) = B_2(x_1^{n-1}(\alpha)), \\ B_1(x_1^n(\beta)) = B_1(x_2^{n-1}(\beta)), \qquad & x_2^n(1) = 1, \end{aligned} \tag{4.65}$$

where the transmission operators are

$$\begin{aligned} B_1(\cdot) &= M(\cdot)\partial_\xi(\cdot) - \frac{1}{1-\beta}\int_{(\cdot)}^{1} M(\tilde{x})\, d\tilde{x}, \\ B_2(\cdot) &= M(\cdot)\partial_\xi(\cdot) - \frac{1}{\alpha}\int_{0}^{(\cdot)} M(\tilde{x})\, d\tilde{x}. \end{aligned} \tag{4.66}$$

**Theorem 4.26.** *The iteration* (4.65) *with transmission conditions* (4.66) *is optimal: convergence is achieved in two iterations.*

*Proof.* Let $x_1^1(\xi)$ and $x_2^1(\xi)$ be functions on subdomain one and two, obtained from solving (4.65) with arbitrary initial functions $x_1^0(\xi)$ and $x_2^0(\xi)$. Applying Corollary 4.3 with $\xi_l = \xi_u = \beta$ to $x_1^1(\xi)$ and $x_2^1(\xi)$, differentiating (4.5) and (4.6) then evaluating

the resulting equations at $\beta$, we obtain

$$M(x_1^1(\beta))\partial_\xi x_1^1(\beta) = \frac{1}{\beta}\int_0^{x_1^1(\beta)} M(\tilde{x})\,d\tilde{x},$$
$$M(x_2^1(\beta))\partial_\xi x_2^1(\beta) = \frac{1}{1-\beta}\int_{x_2^1(\beta)}^1 M(\tilde{x})\,d\tilde{x}.$$

The transmission condition on $\Omega_1$ implies

$$\frac{1}{\beta}\int_0^{x_1^2(\beta)} M(\tilde{x})\,d\tilde{x} - \frac{1}{1-\beta}\int_{x_1^2(\beta)}^1 M(\tilde{x})\,d\tilde{x}$$
$$= \frac{1}{1-\beta}\int_{x_2^1(\beta)}^1 M(\tilde{x})\,d\tilde{x} - \frac{1}{1-\beta}\int_{x_2^1(\beta)}^1 M(\tilde{x})\,d\tilde{x} = 0,$$

hence

$$\int_0^{x_1^2(\beta)} M(\tilde{x})\,d\tilde{x} = \frac{\beta}{1-\beta}\int_{x_1^2(\beta)}^1 M(\tilde{x})\,d\tilde{x}.$$

Using the fact

$$\int_{x_1^2(\beta)}^1 M(\tilde{x})\,d\tilde{x} = \int_0^1 M(\tilde{x})\,d\tilde{x} - \int_0^{x_1^2(\beta)} M(\tilde{x})\,d\tilde{x},$$

we have

$$\int_0^{x_1^2(\beta)} M(\tilde{x})\,d\tilde{x} = \beta\int_0^1 M(\tilde{x})\,d\tilde{x}.$$

From Corollary 4.2 the exact solution also satisfies

$$\int_0^{x(\beta)} M(\tilde{x})\,d\tilde{x} = \beta\int_0^1 M(\tilde{x})\,d\tilde{x} = \int_0^{x_1^2(\beta)} M(\tilde{x})\,d\tilde{x},$$

hence $x_1^2(\beta) = x(\beta)$ and we have convergence at the boundary. Convergence over interior of $\Omega_1$ follows from Lemma 4.1.

On $\Omega_2$, applying Corollary 4.3 with $\xi_l = \xi_u = \alpha$ to $x_1^1(\xi)$ and $x_2^2(\xi)$, differentiating, and then evaluating the resulting equations at $\alpha$, we obtain

$$M(x_1^1(\alpha))\partial_\xi x_1^1(\alpha) = \frac{1}{\alpha}\int_0^{x_1^1(\alpha)} M(\tilde{x})\,d\tilde{x},$$
$$M(x_2^2(\alpha))\partial_\xi x_2^2(\alpha) = \frac{1}{1-\alpha}\int_{x_2^2(\alpha)}^1 M(\tilde{x})\,d\tilde{x}.$$

The transmission condition requires

$$\frac{1}{1-\alpha}\int_{x_2^2(\alpha)}^1 M(\tilde{x})\,d\tilde{x} - \frac{1}{\alpha}\int_0^{x_2^2(\alpha)} M(\tilde{x})\,d\tilde{x}$$
$$= \frac{1}{\alpha}\int_0^{x_1^1(\alpha)} M(\tilde{x})\,d\tilde{x} - \frac{1}{\alpha}\int_0^{x_1^1(\alpha)} M(\tilde{x})\,d\tilde{x} = 0,$$

hence

$$\int_0^{x_2^2(\alpha)} M(\tilde{x})\,d\tilde{x} = \frac{\alpha}{1-\alpha}\int_{x_2^n(\alpha)}^1 M(\tilde{x})\,d\tilde{x}.$$

Making use of Corollary 4.2 as before, we find $x_2^2(\alpha) = x(\alpha)$, establishing convergence at the boundary. As before, convergence over the interior of $\Omega_2$ follows from Lemma 4.1. □

### 4.3.2 Alternating Iterations

The alternating optimal Schwarz iterations, for both non-overlapping and overlapping subdomains, follow from their parallel counterparts. In the non-overlapping case, we once again decompose $\Omega_c$ into $\Omega_1 = [0, \alpha]$ and $\Omega_2 = [\alpha, 1]$, and consider the iteration:

for $n = 1, 2, \ldots$

$$\begin{aligned} (M(x_1^n)x_{1,\xi}^n)_\xi = 0, \ \ \xi \in \Omega_1, \qquad & (M(x_2^n)x_{2,\xi}^n)_\xi = 0, \ \ \xi \in \Omega_2, \\ x_1^n(0) = 0, \qquad & B_2(x_2^n(\alpha)) = B_2(x_1^n(\alpha)), \\ B_1(x_1^n(\alpha)) = B_1(x_2^{n-1}(\alpha)), \qquad & x_2^n(1) = 1, \end{aligned} \tag{4.67}$$

where

$$\begin{aligned} B_1(\cdot) &= M(\cdot)\partial_\xi(\cdot) - \frac{1}{1-\alpha}\int_{(\cdot)}^1 M(\tilde{x})\,d\tilde{x}, \\ B_2(\cdot) &= M(\cdot)\partial_\xi(\cdot) - \frac{1}{\alpha}\int_0^{(\cdot)} M(\tilde{x})\,d\tilde{x}. \end{aligned} \tag{4.68}$$

As we show in the following Theorem, convergence is theoretically achieved after solving 3 subdomain problems in the alternating case: one fewer subdomain solve than what is required in the parallel optimal method.

**Theorem 4.27.** *The alternating iteration* (4.67) *with transmission conditions* (4.68) *is optimal: convergence is achieved in three subdomain solves.*

*Proof.* Applying Corollary 4.3 to $x_{1,2}^n(\xi)$, $n \geq 1$, differentiating the resulting equations, then evaluating the result at $\alpha$, we find

$$\begin{aligned} M(x_1^n(\alpha))\partial_\xi x_1^n(\alpha) &= \frac{1}{\alpha}\int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x}, \\ M(x_2^n(\alpha))\partial_\xi x_2^n(\alpha) &= \frac{1}{1-\alpha}\int_{x_2^n(\alpha)}^1 M(\tilde{x})\,d\tilde{x}. \end{aligned}$$

Suppose $x_1^1(\xi)$ has been obtained by solving (4.67) on $\Omega_1$ using an arbitrary initial function $x_2^0(\xi)$. Substituting the expressions previously obtained into the transmission condition for $\Omega_2$, with $n = 1$, gives

$$\frac{1}{1-\alpha}\int_{x_2^1(\alpha)}^{1} M(\tilde{x})\,d\tilde{x} - \frac{1}{\alpha}\int_{0}^{x_2^1(\alpha)} M(\tilde{x})\,d\tilde{x} = \frac{1}{\alpha}\int_{0}^{x_1^1(\alpha)} M(\tilde{x})\,d\tilde{x} - \frac{1}{\alpha}\int_{0}^{x_1^1(\alpha)} M(\tilde{x})\,d\tilde{x} = 0,$$

and hence

$$\int_{0}^{x_2^1(\alpha)} M(\tilde{x})\,d\tilde{x} = \frac{\alpha}{1-\alpha}\int_{x_2^1(\alpha)}^{1} M(\tilde{x})\,d\tilde{x}.$$

Noting that

$$\int_{x_2^1(\alpha)}^{1} M(\tilde{x})\,d\tilde{x} = \int_{0}^{1} M(\tilde{x})\,d\tilde{x} - \int_{0}^{x_2^1(\alpha)} M(\tilde{x})\,d\tilde{x},$$

we have

$$\int_{0}^{x_2^1(\alpha)} M(\tilde{x})\,d\tilde{x} = \alpha\int_{0}^{1} M(\tilde{x})\,d\tilde{x}.$$

By Corollary 4.2 we see that

$$\int_{0}^{x_2^1(\alpha)} M(\tilde{x})\,d\tilde{x} = \int_{0}^{x(\alpha)} M(\tilde{x})\,d\tilde{x},$$

and hence $x_2^1(\alpha) = x(\alpha)$, thus we have convergence at the boundary. Convergence in the interior of $\Omega_2$ follows from Lemma 4.1.

Returning to $\Omega_1$, the transmission condition for $x_1^2(\xi)$ requires

$$\frac{1}{\alpha}\int_0^{x_1^2(\alpha)} M(\tilde{x})\,d\tilde{x} - \frac{1}{1-\alpha}\int_{x_1^2(\alpha)}^1 M(\tilde{x})\,d\tilde{x} = \frac{1}{1-\alpha}\int_0^{x(\alpha)} M(\tilde{x})\,d\tilde{x} - \frac{1}{1-\alpha}\int_0^{x(\alpha)} M(\tilde{x})\,d\tilde{x} = 0,$$

which implies

$$\int_0^{x_1^2(\alpha)} M(\tilde{x})\,d\tilde{x} = \frac{\alpha}{1-\alpha}\int_{x_1^2(\alpha)}^1 M(\tilde{x})\,d\tilde{x}.$$

It follows from the previous equation, combined with Corollary 4.2, that

$$\int_0^{x_1^2(\alpha)} M(\tilde{x})\,d\tilde{x} = \alpha\int_0^{x(\alpha)} M(\tilde{x})\,d\tilde{x},$$

hence $x_1^2(\alpha) = x(\alpha)$. As before, convergence over the interior follows from Lemma 4.1 and thus the solution has been obtained in three subdomain solves. □

The final two subdomain variation to consider is for an alternating iteration on overlapping subdomains. Let $\Omega_1 = [0, \beta]$ and $\Omega_2 = [\alpha, 1]$, with $\alpha < \beta$, and consider the iteration: for $n = 1, 2, \ldots$

$$\begin{aligned}
(M(x_1^n)x_{1,\xi}^n)_\xi = 0,\ \ \xi \in \Omega_1, \qquad & (M(x_2^n)x_{2,\xi}^n)_\xi = 0,\ \ \xi \in \Omega_2, \\
x_1^n(0) = 0, \qquad & B_2(x_2^n(\alpha)) = B_2(x_1^n(\alpha)), \\
B_1(x_1^n(\beta)) = B_1(x_2^{n-1}(\beta)), \qquad & x_2^n(1) = 1,
\end{aligned} \tag{4.69}$$

where

$$\begin{aligned} B_1(\cdot) &= M(\cdot)\partial_\xi(\cdot) - \frac{1}{1-\beta}\int_{(\cdot)}^{1} M(\tilde{x})\,d\tilde{x}, \\ B_2(\cdot) &= M(\cdot)\partial_\xi(\cdot) - \frac{1}{\alpha}\int_{0}^{(\cdot)} M(\tilde{x})\,d\tilde{x}. \end{aligned} \tag{4.70}$$

**Theorem 4.28.** *The alternating iteration* (4.69) *with transmission conditions* (4.70) *is optimal: convergence is achieved in three subdomain solves.*

The proof is a simple combination of the methods used to prove Theorems 4.26 and 4.27, and hence is omitted.

### 4.3.3 A Parallel Iteration for Three Non-Overlapping Subdomains

It is possible to extend the parallel optimal iteration to three non-overlapping subdomains, where in this case "optimal" indicates that convergence is obtained in three DD iterations. We decompose $\Omega_c$ into $\Omega_1 = [0, \alpha_1]$, $\Omega_2 = [\alpha_1, \alpha_2]$, and $\Omega_3 = [\alpha_2, 1]$, and over each subdomain we solve the differential equation

$$(M(x_i^n)x_{i,\xi}^n)_\xi = 0, \quad \xi \in \Omega_i, \tag{4.71}$$

with conditions $x_1^n(0) = 0$ and $x_3^n(1) = 1$. Furthermore, we require a pair of transmission conditions at both interfaces. At $\xi = \alpha_1$ we impose the conditions

$$M(x_1^n)x_{1,\xi}^n - \frac{1}{\alpha_2 - \alpha_1}\int_{x_1^n(\alpha_1)}^{x_2^{n-1}(\alpha_2)} M(\tilde{x})\,d\tilde{x} = M(x_2^{n-1})x_{2,\xi}^{n-1} - \frac{1}{\alpha_2 - \alpha_1}\int_{x_2^{n-1}(\alpha_1)}^{x_2^{n-1}(\alpha_2)} M(\tilde{x})\,d\tilde{x}, \quad (4.72)$$

$$M(x_2^n)x_{2,\xi}^n - \frac{1}{\alpha_1}\int_0^{x_2^n(\alpha_1)} M(\tilde{x})\,d\tilde{x} = M(x_1^{n-1})x_{1,\xi}^{n-1} - \frac{1}{\alpha_1}\int_0^{x_1^{n-1}(\alpha_1)} M(\tilde{x})\,d\tilde{x}, \quad (4.73)$$

and at $\xi = \alpha_2$ we use

$$M(x_2^n)x_{2,\xi}^n - \frac{1}{1 - \alpha_2}\int_{x_2^n(\alpha_2)}^{1} M(\tilde{x})\,d\tilde{x} = M(x_3^{n-1})x_{3,\xi}^{n-1} - \frac{1}{1 - \alpha_2}\int_{x_3^{n-1}(\alpha_2)}^{1} M(\tilde{x})\,d\tilde{x}, \quad (4.74)$$

$$M(x_3^n)x_{3,\xi}^n - \frac{1}{\alpha_2 - \alpha_1}\int_{x_2^{n-1}(\alpha_1)}^{x_3^n(\alpha_2)} M(\tilde{x})\,d\tilde{x} = M(x_2^{n-1})x_{2,\xi}^{n-1} - \frac{1}{\alpha_2 - \alpha_1}\int_{x_2^{n-1}(\alpha_1)}^{x_2^{n-1}(\alpha_2)} M(\tilde{x})\,d\tilde{x}. \quad (4.75)$$

**Theorem 4.29.** *The parallel Schwarz iteration* (4.71) *with transmission conditions (4.72–4.75) is optimal: convergence is achieved in three iterations.*

*Proof.* By direct integration of (4.71) over each subdomain, it is clear the following equations must hold:

$$M(x_1^n)x_{1,\xi}^n = \frac{1}{\alpha_1}\int_0^{x_1^n(\alpha_1)} M(\tilde{x})\,d\tilde{x},$$
$$M(x_2^n)x_{2,\xi}^n = \frac{1}{\alpha_2-\alpha_1}\int_{x_2^n(\alpha_1)}^{x_2^n(\alpha_2)} M(\tilde{x})\,d\tilde{x},$$
$$M(x_3^n)x_{3,\xi}^n = \frac{1}{1-\alpha_2}\int_{x_3^n(\alpha_2)}^{1} M(\tilde{x})\,d\tilde{x},$$

where the limits at $\alpha_1$ and $\alpha_2$ are determined by the transmission conditions. Using these equations we are able to eliminate all non-integral terms in (4.72–4.75).

Let $x_i^1(\xi)$, $i = 1, 2, 3$, be determined using arbitrary initial data in the transmission conditions (4.72 – 4.75) for the first DD iteration. We now use these functions to evaluate the transmission conditions for the second iteration. Of particular interest are (4.73) and (4.74), which, when simplified, give

$$\int_0^{x_2^2(\alpha_1)} M(\tilde{x})\,d\tilde{x} = \alpha_1 C \quad \text{and} \quad \int_0^{x_2^2(\alpha_2)} M(\tilde{x})\,d\tilde{x} = \alpha_2 C, \tag{4.76}$$

where

$$C = \int_0^1 M(\tilde{x})\,d\tilde{x}.$$

Finally, performing the third DD iteration, the transmission conditions, once simpli-

fied, give the following equations for the boundary conditions:

$$\frac{1}{\alpha_1}\int_0^{x_1^3(\alpha_1)} M(\tilde{x})\,d\tilde{x} = \frac{1}{\alpha_2}\int_0^{x_2^2(\alpha_2)} M(\tilde{x})\,d\tilde{x}, \tag{4.77}$$

$$\int_0^{x_2^3(\alpha_1)} M(\tilde{x})\,d\tilde{x} = \alpha_1 C,$$

$$\int_0^{x_2^3(\alpha_2)} M(\tilde{x})\,d\tilde{x} = \alpha_2 C,$$

$$(1-\alpha_1)\int_0^{x_3^3(\alpha_2)} M(\tilde{x})\,d\tilde{x} = (1-\alpha_2)\int_0^{x_2^2(\alpha_1)} M(\tilde{x})\,d\tilde{x} + (\alpha_2-\alpha_1)C. \tag{4.78}$$

Substituting the results of the second iteration, (4.76), into (4.77) and (4.78), we find

$$\int_0^{x_1^3(\alpha_1)} M(\tilde{x})\,d\tilde{x} = \int_0^{x_2^3(\alpha_1)} M(\tilde{x})\,d\tilde{x} = \alpha_1 C,$$
$$\int_0^{x_2^3(\alpha_2)} M(\tilde{x})\,d\tilde{x} = \int_0^{x_3^3(\alpha_2)} M(\tilde{x})\,d\tilde{x} = \alpha_2 C.$$

It follows that $x_1^3(\alpha_1) = x_2^3(\alpha_1)$ and $x_2^3(\alpha_2) = x_3^3(\alpha_2)$. Furthermore, from Corollary 4.2, we know the single domain solution $x(\xi)$ satisfies

$$\int_0^{x(\alpha_1)} M(\tilde{x})\,d\tilde{x} = \alpha_1 C \quad \text{and} \quad \int_0^{x(\alpha_2)} M(\tilde{x})\,d\tilde{x} = \alpha_2 C,$$

hence we conclude $x_1^3(\alpha_1) = x_2^3(\alpha_1) = x(\alpha_1)$ and $x_2^3(\alpha_2) = x_3^3(\alpha_2) = x(\alpha_2)$. The transmission conditions give the exact boundary values in three iterations, and hence are optimal. As in the two subdomain case, convergence over the interior of subdomains follows from the Lemma 4.1. □

### 4.3.4 A Parallel Iteration for Multiple Non-Overlapping Subdomains

We consider the following extension of the optimal iteration to $S \geq 4$ subdomains. We decompose $\Omega_c$ into subdomains $\Omega_i = [\alpha_{i-1}, \alpha_i]$ for $i = 1, \ldots, S$, defining $\alpha_0 = 0$ and $\alpha_S = 1$. For $i = 1, \ldots, S$ we solve the equation

$$(M(x_i^n)x_{i,\xi}^n)_\xi = 0, \quad \xi \in \Omega_i, \tag{4.79}$$

with original Dirichlet conditions $x_1^n(0) = 0$ and $x_S^n(1) = 1$. Finally, we enforce the nonlinear transmission conditions

$$M(x_i^n)x_{i,\xi}^n - \frac{1}{\alpha_{i-1}} \int_0^{x_i^n(\alpha_{i-1})} M(\tilde{x})\, d\tilde{x} = M(x_{i-1}^{n-1})x_{i-1,\xi}^{n-1} - \frac{1}{\alpha_{i-1}} \int_0^{x_{i-1}^{n-1}(\alpha_{i-1})} M(\tilde{x})\, d\tilde{x}, \tag{4.80}$$

at $\xi = \alpha_{i-1}$, for $i = 2, \ldots, S$, and

$$M(x_i^n)x_{i,\xi}^n - \frac{1}{1-\alpha_i} \int_{x_i^n(\alpha_i)}^1 M(\tilde{x})\, d\tilde{x} = M(x_{i+1}^{n-1})x_{i+1,\xi}^{n-1} - \frac{1}{1-\alpha_i} \int_{x_{i+1}^{n-1}(\alpha_i)}^1 M(\tilde{x})\, d\tilde{x}, \tag{4.81}$$

at $\xi = \alpha_i$, for $i = 1, \ldots, S-1$.

By direct integration, we know that each function $x_i^n(\xi)$ satisfies

$$M(x_i^n)x_{i,\xi}^n = \frac{1}{\alpha_i - \alpha_{i-1}} \int_{x_i^n(\alpha_{i-1})}^{x_i^n(\alpha_i)} M(\tilde{x})\, d\tilde{x},$$

given boundary values $x_i^n(\alpha_{i-1})$ and $x_i^n(\alpha_i)$. To simplify expressions we introduce the notation

$$z_i^{-,n} = \int_0^{x_i^n(\alpha_{i-1})} M(\tilde{x})\, d\tilde{x} \quad \text{and} \quad z_i^{+,n} = \int_0^{x_i^n(\alpha_i)} M(\tilde{x})\, d\tilde{x}.$$

Substituting these expressions for $x_i^n$, $x_{i-1}^n$, and $x_{i+1}^n$ into the transmission conditions and rearranging, we obtain the equations

$$\begin{aligned} &\alpha_{i-1}(\alpha_{i-1}-\alpha_{i-2})z_i^{+,n} - \alpha_i(\alpha_{i-1}-\alpha_{i-2})z_i^{-,n} = \\ &\alpha_{i-2}(\alpha_i-\alpha_{i-1})z_{i-1}^{+,n-1} - \alpha_{i-1}(\alpha_i-\alpha_{i-1})z_{i-1}^{-,n-1}, \end{aligned} \tag{4.82}$$

for $i = 2, \ldots, S$, and

$$\begin{aligned} &(1-\alpha_{i-1})(\alpha_{i+1}-\alpha_i)z_i^{+,n} - (1-\alpha_i)(\alpha_{i+1}-\alpha_i)z_i^{-,n} = \\ &(1-\alpha_i)(\alpha_i-\alpha_{i-1})z_{i+1}^{+,n-1} - (1-\alpha_{i+1})(\alpha_i-\alpha_{i-1})z_{i+1}^{-,n-1}, \end{aligned} \tag{4.83}$$

for $i = 1, \ldots, S-1$.

Multiplying (4.82) by $(1-\alpha_{i-1})(\alpha_{i+1}-\alpha_i)$ and (4.83) by $\alpha_{i-1}(\alpha_{i-1}-\alpha_{i-2})$, then taking the difference of the resulting equations, we obtain

$$\begin{aligned} z_i^{-,n} =& \alpha_{i-1}\left(\frac{1-\alpha_{i-1}}{\alpha_{i-1}-\alpha_{i-2}}\right) z_{i-1}^{-,n-1} - \alpha_{i-2}\left(\frac{1-\alpha_{i-1}}{\alpha_{i-1}-\alpha_{i-2}}\right) z_{i-1}^{+,n-1} \\ &- \alpha_{i-1}\left(\frac{1-\alpha_{i+1}}{\alpha_{i+1}-\alpha_i}\right) z_{i+1}^{-,n-1} + \alpha_{i-1}\left(\frac{1-\alpha_i}{\alpha_{i+1}-\alpha_i}\right) z_{i+1}^{+,n-1}. \end{aligned} \tag{4.84}$$

Similarly, multiplying (4.82) by $(1-\alpha_i)(\alpha_{i+1}-\alpha_i)$ and (4.83) by $\alpha_i(\alpha_{i-1}-\alpha_{i-2})$, then

taking the difference, we obtain

$$\begin{aligned} z_i^{+,n} =& \alpha_{i-1}\left(\frac{1-\alpha_i}{\alpha_{i-1}-\alpha_{i-2}}\right) z_{i-1}^{-,n-1} - \alpha_{i-2}\left(\frac{1-\alpha_i}{\alpha_{i-1}-\alpha_{i-2}}\right) z_{i-1}^{+,n-1} \\ & - \alpha_i\left(\frac{1-\alpha_{i+1}}{\alpha_{i+1}-\alpha_i}\right) z_{i+1}^{-,n-1} + \alpha_i\left(\frac{1-\alpha_i}{\alpha_{i+1}-\alpha_i}\right) z_{i+1}^{+,n-1}, \end{aligned} \tag{4.85}$$

where both (4.84) and (4.85) have subscripts $i = 3, \ldots, S-2$. The cases of $i = 1, 2, S-1$, and $S$ are somewhat different, they are as follows:

$$\begin{aligned} z_1^{+,n} &= -\alpha_1\left(\frac{1-\alpha_2}{\alpha_2-\alpha_1}\right) z_2^{-,n-1} + \alpha_1\left(\frac{1-\alpha_1}{\alpha_2-\alpha_1}\right) z_2^{+,n-1}, \\ z_2^{-,n} &= -\alpha_1\left(\frac{1-\alpha_3}{\alpha_3-\alpha_2}\right) z_3^{-,n-1} + \alpha_1\left(\frac{1-\alpha_2}{\alpha_3-\alpha_2}\right) z_3^{+,n-1}, \\ z_2^{+,n} &= -\alpha_2\left(\frac{1-\alpha_3}{\alpha_3-\alpha_2}\right) z_3^{-,n-1} + \alpha_2\left(\frac{1-\alpha_2}{\alpha_3-\alpha_2}\right) z_3^{+,n-1}, \\ z_{S-1}^{-,n} &= \alpha_{S-2}\left(\frac{1-\alpha_{S-2}}{\alpha_{S-2}-\alpha_{S-3}}\right) z_{S-2}^{-,n-1} - \alpha_{S-3}\left(\frac{1-\alpha_{S-2}}{\alpha_{S-2}-\alpha_{S-3}}\right) z_{S-2}^{+,n-1} + \alpha_{S-2}C, \\ z_{S-1}^{+,n} &= \alpha_{S-2}\left(\frac{1-\alpha_{S-1}}{\alpha_{S-2}-\alpha_{S-3}}\right) z_{S-2}^{-,n-1} - \alpha_{S-3}\left(\frac{1-\alpha_{S-1}}{\alpha_{S-2}-\alpha_{S-3}}\right) z_{S-2}^{+,n-1} + \alpha_{S-1}C, \\ z_S^{-,n} &= \alpha_{S-1}\left(\frac{1-\alpha_{S-1}}{\alpha_{S-1}-\alpha_{S-2}}\right) z_{S-1}^{-,n-1} - \alpha_{S-2}\left(\frac{1-\alpha_{S-1}}{\alpha_{S-1}-\alpha_{S-2}}\right) z_{S-1}^{+,n-1}, \end{aligned} \tag{4.86}$$

where

$$C = \int_0^1 M(\tilde{x})\, d\tilde{x}.$$

The difference equations (4.84–4.86) can be expressed as a matrix iteration

$$z^n = Az^{n-1} + b,$$

where

$$z^n = [z_1^{+,n}, z_2^{-,n}, z_2^{+,n}, \ldots, z_i^{-,n}, z_i^{+,n}, \ldots, z_{S-1}^{-,n}, z_{S-1}^{+,n}, z_S^{-,n}]^T.$$

For $S \geq 3$ subdomains, the matrix $A$ will be of size $2(S-1) \times 2(S-1)$ with nonzero blocks as follows:

$$A = \begin{pmatrix} 0 & \times & \times & 0 & 0 & \cdots & & & & & 0 \\ & & & U_2 & & & & & & & \\ & L_3 & & & U_3 & & & & & & \\ & & & L_4 & & U_4 & & & & & \\ & & & & L_5 & & U_5 & & & & \\ & & & & & \ddots & & \ddots & & & \\ & & & & & & L_{S-2} & & U_{S-2} & & \\ & & & & & & & L_{S-1} & & & \\ 0 & \cdots & & & & & & 0 & 0 & \times & \times & 0 \end{pmatrix}$$

The interior rows and columns of $A$ can be partitioned into a grid of $(S-2)^2$ blocks, each of dimension $2 \times 2$. The matrices $U_i$ and $L_i$ are $2 \times 2$ blocks of the form

$$U_i = \begin{pmatrix} -\alpha_{i-1}\left(\frac{1-\alpha_{i+1}}{\alpha_{i+1}-\alpha_i}\right) & \alpha_{i-1}\left(\frac{1-\alpha_i}{\alpha_{i+1}-\alpha_i}\right) \\ -\alpha_i\left(\frac{1-\alpha_{i+1}}{\alpha_{i+1}-\alpha_i}\right) & \alpha_i\left(\frac{1-\alpha_i}{\alpha_{i+1}-\alpha_i}\right) \end{pmatrix}$$

for $i = 2, \ldots, S-2$, and

$$L_i = \begin{pmatrix} \alpha_{i-1}\left(\frac{1-\alpha_{i-1}}{\alpha_{i-1}-\alpha_{i-2}}\right) & -\alpha_{i-2}\left(\frac{1-\alpha_{i-1}}{\alpha_{i-1}-\alpha_{i-2}}\right) \\ \alpha_{i-1}\left(\frac{1-\alpha_i}{\alpha_{i-1}-\alpha_{i-2}}\right) & -\alpha_{i-2}\left(\frac{1-\alpha_i}{\alpha_{i-1}-\alpha_{i-2}}\right) \end{pmatrix}$$

for $i = 3, \ldots, S-1$. We shall refer to the matrices $U_i$ as being in the first block superdiagonal and the matrices $L_i$ as being in the first block subdiagonal, with individual matrices $U_i$ and $L_i$ being superdiagonal blocks and subdiagonal blocks, respectively.

Similarly we may refer to the $k^{\text{th}}$ block subdiagonal or superdiagonal. The elements of $A$ denoted by $\times$ are the coefficients from the equations for $z_1^{+,n}$ and $z_S^{-,n}$, but it will be shown that they can essentially be ignored in future calculations. All omitted entries are zero.

To establish convergence, we denote the exact solution of this iteration by $z^*$ and hence define the error at each iteration by $e^n = z^n - z^*$. This error measure satisfies the iteration $e^n = Ae^{n-1}$. As with all previous iterations, we wish to show the error measure contracts to zero. For convergence, it is sufficient that $\rho(A) < 1$. However, for optimal methods we require the error to be zero after finitely many steps, so there must be some $\tilde{n} \in \mathbb{N}$ such that $A^{\tilde{n}} \equiv 0$. That is, we require the matrix $A$ to be *nilpotent*. Before establishing this result, we first give a useful Lemma regarding blocks $U_i$ and $L_i$.

**Lemma 4.30.** *For $i = 2, \ldots, S-2$, we have $U_i L_{i+1} = 0_{2\times 2} = L_{i+1} U_i$.*

This follows immediately from direct calculation, and it is due to this convenient result that we can establish that the proposed iterations are indeed optimal.

**Theorem 4.31.** *The parallel Schwarz iteration* (4.79) *on $S \geq 4$ subdomains with transmission conditions (4.80–4.81) is optimal: convergence is achieved in $S$ iterations.*

*Proof.* Convergence of this iteration will follow from the nilpotence of $A$. We will demonstrate that $A^{S-1} \equiv 0$ for all $S \geq 4$. To do so, we establish the form of $A^k$ for $k = 1, 2, \ldots$ via induction. As in the matrix $A$, the $2(S-2)$ interior rows and $2(S-2)$ interior columns of $A^k$ will be divided into a grid of $(S-2)^2$ blocks, each of size $2 \times 2$. We number block rows and block columns from 2 through $S-1$, starting at the top left corner. The $(i,j)$ *block* of $A$ will be denoted by $A_{i,j}$, and the $(i,j)$ *element* of $A$ will be denoted by $A(i,j)$.

It is our claim that $A^k$ will have nonzero blocks only in $k^{\text{th}}$ block subdiagonal and block superdiagonal. Specifically, the superdiagonal block in the $(i, i+k)$ position will be

$$B_{i,i+k} = U_i U_{i+1} \cdots U_{i+k-1}, \quad i = 2, \ldots, S-k-1, \tag{4.87}$$

and the subdiagonal block in the $(i+k, i)$ position will be

$$B_{i+k,i} = L_{i+k} \cdots L_{i+2} L_{i+1}, \quad i = 2, \ldots, S-k-1. \tag{4.88}$$

It is clear that these nonzero blocks will only occur for $k \leq S-3$. For $k \geq S-2$ there will be no nonzero blocks in the interior of the matrix $A^k$.

Finally, $A^k$ will have four other nonzero elements for $k \leq S-2$: $A(1, 2k)$, $A(1, 2k+1)$, $A(2S-2, 2S-2k-2)$, and $A(2S-2, 2S-2k-1)$. When $k = S-1$, we will have $A^k \equiv 0$, and hence convergence in $S$ iterations.

To illustrate the sequence of matrices generated, we consider the case of $S = 6$, producing a $4 \times 4$ grid of blocks in the matrix interior. In this case:

$$A = \begin{pmatrix} 0 & \times & \times & \cdots & 0 & 0 & 0 \\ \vdots & & U_2 & & & & \vdots \\ & L_3 & & U_3 & & & \\ & & L_4 & & U_4 & & \\ & & & L_5 & & & \\ 0 & \cdots & & 0 & 0 & \times & \times & 0 \end{pmatrix},$$

$$A^2 = \begin{pmatrix} 0 & 0 & 0 & \times & \times & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & U_2U_3 & & & & \vdots \\ & & & & & & U_3U_4 & & & \\ & L_4L_3 & & & & & & & & \\ & & & L_5L_4 & & & & & & \\ 0 & 0 & 0 & 0 & 0 & \times & \times & 0 & 0 & 0 \end{pmatrix},$$

$$A^3 = \begin{pmatrix} 0 & \cdots & 0 & \times & \times & 0 & 0 & 0 \\ \vdots & & & & & & U_2U_3U_4 & \vdots \\ & & & & & & & \\ & L_5L_4L_3 & & & & & & \\ 0 & 0 & 0 & \times & \times & 0 & \cdots & 0 \end{pmatrix},$$

and

$$A^4 = \left(\begin{array}{c|ccc|c} 0 & \cdots & 0 & \times & \times & 0 \\ \hline \vdots & & & & & \vdots \\ \hline 0 & \times & \times & 0 & \cdots & 0 \end{array}\right).$$

To establish the veracity of our claim, we proceed by induction, noting that $A^1$ satisfies the given description. Assuming now that $A^k$ is as described, we prove that $A^{k+1}$ is also of this form, and hence $A^k$ is as described for all positive integers $k$. Writing $A^{k+1} = A \times A^k$, we note that $A$ can be expressed as the sum of matrices $L$ and $U$, where $L$ is block lower triangular, containing all $L_i$ blocks, and $U$ is block upper triangular, containing all $U_i$ blocks.

The matrix $U$ shifts the $i^{\text{th}}$ row of blocks in $A^k$ to row $i-1$ (one block up) and left multiplies this row by $U_{i-1}$. This deletes the top row of blocks in $A^k$ as a result, and introduces zeros in the bottom row of blocks. This shifts the $k^{\text{th}}$ block superdiagonal of $A^k$ to the $k+1^{\text{th}}$ block superdiagonal, where

$$A^{k+1}_{i-1,i+k} = U_{i-1}B_{i,i+k} = U_{i-1}U_iU_{i+1}\cdots U_{i+k-1} = B_{i-1,i-1+(k+1)},$$

which is equation 4.87 extended to $k+1$. The row $i$ block in the $k^{\text{th}}$ subdiagonal of $A^k$ will be a product of the form $L_iL_{i-1}\cdots L_{i-k}$. By Lemma 4.30, left multiplying

this by $U_{i-1}$ will produce a zero block.

Similarly, the matrix $L$ shifts the $i^{\text{th}}$ row of blocks in $A^k$ to row $i+1$ (one block down) and left multiplies this row by $L_{i+1}$. The $k^{\text{th}}$ block subdiagonal to the $k+1^{\text{th}}$ block subdiagonal, where

$$A^{k+1}_{i+1,i-k} = L_{i+1}B_{i,i-k} = L_{i+1}L_i \cdots L_{i-k+2}L_{i-k+1} = B_{i+1,i+2-(k+1)},$$

which is consistent with (4.88). The row $i$ block in the $k^{\text{th}}$ superdiagonal of $A^k$ will be a product of the form $U_iU_{i+1}\cdots U_{i+k-1}$. By Lemma 4.30, left multiplying this by $U_{i+1}$ will produce a zero block.

Finally, we note that the nonzero elements $A(1,2)$ and $A(1,3)$ will have a nonzero product with the top block of the $k^{\text{th}}$ block superdiagonal of $A^k$, resulting in nonzero elements in positions $(1, 2k+2)$ and $(1, 2k+3)$ of $A^{k+1}$. Similarly, the nonzero elements $A(2(S-1), 2(S-2))$, and $A(2(S-1), 2(S-2)+1)$ will have a nonzero product with the bottom block of the $k^{\text{th}}$ block subdiagonal of $A^k$, resulting in nonzero elements in positions $(2(S-1), 2(S-k-2))$ and $(2(S-1), 2(S-k-2)+1)$. The remaining four nonzero elements of $A^k$ do not contribute to any elements of $A^{k+1}$.

Having considered all parts of the product $A^{k+1}$, we see that the position and form of blocks follow (4.87) and (4.88), with remaining nonzero entries as described. We conclude that our claim about the form of $A^k$ is true, hence we have $A^{S-1} \equiv 0$ for $S \geq 4$. We thus have convergence in $S$ iterations. □

## 4.4 Optimized Schwarz Methods

While the deficiencies of classical Schwarz methods can be overcome through the use of optimal Schwarz methods, as observed in the previous section, the optimal conditions are often non-local, involving the evaluation of integrals. This makes the optimal conditions significantly more expensive to use in practice. However, it has been shown in [22] that the optimal transmission conditions of (4.64) can be approximated by developing a nonlinear Robin type transmission condition. While one would typically avoid introducing nonlinear transmission conditions if possible, as the differential equation (4.1) is already nonlinear, the extra cost of a single additional nonlinear equation is negligible during implementation.

On subdomain one the optimal transmission condition is

$$M(x_1^n(\alpha))\partial_\xi x_1^n(\alpha) - \frac{\int_{x_1^n(\alpha)}^1 M(\tilde{x})\,d\tilde{x}}{1-\alpha} = M(x_2^{n-1}(\alpha))\partial_\xi x_2^{n-1}(\alpha) - \frac{\int_{x_2^{n-1}(\alpha)}^1 M(\tilde{x})\,d\tilde{x}}{1-\alpha}.$$

Using

$$\int_x^1 M(\tilde{x})\,d\tilde{x} = C - \int_0^x M(\tilde{x})\,d\tilde{x}, \quad \text{where} \quad C = \int_0^1 M(\tilde{x})\,d\tilde{x},$$

the boundary condition can be rewritten as

$$M(x_1^n(\alpha))\partial_\xi x_1^n(\alpha) - \frac{C - \int_0^{x_1^n(\alpha)} M(\tilde{x})\,d\tilde{x}}{1-\alpha} = M(x_2^{n-1}(\alpha))\partial_\xi x_2^{n-1}(\alpha) - \frac{C - \int_0^{x_2^{n-1}(\alpha)} M(\tilde{x})\,d\tilde{x}}{1-\alpha}.$$

Canceling the constant terms and using the mean value theorem for integrals, we rewrite this as

$$M(x_1^n(\alpha))\partial_\xi x_1^n(\alpha) + \frac{M(x_1^*)}{1-\alpha}x_1^n(\alpha) = M(x_2^{n-1}(\alpha))\partial_\xi x_2^{n-1}(\alpha) + \frac{M(x_2^*)}{1-\alpha}x_2^{n-1}(\alpha),$$

or

$$M(x_1^n(\alpha))\partial_\xi x_1^n(\alpha) + \bar{p}_1^n x_1^n(\alpha) = M(x_2^{n-1}(\alpha))\partial_\xi x_2^{n-1}(\alpha) + \hat{p}_1^n x_2^{n-1}(\alpha),$$

where $\bar{p}_1^n, \hat{p}_1^n > 0$, as $M$ is strictly positive. Similarly, on subdomain two

$$M(x_2^n(\alpha))\partial_\xi x_2^n(\alpha) - \bar{p}_2^n x_2^n(\alpha) = M(x_1^{n-1}(\alpha))\partial_\xi x_1^{n-1}(\alpha) - \hat{p}_2^n x_1^{n-1}(\alpha),$$

where $\bar{p}_2^n, \hat{p}_2^n > 0$. This approximation in [22] led to the following optimized algorithm. Decompose $\Omega_c = [0,1]$ into subdomains $\Omega_1 = [0,\alpha]$ and $\Omega_2 = [\alpha,1]$, and apply the iteration: for $n = 1, 2, \ldots$

$$\begin{aligned}
(M(x_1^n)x_{1,\xi}^n)_\xi = 0, \quad \xi \in \Omega_1, \qquad & (M(x_2^n)x_{2,\xi}^n)_\xi = 0, \quad \xi \in \Omega_2, \\
x_1^n(0) = 0, \qquad & \widetilde{\mathcal{B}}_2(x_2^n(\alpha)) = \widetilde{\mathcal{B}}_2(x_1^{n-1}(\alpha)), \\
\widetilde{\mathcal{B}}_1(x_1^n(\alpha)) = \widetilde{\mathcal{B}}_1(x_2^{n-1}(\alpha)), \qquad & x_2^n(1) = 1,
\end{aligned} \tag{4.89}$$

where the transmission operators are

$$\widetilde{\mathcal{B}}_1(\cdot) \equiv M(\cdot)\partial_\xi(\cdot) + pI(\cdot), \quad \widetilde{\mathcal{B}}_2(\cdot) \equiv M(\cdot)\partial_\xi(\cdot) - pI(\cdot), \tag{4.90}$$

with $I(\cdot)$ the identity operator and $p$ a positive parameter.

The subdomain solutions from the optimized Schwarz algorithm (4.89) are represented implicitly in the following Lemma.

**Lemma 4.32.** *Under the assumptions of Lemmas 4.4 and 4.5, the subdomain solutions on $\Omega_1$ and $\Omega_2$ of (4.89) are given implicitly by*

$$\int_0^{x_1^n(\xi)} M(\tilde{x})\,d\tilde{x} = R_1(x_1^n(\alpha))\xi \quad \text{and} \quad \int_{x_2^n(\xi)}^1 M(\tilde{x})\,d\tilde{x} = R_2(x_2^n(\alpha))(1-\xi),$$

*where the operators $R_1$ and $R_2$ are given by*

$$R_1(x) = \frac{1}{\alpha}\int_0^x M(\tilde{x})\,d\tilde{x} \quad \text{and} \quad R_2(x) = \frac{1}{1-\alpha}\int_x^1 M(\tilde{x})\,d\tilde{x}. \tag{4.91}$$

*The transmission conditions force the operator values to satisfy the recurrence relations*

$$R_1(x_1^{n+1}(\alpha)) + px_1^{n+1}(\alpha) = R_2(x_2^n(\alpha)) + px_2^n(\alpha), \tag{4.92}$$

*and*

$$R_2(x_2^n(\alpha)) - px_2^n(\alpha) = R_1(x_1^{n-1}(\alpha)) - px_1^{n-1}(\alpha). \tag{4.93}$$

*Proof.* The implicit representation of the subdomain solutions follows from Lemmas 4.4 and 4.5. The recurrence relations (4.92) and (4.93) follow from the transmission conditions. □

As discussed in [22], equations (4.92–4.93) are an example of a Peaceman–Rachford type iteration [50], and the convergence of the optimized Schwarz iteration follows

from analysis of similar nonlinear Peaceman–Rachford iterations in [41, 49]. The following result was originally presented by Gander and Haynes in [22].

**Theorem 4.33.** *Under the assumptions of Lemma 4.1, the iteration (4.92–4.93) converges globally to the exact solution $x(\alpha)$ for all $p > 0$. Furthermore, we have the convergence estimate for the optimized Schwarz iteration*

$$||x - x_1^{2n+1}||_\infty \leq \frac{\hat{m}}{\check{m}} \cdot \frac{p + \frac{1}{\alpha}\hat{m}}{p + \frac{1}{\alpha}\check{m}} \rho_{robin}^n |x(\alpha) - x_1^0(\alpha)|,$$

$$||x - x_2^{2n+1}||_\infty \leq \frac{\hat{m}}{\check{m}} \cdot \frac{p + \frac{1}{1-\alpha}\hat{m}}{p + \frac{1}{1-\alpha}\check{m}} \rho_{robin}^n |x(\alpha) - x_2^0(\alpha)|,$$

*where an upper bound for the contraction factor is*

$$\rho_{robin} = \sqrt{\frac{p^2 + \frac{\hat{m}^2}{(1-\alpha)^2} - 2p\frac{\check{m}}{1-\alpha}}{p^2 + \frac{\hat{m}^2}{(1-\alpha)^2} + 2p\frac{\check{m}}{1-\alpha}}} \cdot \sqrt{\frac{p^2 + \frac{\hat{m}^2}{\alpha^2} - 2p\frac{\check{m}}{\alpha}}{p^2 + \frac{\hat{m}^2}{\alpha^2} + 2p\frac{\check{m}}{\alpha}}}.$$

*Proof.* We rewrite the iteration (4.92) and (4.93) as

$$(pI + R_1)x_1^{n+1}(\alpha) = (pI + R_2)x_2^n(\alpha),$$

$$(pI - R_2)x_2^n(\alpha) = (pI - R_1)x_1^{n-1}(\alpha).$$

The operators $R_1$ and $-R_2$ defined in (4.91) are continuous and uniformly monotonic increasing, as

$$R_1'(x) = \frac{1}{\alpha}M(x) \geq \frac{1}{\alpha}\check{m} > 0,$$

$$-R_2'(x) = \frac{1}{1-\alpha}M(x) \geq \frac{1}{1-\alpha}\check{m} > 0.$$

Furthermore, as $p > 0$, $pI - R_2$ and $pI + R_1$ are also continuous and uniformly monotonic, hence invertible. This implies that $x_2^n(\alpha)$ and $x_1^{n+1}(\alpha)$ are well-defined. Eliminating the former, we obtain the recursion formula

$$x_1^{n+1}(\alpha) = G x_1^{n-1}(\alpha),$$

where

$$G \equiv (pI + R_1)^{-1}(pI + R_2)(pI - R_2)^{-1}(pI - R_1).$$

We can express $G$ as the composition

$$G = (pI + R_1)^{-1} G_1 G_2 (pI + R_1),$$

where

$$G_1 = (pI + R_2)(pI - R_2)^{-1} \quad \text{and} \quad G_2 = (pI - R_1)(pI + R_1)^{-1}$$

are strict contractions for all $p > 0$, due to the uniform monotonicity and Lipschitz continuity of the operators $R_1$ and $-R_2$ [49]. Therefore, the iteration $z^n(\alpha) = G_1 G_2 z^{n-2}(\alpha)$, with $z^0(\alpha) = (pI + R_1)x_1^0(\alpha)$, is convergent. Furthermore, as $z^{2n}(\alpha) = (pI + R_1)x_1^{2n}(\alpha)$, $x_1^{2n}(\alpha)$ also converges globally to some limit $x_1^*(\alpha)$. We can similarly show the odd iterates $x_1^{2n+1}(\alpha)$ converge to the same limit. Likewise, the sequence $x_2^n(\alpha)$ converges to a limit point $x_2^*(\alpha)$. The points $x_1^*(\alpha)$ and $x_2^*(\alpha)$ must satisfy the limiting versions of (4.92) and (4.93): by adding these two equations we

find $x_1^*(\alpha) = x_2^*(\alpha) =: x^*(\alpha)$ and by taking their difference we find $x^*(\alpha)$ satisfies $R_1(x^*(\alpha)) = R_2(x^*(\alpha))$, that is,

$$\frac{1}{\alpha}\int_0^{x^*(\alpha)} M(\tilde{x})\,d\tilde{x} = \frac{1}{\alpha - 1}\left(\int_0^{x^*(\alpha)} M(\tilde{x})\,d\tilde{x} - C\right).$$

As in the classical and optimal cases, we can easily show $x^*(\alpha) = x(\alpha)$.

An upper bound for the contraction factor for $z^n(\alpha)$, $\rho_{\text{robin}}$, is found by computing the Lipschitz constant of the operator $G_1G_2$ as the product of the Lipschitz constants of $G_1$ and $G_2$ [49]. The convergence factor of $x_1^n(\alpha)$ is related to $\rho_{\text{robin}}$ by:

$$|x^*(\alpha) - x_1^{2n}(\alpha)| \leq L\tilde{L}\rho_{\text{robin}}^n |x^*(\alpha) - x_1^0(\alpha)|,$$

where $L$ and $\tilde{L}$ are the Lipschitz constants for $(pI + R_1)^{-1}$ and $(pI + R_1)$, respectively. It can be shown that $L = (p + \frac{1}{\alpha}\check{m})^{-1}$ and $\tilde{L} = p + \frac{1}{\alpha}\hat{m}$. Combining this fact with the estimate $|x_1^{2n}(\xi) - x(\xi)| \leq \frac{\hat{m}}{\check{m}}|x(\alpha) - x_1^{2n}(\alpha)|$, we have

$$|x_1^{2n}(\xi) - x(\xi)| \leq \frac{\hat{m}}{\check{m}} \cdot \frac{p + \frac{1}{\alpha}\hat{m}}{p + \frac{1}{\alpha}\check{m}} \rho_{\text{robin}}^n |x(\alpha) - x_1^0(\alpha)|.$$

The estimate on subdomain two follows similarly. □

As in the classical and optimal cases, we can easily modify the parallel iteration to obtain an alternating version. If $\Omega_1 = [0, \alpha]$ and $\Omega_2 = [\alpha, 1]$, the alternating optimized

Schwarz iteration is: for $n = 1, 2, \ldots$

$$(M(x_1^n)x_{1,\xi}^n)_\xi = 0, \quad \xi \in \Omega_1, \qquad (M(x_2^n)x_{2,\xi}^n)_\xi = 0, \quad \xi \in \Omega_2,$$

$$x_1^n(0) = 0, \qquad \widetilde{\mathcal{B}}_2(x_2^n(\alpha)) = \widetilde{\mathcal{B}}_2(x_1^n(\alpha)),$$

$$\widetilde{\mathcal{B}}_1(x_1^n(\alpha)) = \widetilde{\mathcal{B}}_1(x_2^{n-1}(\alpha)), \qquad x_2^n(1) = 1,$$

where the nonlinear transmission operators $\widetilde{\mathcal{B}}_i$, $i = 1, 2$ are the same as in (4.90). Again using of the operators (4.91), we see from the transmission conditions that the operator values must satisfy the recurrence relations

$$R_1(x_1^{n+1}(\alpha)) + px_1^{n+1}(\alpha) = R_2(x_2^n(\alpha)) + px_2^n(\alpha) \tag{4.94}$$

and

$$R_2(x_2^n(\alpha)) - px_2^n(\alpha) = R_1(x_1^n(\alpha)) - px_1^n(\alpha). \tag{4.95}$$

We state the alternating convergence result in the following Theorem.

**Theorem 4.34.** *Under the assumptions of Lemma 4.1, the iteration (4.94–4.95) converges globally to the exact solution $x(\alpha)$ for all $p > 0$. Moreover, we have the linear convergence estimate*

$$||x - x_1^n||_\infty \leq \frac{\hat{m}}{\check{m}} \cdot \frac{p + \frac{1}{\alpha}\hat{m}}{p + \frac{1}{\alpha}\check{m}} \rho_{robin}^n |x(\alpha) - x_1^0(\alpha)|,$$

$$||x - x_2^n||_\infty \leq \frac{\hat{m}}{\check{m}} \cdot \frac{p + \frac{1}{1-\alpha}\hat{m}}{p + \frac{1}{1-\alpha}\check{m}} \rho_{robin}^n |x(\alpha) - x_2^0(\alpha)|,$$

*where an estimate on the contraction factor is*

$$\rho_{robin} = \sqrt{\frac{p^2 + \frac{\hat{m}^2}{(1-\alpha)^2} - 2p\frac{\check{m}}{1-\alpha}}{p^2 + \frac{\hat{m}^2}{(1-\alpha)^2} + 2p\frac{\check{m}}{1-\alpha}}} \cdot \sqrt{\frac{p^2 + \frac{\hat{m}^2}{\alpha^2} - 2p\frac{\check{m}}{\alpha}}{p^2 + \frac{\hat{m}^2}{\alpha^2} + 2p\frac{\check{m}}{\alpha}}}.$$

The proof of the preceding Theorem is very similar to that of the parallel iteration, the main changes being in the iteration counters at various steps, hence it is omitted. We do note that we obtain the same contraction rate, $\rho_{\text{robin}}$, now for every iteration, instead of every second iteration.

## 4.5 Linearized DD Methods

As presented by Gander and Haynes in [22], it is possible to avoid solving the nonlinear system of the classical parallel Schwarz iteration (4.12) by replacing the arguments of the nonlinear function $M$, $x_1^n$ and $x_2^n$, with the corresponding functions from the previous iteration. By doing so, the nonlinear equations to be solved at each DD iteration are replaced with linear equations. While the linearized equations may require more DD iterations to reach the single domain solution, the cost of each iteration is significantly reduced.

Once again taking $\Omega_1 = [0, \beta]$, $\Omega_2 = [\alpha, 1]$, and $\alpha < \beta$, consider the iteration: for

$n = 1, 2, \ldots$

$$\begin{aligned} (M(x_1^{n-1})x_{1,\xi}^n)_\xi = 0, \quad \xi \in \Omega_1, \qquad & (M(x_2^{n-1})x_{2,\xi}^n)_\xi = 0, \quad \xi \in \Omega_2, \\ x_1^n(0) = 0, \qquad & x_2^n(\alpha) = x_1^{n-1}(\alpha), \\ x_1^n(\beta) = x_2^{n-1}(\beta), \qquad & x_2^n(1) = 1. \end{aligned} \tag{4.96}$$

An alternating Schwarz version of the algorithm (4.96) was previously presented in [23]. The resulting iteration is: for $n = 1, 2, \ldots$

$$\begin{aligned} (M(x_1^{n-1})x_{1,\xi}^n)_\xi = 0, \ \ \xi \in \Omega_1, \qquad & (M(x_2^{n-1})x_{2,\xi}^n)_\xi = 0, \ \ \xi \in \Omega_2, \\ x_1^n(0) = 0, \qquad & x_2^n(\alpha) = x_1^n(\alpha), \\ x_1^n(\beta) = x_2^{n-1}(\beta), \qquad & x_2^n(1) = 1. \end{aligned} \tag{4.97}$$

Both parallel and alternating iterations are numerically well behaved, but convergence to the single domain solution has yet to be established in either case. Comparisons of these linearized iterations to their nonlinear counterparts are given in Chapter 7, p. 167.

To extend upon the idea of linearized Schwarz iterations, we also consider the following iteration using optimized transmission conditions. Decomposing $\Omega_c$ into

$\Omega_1 = [0, \alpha]$ and $\Omega_2 = [\alpha, 1]$. For $n = 1, 2, \ldots$

$$\begin{gathered}
(M(x_1^{n-1})x_{1,\xi}^n)_\xi = 0, \quad \xi \in \Omega_1, \\
x_1^n(0) = 0, \qquad M(x_1^{n-1})x_{1,\xi}^n + px_1^n(\alpha) = M(x_2^{n-2})x_{2,\xi}^{n-1} + px_2^{n-1}(\alpha), \\
(M(x_2^{n-1})x_{2,\xi}^n)_\xi = 0, \quad \xi \in \Omega_2, \\
M(x_2^{n-1})x_{2,\xi}^n - px_2^n(\alpha) = M(x_1^{n-2})x_{1,\xi}^{n-1} - px_1^{n-1}(\alpha), \qquad x_2^n(1) = 1.
\end{gathered} \tag{4.98}$$

Similarly, we can easily formulate an alternating version of the iteration (4.98): for $n = 1, 2, \ldots$

$$\begin{gathered}
(M(x_1^{n-1})x_{1,\xi}^n)_\xi = 0, \quad \xi \in \Omega_1, \\
x_1^n(0) = 0, \qquad (M(x_1^{n-1})x_{1,\xi}^n)_\xi + px_1^n(\alpha) = (M(x_2^{n-2})x_{2,\xi}^{n-1})_\xi + px_2^{n-1}(\alpha), \\
(M(x_2^{n-1})x_{2,\xi}^n)_\xi = 0, \quad \xi \in \Omega_2, \\
(M(x_2^{n-1})x_{2,\xi}^n)_\xi - px_2^n(\alpha) = (M(x_1^{n-1})x_{1,\xi}^n)_\xi - px_1^n(\alpha), \qquad x_2^n(1) = 1.
\end{gathered} \tag{4.99}$$

As in the linearized classical Schwarz iterations, freezing the argument of $M$ at the previous iteration means the algorithm will require more DD iterations, but will require solving far fewer linear systems in the numerical implementation when compared to the nonlinear method.

We note that numerical results for all steady Schwarz iterations presented in this chapter can be found in Section 7.3 (p. 156).

# Chapter 5

# DD Methods for the Time Dependent Mesh Equation

As discussed in Chapter 2, a time dependent mesh transformation for a given time dependent function $u(x,t)$ may be found by solving any one of several nonlinear parabolic equations subject to appropriate initial and boundary conditions. In this chapter we focus on the solution of (MMPDE5) (p. 18) on $\Omega_c = [0,1]$, solving the initial boundary value problem

$$\frac{\partial x}{\partial t} = \frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M(x,t)\frac{\partial x}{\partial \xi}\right), \quad x(\xi,0) = x_0(\xi), \quad x(0,t) = 0, \quad x(1,t) = 1. \qquad (5.1)$$

In the following analysis we solve nonlinear BVPs of the form

$$x - q\frac{d}{d\xi}\left(M(x)\frac{dx}{d\xi}\right) = f, \quad x(\xi,0) = x_0(\xi), \quad x(a) = \gamma_a, \quad x(b) = \gamma_b, \tag{5.2}$$

where $q$ is a constant and $f = f(\xi)$ a source function, both given. Under the assumptions of Lemma 4.1 the well-posedness of (5.2) can be found in [27]. We consider the solution of (5.1) by first discretizing in time using an implicit method (backward Euler) and then solving the sequence of elliptic problems using Schwarz DD iterations. Given the solution $x^{k-1}(\xi)$ at $t_{k-1}$, the solution at time step $k$ satisfies

$$x^k - \frac{\Delta t}{\tau}(M(x^k, t_k)x^k_\xi)_\xi = x^{k-1}, \quad x^k(0,t_k) = 0, \quad x^k(1,t_k) = 1. \tag{5.3}$$

New results are stated in Theorems 5.4 – 5.6. Theorems 5.4 – 5.5 are previously unpublished and discuss the convergence of a time dependent multidomain parallel classical Schwarz iteration. Theorem 5.6, previously submitted in [30], covers the convergence of a time dependent alternating classical Schwarz iteration for any number of subdomains.

## 5.1 A Parallel Two Subdomain Method

We begin by solving (5.3) at each time step ($k = 1, 2, \ldots$) via the classical parallel Schwarz iteration on two subdomains $\Omega_1 = [0, \beta]$ and $\Omega_2 = [\alpha, 1]$, with $\alpha < \beta$. This was previously presented by Gander and Haynes in [22]. For $n = 1, 2, \ldots$

$$x_1^{k,n} - \frac{\Delta t}{\tau}(M(x_1^{k,n}, t_k)x_{1,\xi}^{k,n})_\xi = x^{k-1}, \qquad x_2^{k,n} - \frac{\Delta t}{\tau}(M(x_2^{k,n}, t_k)x_{2,\xi}^{k,n})_\xi = x^{k-1},$$

$$x_1^{k,n}(0, t_k) = 0, \qquad x_2^{k,n}(\alpha, t_k) = x_1^{k,n-1}(\alpha, t_k), \tag{5.4}$$

$$x_1^{k,n}(\beta, t_k) = x_2^{k,n-1}(\beta, t_k), \qquad x_2^{k,n}(1, t_k) = 1.$$

To show convergence of (5.4) we make use of the classical maximum principle. A contraction rate is obtained by constructing supersolutions and using the following comparison principle [27].

**Lemma 5.1.** *Suppose $Lu = au'' + bu' + cu$ is a linear, elliptic operator with $c \leq 0$ in a bounded domain $\Omega$. Suppose that in $\Omega$, $Lu \geq 0 (\leq 0)$ with $u \in C^2(\Omega) \cup C^0(\bar{\Omega})$. Then*

$$\sup_\Omega u \leq \sup_{\partial\Omega} max(u, 0) \qquad \left(\inf_\Omega u \geq \inf_{\partial\Omega} min(u, 0)\right).$$

We will also make use of the following lemma in establishing convergence.

**Lemma 5.2.** *For $0 < a < b$ we have*

$$\frac{\sinh(a)}{\sinh(b)} < \frac{a}{b}.$$

*Proof.* Consider

$$f(x) = \frac{\sinh(x)}{x},$$

a strictly increasing function for $x > 0$. Thus for $0 < a < b$ we know

$$\frac{\sinh(a)}{a} < \frac{\sinh(b)}{b} \quad \text{or} \quad \frac{\sinh(a)}{\sinh(b)} < \frac{a}{b},$$

the stated result. □

**Theorem 5.3.** *Under the assumptions of Lemma 4.1, the iteration* (5.4) *converges for any time step* $\Delta t > 0$ *and for any relaxation parameter* $\tau > 0$. *The convergence factor at the interfaces is bounded by*

$$\rho_{time} = \frac{\sinh(\sqrt{\theta}\alpha)}{\sinh(\sqrt{\theta}\beta)} \frac{\sinh(\sqrt{\theta}(1-\beta))}{\sinh(\sqrt{\theta}(1-\alpha))} < 1, \quad \textit{where} \quad \theta = \frac{\tau}{\Delta t} \frac{1}{\hat{m}}. \tag{5.5}$$

*Proof.* We define an error measure

$$e_{1,2}^{k,n}(\xi) = \int_{x_{1,2}^{k,n}(\xi)}^{x^k(\xi)} M(\tilde{x}, t_k)\, d\tilde{x},$$

and obtain the derivative

$$\frac{de_{1,2}^{k,n}}{d\xi} = M(x^k, t_k)\frac{dx^k}{d\xi} - M(x_{1,2}^{k,n}, t_k)\frac{dx_{1,2}^{k,n}}{d\xi}. \tag{5.6}$$

The mean value theorem for integrals implies

$$e_{1,2}^{k,n} = M(x_{1,2}^*, t_k)(x^k - x_{1,2}^{k,n}), \tag{5.7}$$

for some $x_{1,2}^*$ between $x^k$ and $x_{1,2}^{k,n}$.

Subtracting the equation for $x_1^{k,n}$ from the equation for $x^k$ we obtain

$$x^k - x_1^{k,n} - \frac{\Delta t}{\tau}\left(M(x^k, t_k)x_\xi^k - M(x_1^{k,n}, t_k)x_{1,\xi}^{k,n}\right)_\xi = 0,$$

and by using the relations (5.6) and (5.7), we see the error functions satisfy

$$\frac{d^2e_1^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{M(x_1^*,t_k)}e_1^{k,n} = 0, \qquad \frac{d^2e_2^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{M(x_2^*,t_k)}e_2^{k,n} = 0,$$

$$e_1^{k,n}(0,t_k) = 0, \qquad e_2^{k,n}(\alpha,t_k)) = e_1^{k,n-1}(\alpha,t_k),$$

$$e_1^{k,n}(\beta,t_k) = e_2^{k,n-1}(\beta,t_k), \qquad e_2^{k,n}(1,t_k) = 0.$$

The quantities $M$, $\tau$, and $\Delta t$ are strictly positive, hence the error equations satisfy a maximum principle [27], and the required contraction results.

We obtain a contraction estimate for (5.4) by constructing supersolutions for the subdomain errors. Assume $\tilde{e}_1^{k,n}$ solves the BVP

$$\frac{d^2\tilde{e}_1^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{\hat{m}}\tilde{e}_1^{k,n} = 0, \quad \tilde{e}_1^{k,n}(0) = 0, \quad \tilde{e}_1^{k,n}(\beta) = |e_2^{k,n-1}(\beta)|.$$

The function $\tilde{e}_1^{k,n}$ is found explicitly as

$$\tilde{e}_1^{k,n}(\xi) = |e_2^{k,n-1}(\beta)|\frac{\sinh(\sqrt{\theta}\xi)}{\sinh(\sqrt{\theta}\beta)}, \qquad \text{where} \qquad \theta = \frac{\tau}{\Delta t}\frac{1}{\hat{m}}.$$

We show $\tilde{e}_1^{k,n}$ is a supersolution for $e_1^{k,n}$. Defining $d_1^{k,n} = e_1^{k,n} - \tilde{e}_1^{k,n}$, from some simple calculations we see that $d_1^{k,n}$ satisfies

$$\frac{d^2d_1^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{M(x_1^*,t_k)}d_1^{k,n} = \frac{\tau}{\Delta t}\left(\frac{1}{M(x_1^*,t_k)} - \frac{1}{\hat{m}}\right)\tilde{e}_1^{k,n},$$

$$d_1^{k,n}(0) = 0, \qquad d_1^{k,n}(\beta) = e_2^{k,n-1}(\beta) - |e_2^{k,n-1}(\beta)|.$$

Since $\hat{m} \geq M(x,t_k)$, the right hand side of this differential equation is non-negative. The boundary value at $\xi = \beta$ is non-positive and the coefficient of $\tilde{e}_1^{k,n}$ in the differ-

ential equation is negative. Hence, Lemma 5.1 shows $d_1^{k,n} \leq 0$, that is, $e_1^{k,n} \leq \tilde{e}_1^{k,n}$ for all $\xi \in [0, \beta]$.

Similarly, $\tilde{d}_1^{k,n} = e_1^{k,n} + \tilde{e}_1^{k,n}$ satisfies

$$\frac{d^2\tilde{d}_1^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{M(x_1^*, t_k)}\tilde{d}_1^{k,n} = \frac{\tau}{\Delta t}\left(\frac{1}{\hat{m}} - \frac{1}{M(x_1^*, t_k)}\right)\tilde{e}_1^{k,n},$$

$$\tilde{d}_1^{k,n}(0) = 0, \qquad \tilde{d}_1^{k,n}(\beta) = e_2^{k,n-1}(\beta) + |e_2^{k,n-1}(\beta)|.$$

The right hand side of the equation is non-positive, the boundary conditions are non-negative, and the coefficient of $\tilde{d}_1^{k,n}$ is negative. By Lemma 5.1 we know $\tilde{d}_1^{k,n}(\xi) \geq 0$, that is, $e_1^{k,n}(\xi) \geq -\tilde{e}_1^{k,n}(\xi)$ for all $\xi \in [0, \beta]$. We have thus shown

$$|e_1^{k,n}(\xi)| \leq \tilde{e}_1^{k,n}(\xi) = |e_2^{k,n-1}(\beta)|\frac{\sinh(\sqrt{\theta}\xi)}{\sinh(\sqrt{\theta}\beta)}, \quad \text{for} \quad \xi \in \Omega_1.$$

One can similarly show

$$|e_2^{k,n}(\xi)| \leq |e_1^{k,n-1}(\alpha)|\frac{\sinh(\sqrt{\theta}(1-\xi))}{\sinh(\sqrt{\theta}(1-\alpha))}, \quad \text{for} \quad \xi \in \Omega_2,$$

and by combining these relations we have

$$|e_1^{k,n+1}(\alpha)| \leq |e_1^{k,n-1}(\alpha)|\frac{\sinh(\sqrt{\theta}\alpha)}{\sinh(\sqrt{\theta}\beta)}\frac{\sinh(\sqrt{\theta}(1-\beta))}{\sinh(\sqrt{\theta}(1-\alpha))}.$$

The contraction rate estimate $\rho_{\text{time}}$ stated in (5.5) is less than one for $\alpha < \beta$. This follows from Lemma 5.2, as

$$\frac{\sinh(\sqrt{\theta}\alpha)}{\sinh(\sqrt{\theta}\beta)}\frac{\sinh(\sqrt{\theta}(1-\beta))}{\sinh(\sqrt{\theta}(1-\alpha))} \leq \frac{\sqrt{\theta}\alpha}{\sqrt{\theta}\beta}\frac{\sqrt{\theta}(1-\beta)}{\sqrt{\theta}(1-\alpha)} = \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha}.$$

□

We note that the convergence of the iteration will improve as $\Delta t \to 0$, $\hat{m} \to 0$, or $\tau \to \infty$, since $\lim_{\theta \to \infty} \rho_{\text{time}} = 0$. Furthermore, the steady contraction rate is obtained in the opposite limit, since

$$\lim_{\theta \to 0} \rho_{\text{time}} = \frac{\alpha}{\beta} \frac{1-\beta}{1-\alpha}.$$

## 5.2 A Parallel Multidomain Method

We extend parallel Schwarz method to arbitrarily many subdomains in the following new, previously unpublished result. We decompose $\Omega_c$ into $\Omega_i = [\alpha_i, \beta_i]$ for $i = 1, \ldots, S$, with $\alpha_1 = 0$ and $\beta_S = 1$ and at each time step $k$ implement the following iteration. For $n = 1, 2, \ldots$

$$\begin{aligned} x_i^{k,n} - \frac{\Delta t}{\tau}(M(x_i^{k,n}, t_k) x_{i,\xi}^{k,n})_\xi &= x^{k-1}, \\ x_i^{k,n}(\alpha_i, t_k) &= x_{i-1}^{k,n-1}(\alpha_i, t_k), \\ x_i^{k,n}(\beta_i, t_k) &= x_{i+1}^{k,n-1}(\beta_i, t_k), \end{aligned} \tag{5.8}$$

for $i = 1, \ldots, S$, where we define $x_0^{k,n}(\alpha_1, t) \equiv 0$ and $x_{S+1}^{k,n}(\beta_S, t) \equiv 1$.

**Theorem 5.4.** *Under the assumptions of Lemma 4.1, the iteration (5.8) converges for any time step $\Delta t > 0$ and for any relaxation parameter $\tau > 0$.*

*Proof.* We define error measures

$$e_i^{k,n}(\xi) = \int_{x_i^{k,n}(\xi)}^{x^k(\xi)} M(\tilde{x}, t_k)\, d\tilde{x}, \quad \text{for} \quad i = 1, \ldots, S.$$

Upon differentiating, we find

$$\frac{de_i^{k,n}}{d\xi} = M(x^k, t_k)\frac{dx^k}{d\xi} - M(x_i^{k,n}, t_k)\frac{dx_i^{k,n}}{d\xi},$$

and from the mean value theorem for integrals, we obtain

$$e_i^{k,n} = M(x_i^*, t_k)(x^k - x_i^{k,n}),$$

for some $x_i^*$ between $x^k$ and $x_i^{k,n}$. Using these relations, the error functions are seen to satisfy

$$\frac{d^2e_i^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{M(x_i^*, t_k)}e_i^{k,n} = 0, \quad \xi \in (\alpha_i, \beta_i),$$

$$e_i^{k,n}(\alpha_i, t_k)) = e_{i-1}^{k,n-1}(\alpha_i, t_k),$$

$$e_i^{k,n}(\beta_i, t_k) = e_{i+1}^{k,n-1}(\beta_i, t_k).$$

As $M$, $\tau$, and $\Delta t$ are strictly positive, the error equation for each subdomain satisfies a maximum principle [27], proving the desired contraction. We construct a supersolution for the error on an arbitrary subdomain.

Let $\tilde{e}_i^{k,n}$ be the solution to

$$\frac{d^2\tilde{e}_i^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{\hat{m}}\tilde{e}_i^{k,n} = 0,$$

$$\tilde{e}_i^{k,n}(\alpha_i) = |e_{i-1}^{k,n-1}(\alpha_i)|, \quad \tilde{e}_i^{k,n}(\beta_i) = |e_{i+1}^{k,n-1}(\beta_i)|,$$

then $\tilde{e}_i^{k,n}$ may be obtained explicitly as

$$\tilde{e}_i^{k,n}(\xi) = |e_{i-1}^{k,n-1}(\alpha_i)|\frac{\sinh(\sqrt{\theta}(\beta_i - \xi))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))} + |e_{i+1}^{k,n-1}(\beta_i)|\frac{\sinh(\sqrt{\theta}(\xi - \alpha_i))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))},$$

where

$$\theta = \frac{\tau}{\Delta t}\frac{1}{\hat{m}}.$$

We now show $\tilde{e}_i^{k,n}$ is a supersolution for $e_i^{k,n}$. Define $d_i^{k,n} = e_i^{k,n} - \tilde{e}_i^{k,n}$, which satisfies

$$\frac{d^2 d_i^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{M(x_i^*, t_k)}e_i^{k,n} + \frac{\tau}{\Delta t}\frac{1}{\hat{m}}\tilde{e}_i^{k,n} = 0,$$

$$d_i^{k,n}(\alpha_i) = e_{i-1}^{k,n-1}(\alpha_i) - |e_{i-1}^{k,n-1}(\alpha_i)|,$$

$$d_i^{k,n}(\beta_i) = e_{i+1}^{k,n-1}(\beta_i) - |e_{i+1}^{k,n-1}(\beta_i)|.$$

Adding and subtracting

$$\frac{\tau}{\Delta t}\frac{1}{M(x_i^*, t_k)}\tilde{e}_i^{k,n},$$

we see $d_i^{k,n}$ satisfies

$$\frac{d^2 d_i^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{M(x_1^*, t_k)}d_i^{k,n} = \frac{\tau}{\Delta t}\left(\frac{1}{M(x_1^*, t_k)} - \frac{1}{\hat{m}}\right)\tilde{e}_i^{k,n},$$

$$d_i^{k,n}(\alpha_i) = e_{i-1}^{k,n-1}(\alpha_i) - |e_{i-1}^{k,n-1}(\alpha_i)|,$$

$$d_i^{k,n}(\beta_i) = e_{i+1}^{k,n-1}(\beta_i) - |e_{i+1}^{k,n-1}(\beta_i)|.$$

The right hand side of the differential equation is non-negative, both boundary values are non-positive, and the coefficient of $d_i^{k,n}$ in the differential equation is negative. Hence, Lemma 5.1 shows $d_i^{k,n} \leq 0$, that is, $e_i^{k,n} \leq \tilde{e}_i^{k,n}$ throughout $\Omega_i$.

Similarly, the quantity $\tilde{d}_i^{k,n} = e_i^{k,n} + \tilde{e}_i^{k,n}$ satisfies

$$\frac{d^2\tilde{d}_i^{k,n}}{d\xi^2} - \frac{\tau}{\Delta t}\frac{1}{M(x_i^*,t_k)}\tilde{d}_i^{k,n} = \frac{\tau}{\Delta t}\left(\frac{1}{\hat{m}} - \frac{1}{M(x_i^*,t_k)}\right)\tilde{e}_i^{k,n},$$

$$\tilde{d}_i^{k,n}(\alpha_i) = e_{i-1}^{k,n-1}(\alpha_i) + |e_{i-1}^{k,n-1}(\alpha_i)|,$$

$$\tilde{d}_i^{k,n}(\beta_i) = e_{i+1}^{k,n-1}(\beta_i) + |e_{i+1}^{k,n-1}(\beta_i)|,$$

and Lemma 5.1 guarantees that $\tilde{d}_i^{k,n}(\xi) \geq 0$, that is, $e_i^{k,n}(\xi) \geq -\tilde{e}_i^{k,n}(\xi)$ throughout $\Omega_i$. We have shown $|e_i^{k,n}(\xi)| \leq \tilde{e}_i^{k,n}(\xi)$, and thus, for $\xi \in [\alpha_i, \beta_i]$

$$|e_i^{k,n}(\xi)| \leq |e_{i-1}^{k,n-1}(\alpha_i)|\frac{\sinh(\sqrt{\theta}(\beta_i - \xi))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))} + |e_{i+1}^{k,n-1}(\beta_i)|\frac{\sinh(\sqrt{\theta}(\xi - \alpha_i))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))}. \tag{5.9}$$

Following the analysis in the steady case (see Theorem 4.13, p. 70), we introduce the quantities:

$$\begin{aligned} r_i &= \frac{\sinh(\sqrt{\theta}(\beta_{i-1} - \alpha_i))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))}, \quad p_i = \frac{\sinh(\sqrt{\theta}(\beta_i - \beta_{i-1}))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))}, \\ q_i &= \frac{\sinh(\sqrt{\theta}(\alpha_{i+1} - \alpha_i))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))}, \quad s_i = \frac{\sinh(\sqrt{\theta}(\beta_i - \alpha_{i+1}))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))}. \end{aligned} \tag{5.10}$$

The error at the interface $\xi = \beta_{i-1}$, for $i = 2, \ldots, N$, satisfies

$$\begin{aligned} |e_i^{k,n+2}(\beta_{i-1})| \leq\ & r_i r_{i+1}|e_{k,i+2}^n(\beta_{i+1})| + r_i p_{i+1}|e_i^{k,n}(\alpha_{i+1})| \\ & + p_i q_{i-1}|e_i^{k,n}(\beta_{i-1})| + p_i s_{i-1}|e_{i-2}^{k,n}(\alpha_{i-1})|, \end{aligned} \tag{5.11}$$

while at $\xi = \alpha_{i+1}$, for $i = 1, \ldots, N-1$, we have

$$\begin{aligned} |e_i^{k,n+2}(\alpha_{i+1})| \leq\ & q_i r_{i+1}|e_{i+2}^{k,n}(\beta_{i+1})| + q_i p_{i+1}|e_i^{k,n}(\alpha_{i+1})| \\ & + s_i q_{i-1}|e_i^{k,n}(\beta_{i-1})| + s_i s_{i-1}|e_{i-2}^{k,n}(\alpha_{i-1})|. \end{aligned} \tag{5.12}$$

Inequality (5.11) is obtained by evaluating (5.9) to find $e_i^{k,n+2}(\beta_{i-1})$, then using (5.9) twice more to rewrite $e_{i+1}^{k,n+1}(\beta_i)$ in terms of $e_{i+2}^{k,n}$ and $e_i^{k,n}$, and $e_{i-1}^{k,n+1}(\alpha_i)$ in terms of $e_i^{k,n}$ and $e_{i-2}^{k,n}$. Applying absolute values, using the triangle inequality and noting the quantities of (5.10) are non-negative gives (5.11). Inequality (5.12) follows similarly.

In the case of an even number of subdomains (the odd case can be handled in a similar manner), the relations of (5.9) may be written as $\boldsymbol{e}^{k,n+2} \leq M_{\boldsymbol{e}}\boldsymbol{e}^{k,n}$ and $\hat{\boldsymbol{e}}^{k,n+2} \leq M_{\hat{\boldsymbol{e}}}\hat{\boldsymbol{e}}^{k,n}$, where

$$\boldsymbol{e}^{k,n} = (|e_1^{k,n}(\alpha_2)|, |e_3^{k,n}(\beta_2)|, |e_3^{k,n}(\alpha_4)|, \ldots, |e_{S-1}^{k,n}(\beta_{S-2})|, |e_{S-1}^{k,n}(\alpha_S)|)^T,$$

$$\hat{\boldsymbol{e}}^{k,n} = (|e_2^{k,n}(\beta_1)|, |e_2^{k,n}(\alpha_3)|, |e_4^{k,n}(\beta_3)|, \ldots, |e_{S-2}^{k,n}(\alpha_{S-1})|, |e_S^{k,n}(\beta_{S-1})|)^T,$$

and the $(S-1) \times (S-1)$ matrices are given as

$$M_{\boldsymbol{e}} = \begin{pmatrix}
p_2q_1 & r_2q_1 & & & & & & \\
p_3s_2 & p_3q_2 & p_4r_3 & r_4r_3 & & & & \\
s_3s_2 & s_3q_2 & p_4q_3 & r_4q_3 & & & & \\
 & & p_5s_4 & p_5q_4 & p_6r_5 & r_6r_5 & & \\
 & & s_5s_4 & s_5q_4 & p_6q_5 & r_6q_5 & & \\
 & & & & \ddots & \ddots & & \\
 & & & & & p_{S-1}s_{S-2} & p_{S-1}q_{S-2} & p_Sr_{S-1} \\
 & & & & & s_{S-1}s_{S-2} & s_{S-1}q_{S-2} & p_Sq_{S-1}
\end{pmatrix},$$

$$M_{\hat{\boldsymbol{e}}} = \begin{pmatrix}
p_2q_1 & q_3r_2 & r_3r_2 & & & & & \\
s_2q_1 & p_3q_2 & r_3q_2 & & & & & \\
 & p_4s_3 & p_4q_3 & p_5r_4 & r_5r_4 & & & \\
 & s_4s_3 & s_4q_3 & p_5q_4 & r_5q_4 & & & \\
 & & & \ddots & \ddots & & & \\
 & & & p_{S-2}s_{S-3} & p_{S-2}q_{S-3} & p_{S-1}r_{S-2} & r_{S-1}r_{S-2} \\
 & & & s_{S-2}s_{S-3} & s_{S-2}q_{S-3} & p_{S-1}q_{S-2} & r_{S-1}q_{S-2} \\
 & & & & & p_Ss_{S-1} & p_Sq_{S-1}
\end{pmatrix}.$$

To demonstrate convergence, we show $\rho(M_{\boldsymbol{e}}) < 1$ and $\rho(M_{\hat{\boldsymbol{e}}}) < 1$. Lemma 5.2 implies that the quantities in (5.10) are strictly less than their steady counterparts in (4.39). It follows that each row sum for $M_{\boldsymbol{e}}$ and $M_{\hat{\boldsymbol{e}}}$ will be strictly less than the corresponding row sum in the steady case. Recalling that each matrix in the proof of Theorem 4.13 satisfies $\|M\|_\infty = 1$, we conclude $\|M_{\boldsymbol{e}}\|_\infty < 1$ and $\|M_{\hat{\boldsymbol{e}}}\|_\infty < 1$, from which $\rho(M_{\boldsymbol{e}}) < 1$ and $\rho(M_{\hat{\boldsymbol{e}}}) < 1$ follow. □

This general partitioning does not allow an explicit bound for the rate of convergence. If we make the assumption that subdomains are of equal length and each pair of adjacent subdomains have equal amounts of overlap, we have the following explicit error estimate on $S$ subdomains.

**Theorem 5.5.** *The Schwarz iteration* (5.8) *on $S$ subdomains with a common overlap ratio $r \in (0, 0.5]$ converges in the infinity norm and the iterates satisfy*

$$\begin{aligned}
\max_{1\le 2i\le S} \|x_{2i}^{k,2n+1}(\xi) - x^k(\xi)\|_\infty &\le \left((p+r)^2 - 4pr\sin^2\frac{\pi}{2(S+1)}\right)^n \frac{1}{\check{m}}\|e^{k,0}\|_2, \\
\max_{1\le 2i+1\le S} \|x_{2i+1}^{k,2n+1}(\xi) - x^k(\xi)\|_\infty &\le \left((p+r)^2 - 4pr\sin^2\frac{\pi}{2(S+1)}\right)^n \frac{1}{\check{m}}\|\hat{e}^{k,0}\|_2,
\end{aligned}$$

*where*

$$p = \frac{\sinh(\sqrt{\theta}(\beta_i - \beta_{i-1}))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))} \quad \textit{and} \quad r = \frac{\sinh(\sqrt{\theta}(\beta_{i-1} - \alpha_i))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))}.$$

The proof is essentially the same as the proof of Theorem 4.14, hence is omitted.

## 5.3 An Alternating Multidomain Method

As in the steady case, we can easily modify the proposed parallel iterations to obtain corresponding alternating Schwarz iterations. To illustrate this, we implement the alternating approach previously discussed in Section 4.2.3 for the time dependent case. This iteration has been described previously in [30]. We also note that the alternating methods which group subdomain problems to allow parallel implementation, such as in Sections 4.2.4 and 4.2.5, can be obtained similarly. We decompose $\Omega_c = [0,1]$ into $S$ subdomains $\Omega_i = [\alpha_i, \beta_i]$ for $i = 1, \ldots, S$, where $\alpha_{i+1} < \beta_i$ for $i = 1, \ldots, S-1$ and $\beta_i < \alpha_{i+2}$ for $i = 1, \ldots, S-2$. We solve the alternating classical Schwarz iteration at each time step $k$: for $n = 1, 2, \ldots$

$$\begin{aligned} x_i^{k,n} - \frac{\Delta t}{\tau}(M(x_i^{k,n}, t_k)x_{i,\xi}^{k,n})_\xi &= x^{k-1}, \quad \xi \in \Omega_i, \\ x_i^{k,n}(\alpha_i, t_k) &= x_{i-1}^{k,n}(\alpha_i, t_k), \\ x_i^{k,n}(\beta_i, t_k) &= x_{i+1}^{k,n-1}(\beta_i, t_k), \end{aligned} \tag{5.13}$$

for $i = 1, \ldots, S$, where $x_0^{k,n}(\alpha_1, t) \equiv 0$ and $x_{S+1}^{k,n}(\beta_S, t) \equiv 1$. For the initial solution $x^0$ one can take a uniformly distributed mesh or a mesh which equidistributes the initial physical solution.

**Theorem 5.6.** *Under the assumptions of Lemma 4.1 and the restrictions on the partitioning of $\Omega_c$ described above, the iteration* (5.13) *converges for any time step*

$\Delta t > 0$ *and for any relaxation parameter value* $\tau > 0$.

*In the case of two subdomains we have the linear convergence estimates*

$$||x^k - x_1^{k,n+1}||_\infty \leq \rho_{time}^n \frac{\hat{m}}{\check{m}} |x^k(\alpha) - x_1^{k,0}(\alpha)|,$$

$$||x^k - x_2^{k,n+1}||_\infty \leq \rho_{time}^n \frac{\hat{m}}{\check{m}} |x^k(\beta) - x_2^{k,0}(\beta)|,$$

*where the contraction rate is bounded by*

$$\rho_{time} = \frac{\sinh(\sqrt{\theta}\alpha)}{\sinh(\sqrt{\theta}\beta)} \frac{\sinh(\sqrt{\theta}(1-\beta))}{\sinh(\sqrt{\theta}(1-\alpha))} < 1, \quad \theta = \frac{\tau}{\Delta t} \frac{1}{\hat{m}}.$$

*For* $S \geq 3$ *subdomains we have the estimate*

$$\max_{1 \leq i \leq S} \|x_i^{k,n}(\xi) - x^k(\xi)\|_\infty \leq \rho_{time}^n \frac{1}{\check{m}} \|\boldsymbol{e}^{k,0}\|_2,$$

*where the contraction rate is bounded by*

$$\rho_{time} \leq r + \frac{p^2(1 - r^{S-2})}{1 - r} < 1,$$

*with* $p$ *and* $r$ *as defined in* (5.10).

*Proof.* Introducing error measures and using the method of supersolutions as in the parallel case, we obtain the inequalities

$$|e_i^{k,n}(\xi)| \leq = |e_{i-1}^{k,n}(\alpha_i)| \frac{\sinh(\sqrt{\theta}(\beta_i - \xi))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i)} + |e_{i+1}^{k,n-1}(\beta_i)| \frac{\sinh(\sqrt{\theta}(\xi - \alpha_i))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i)},$$

for $\xi \in [\alpha_i, \beta_i]$, $i = 1, \ldots, S$, and

$$\theta = \frac{\tau}{\Delta t} \frac{1}{\hat{m}}.$$

Making use of the notation (5.10) and proceeding as in the steady case (see proof of Lemma 4.18), we find that the error at interfaces $\xi = \beta_{i-1}$, for $i = 2, \ldots, S$, satisfies

$$|e_i^{k,n}(\beta_{i-1})| \leq p_i \sum_{j=2}^{i} \left( q_{j-1} \prod_{\ell=j}^{i-1} s_\ell |e_j^{k,n-1}(\beta_{j-1})| \right) + r_i |e_{i+1}^{k,n-1}(\beta_i)|,$$

while at $\xi = \alpha_{i+1}$, $i = 1, \ldots, S-1$ we have

$$|e_i^{k,n}(\alpha_{i+1})| \leq \sum_{j=2}^{i+1} \left( q_{j-1} \prod_{\ell=j}^{i} s_\ell |e_j^{k,n-1}(\beta_{j-1})| \right),$$

where we define $\prod_{\ell=i}^{i-1} s_\ell = 1$. As in the steady case, we can restrict our attention to $\beta$ interfaces. We write these inequalities in matrix form, $\boldsymbol{e}^{k,n+1} \leq M_{\boldsymbol{e}} \boldsymbol{e}^{k,n}$, where

$$\boldsymbol{e}^{k,n} = (|e_2^{k,n}(\beta_1)|, |e_3^{k,n}(\beta_2)|, \ldots, |e_S^{k,n}(\beta_{S-1})|)^T,$$

and the $(S-1) \times (S-1)$ matrix is given as

$$M_{\boldsymbol{e}} = \begin{bmatrix} p_2 q_1 & r_2 & & & & \\ p_3 s_2 q_1 & p_3 q_2 & r_3 & & & \\ p_4 s_3 s_2 q_1 & p_4 s_3 q_2 & p_4 q_3 & r_4 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ & & & & p_{S-1} q_{S-2} & r_{S-1} \\ p_S s_{S-1} \cdots s_2 q_1 & \cdots & & & p_S s_{S-1} q_{S-2} & p_S q_{S-1} \end{bmatrix}.$$

Convergence follows immediately from Lemma 5.2 and the technique of proof in the steady case.

For the two subdomain iteration,

$$x_1^{k,n} - \frac{\Delta t}{\tau}(M(x_1^{k,n}, t_k)x_{1,\xi}^{k,n})_\xi = x^{k-1}, \qquad x_2^{k,n} - \frac{\Delta t}{\tau}(M(x_2^{k,n}, t_k)x_{2,\xi}^{k,n})_\xi = x^{k-1},$$

$$x_1^{k,n}(0, t_k) = 0, \qquad x_2^{k,n}(\alpha, t_k) = x_1^{k,n}(\alpha, t_k),$$

$$x_1^{k,n}(\beta, t_k) = x_2^{k,n-1}(\beta, t_k), \qquad x_2^{k,n}(1, t_k) = 1.$$

the matrix $M_e$ is reduced to the single scalar value $p_2 q_1$, which is our contraction rate, $\rho_{\text{time}}$. Substituting the expressions for $p_2$ and $q_1$, we find

$$\rho_{\text{time}} = \frac{\sinh(\sqrt{\theta}\alpha)}{\sinh(\sqrt{\theta}\beta)} \frac{\sinh(\sqrt{\theta}(1-\beta))}{\sinh(\sqrt{\theta}(1-\alpha))} < 1,$$

the same contraction rate obtained in the two subdomain parallel iteration.

For $S \geq 3$ subdomains, if we make the simplifying assumption that all subdomains are of equal size and each pair of adjacent subdomains have equal amounts of overlap, then we have

$$r = s = \frac{\sinh(\sqrt{\theta}(\beta_{i-1} - \alpha_i))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))} \quad \text{and} \quad p = q = \frac{\sinh(\sqrt{\theta}(\beta_i - \beta_{i-1}))}{\sinh(\sqrt{\theta}(\beta_i - \alpha_i))},$$

and corresponding matrix

$$M_e = \begin{bmatrix} p^2 & r & & & & \\ p^2 r & p^2 & r & & & \\ p^2 r^2 & p^2 r & p^2 & r & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ & & & & p^2 & r \\ p^2 r^{S-2} & \cdots & & & p^2 r & p^2 \end{bmatrix}.$$

As in the steady alternating case, we find

$$\|M_e\|_\infty = r + \frac{p^2(1 - r^{S-2})}{1 - r} < 1,$$

which is an upper bound on the contraction rate. The $L^\infty$ error bound expression follows as in the steady case, see the proof of Theorem 4.15 for details. □

Numerical results for the time dependent Schwarz iterations can be found in Section 7.3 (p. 170).

# Chapter 6

# DD Methods for 2D Mesh Equations

As an initial attempt at solving higher dimensional mesh equations via domain decomposition, we consider the system of equations for two dimensional mesh adaptation [39], previously given in Chapter 2, (p. 29),

$$\begin{aligned}\left[\begin{pmatrix}\frac{\partial x}{\partial \xi}\\ \frac{\partial y}{\partial \xi}\end{pmatrix}^T M \begin{pmatrix}\frac{\partial x}{\partial \xi}\\ \frac{\partial y}{\partial \xi}\end{pmatrix}\right]^{1/2} &= c_1(\eta),\\ \left[\begin{pmatrix}\frac{\partial x}{\partial \eta}\\ \frac{\partial y}{\partial \eta}\end{pmatrix}^T M \begin{pmatrix}\frac{\partial x}{\partial \eta}\\ \frac{\partial y}{\partial \eta}\end{pmatrix}\right]^{1/2} &= c_2(\xi).\end{aligned} \tag{6.1}$$

As these are functions of two variables, $\xi$ and $\eta$, we can decompose the computational domain $\Omega_c = [0,1] \times [0,1]$ in either the $\xi$ direction, the $\eta$ direction, or in both direc-

tions simultaneously. Decomposing in a single or in both directions result in "strip" or "block" configurations of subdomains, respectively. For simplicity, we consider DD applied in the $\xi$ direction only; that is, we decompose the $\xi$ interval $[0,1]$ into subintervals $[\alpha_\xi^i, \beta_\xi^i]$, $i = 1, \ldots, S$, where $\alpha_\xi^1 = 0$, $\beta_\xi^S = 1$, and assume the subintervals satisfy the overlap conditions:

$$\alpha_\xi^i < \alpha_\xi^{i+1} < \beta_\xi^i < \beta_\xi^{i+1}.$$

The resulting decomposition of $\Omega_c$ has $S$ subdomains, denoted by $\Omega_i = [\alpha_\xi^i, \beta_\xi^i] \times [0,1]$ for $i = 1, \ldots, S$. The Dirichlet boundary conditions (2.18) and the 1D EP (4.1) are used along the ends of each strip, transmission conditions are specified along the newly created interfaces.

## 6.1 A Classical Schwarz Method

We begin by considering the 2D adaptive mesh system (6.1) for the two subdomain case, illustrated in Figure 6.1. Decompose $\Omega_c$ into subdomains $\Omega_1$ and $\Omega_2$ and let $x_i^n$ denote the solution on $\Omega_i$. The corresponding parallel classical Schwarz iteration

follows: for $n = 1, 2, \ldots$

$$\left[ \begin{pmatrix} \frac{\partial x_i^n}{\partial \xi} \\ \frac{\partial y_i^n}{\partial \xi} \end{pmatrix}^T M(x_i^n, y_i^n) \begin{pmatrix} \frac{\partial x_i^n}{\partial \xi} \\ \frac{\partial y_i^n}{\partial \xi} \end{pmatrix} \right]^{1/2} = c_{1,i}^n(\eta), \tag{6.2}$$

$$\left[ \begin{pmatrix} \frac{\partial x_i^n}{\partial \eta} \\ \frac{\partial y_i^n}{\partial \eta} \end{pmatrix}^T M(x_i^n, y_i^n) \begin{pmatrix} \frac{\partial x_i^n}{\partial \eta} \\ \frac{\partial y_i^n}{\partial \eta} \end{pmatrix} \right]^{1/2} = c_{2,i}^n(\xi), \tag{6.3}$$

for $i = 1, 2$ and $\boldsymbol{\xi} \in \Omega_i$, with transmission conditions

$$x_1^n(\beta, \eta) = x_2^{n-1}(\beta, \eta), \quad y_1^n(\beta, \eta) = y_2^{n-1}(\beta, \eta),$$

$$x_2^n(\alpha, \eta) = x_1^{n-1}(\alpha, \eta), \quad y_2^n(\alpha, \eta) = y_1^{n-1}(\alpha, \eta).$$

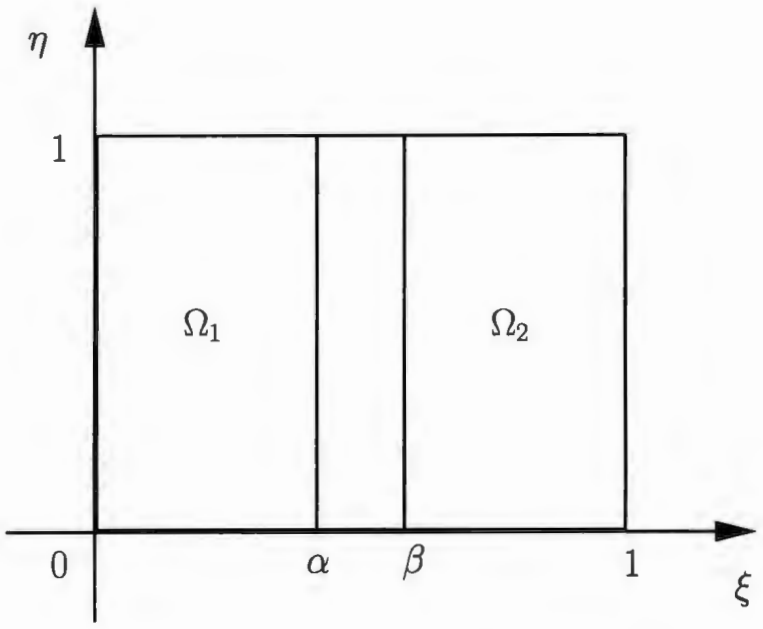


Figure 6.1: Decomposing $\Omega_c$ into two subdomains.

## 6.2 Optimized Schwarz Methods

As in the 1D case, the classical Schwarz iteration converges slowly. As a way to remedy this, we propose the use of higher order, Robin type, transmission conditions along the artificial interfaces. As before, we decompose $\Omega_c = [0,1] \times [0,1]$ into subdomains $\Omega_1 = [0,\beta] \times [0,1]$ and $\Omega_2 = [\alpha,1] \times [0,1]$, where $\alpha \leq \beta$. We consider several different possible transmission conditions, each an attempt to extend the successful 1D optimized conditions to two dimensions.

### 6.2.1 Case One

The first possibility are simple linear Robin conditions, using the derivative normal to the artificial boundaries. Defining for any differentiable function $x(\xi,\eta)$ the operators

$$B_1(x) = x_\xi + px \quad \text{and} \quad B_2(x) = x_\xi - px,$$

these Robin conditions are expressed as

$$\begin{aligned} B_1(x_1^n(\beta,\eta)) &= B_1(x_2^{n-1}(\beta,\eta)), \quad B_1(y_1^n(\beta,\eta)) = B_1(y_2^{n-1}(\beta,\eta)), \\ B_2(x_2^n(\alpha,\eta)) &= B_2(x_1^{n-1}(\alpha,\eta)), \quad B_2(y_2^n(\alpha,\eta)) = B_2(y_1^{n-1}(\alpha,\eta)). \end{aligned} \tag{6.4}$$

### 6.2.2 Case Two

The second possible set of transmission conditions are a type of nonlinear Robin condition, similar to those used in the 1D optimized Schwarz iteration (4.89). We

replace the $x$ equations of (6.4) by

$$B_3(x_1^n(\beta,\eta), y_1^n(\beta,\eta)) = B_3(x_2^{n-1}(\beta,\eta) y_2^{n-1}(\beta,\eta)),$$

$$B_4(x_2^n(\alpha,\eta), y_2^n(\alpha,\eta)) = B_4(x_1^{n-1}(\alpha,\eta), y_1^{n-1}(\alpha,\eta)),$$

defining for any differentiable functions $x(\xi,\eta)$ and $y(\xi,\eta)$ the operators

$$B_3(x,y) = S_1(x,y) + px \quad \text{and} \quad B_4(x,y) = S_1(x,y) - px,$$

where

$$S_1(x,y) = \sqrt{\begin{pmatrix} x_\xi \\ y_\xi \end{pmatrix}^T M \begin{pmatrix} x_\xi \\ y_\xi \end{pmatrix}}.$$

The monitor matrix used in (6.1) is

$$M = \frac{a^2 w \cdot w^T}{1 + b w^T \cdot w} + I, \quad \text{where} \quad w = \frac{1}{x_\xi y_\eta - x_\eta y_\xi} \left[ u_\xi y_\eta - u_\eta y_\xi, -u_\xi x_\eta + u_\eta x_\xi \right]^T.$$

The mesh PDE (6.2) indicates the nonlinear term $S_1$ is constant across the $\xi = \alpha$ and $\xi = \beta$ interfaces. Furthermore, as the system of equations resulting from (6.2-6.3) are already nonlinear, the nonlinear transmission conditions will not have a large impact on the cost of solving the system.

### 6.2.3 Case Three

The third set of transmission conditions has two variations, though both are similar in form and appearance. Modifying the previous iteration, we now use the operators

$$B_5(x,y) = S_2(x,y) + px \quad \text{and} \quad B_6(x,y) = S_2(x,y) - px,$$

$$S_2(x,y) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T M \begin{pmatrix} x_\xi \\ y_\xi \end{pmatrix},$$

where we may take

$$M = \frac{a^2 w \cdot w^T}{1 + bw^T \cdot w} + I \quad \text{or} \quad M = \sqrt{\frac{a^2 w \cdot w^T}{1 + bw^T \cdot w} + I},$$

where the matrix square root is well defined, as the argument is a symmetric positive definite matrix. To differentiate between the two possible choices of monitor function $M$ in subsequent discussion, we denote the former as Case Three a) and the latter by Case Three b). We now replace the $x$ equations of (6.4) by:

$$B_5(x_1^n(\beta,\eta), y_1^n(\beta,\eta)) = B_5(x_2^{n-1}(\beta,\eta), y_2^{n-1}(\beta,\eta)),$$

$$B_6(x_2^n(\alpha,\eta), y_2^n(\alpha,\eta)) = B_6(x_1^{n-1}(\alpha,\eta), y_1^{n-1}(\alpha,\eta)).$$

### 6.2.4 Case Four

The final proposed set of boundary conditions are described by the matrix equations

$$M\boldsymbol{x}_{1,\xi}^n + p\boldsymbol{x}_1^n = M\boldsymbol{x}_{2,\xi}^{n-1} + p\boldsymbol{x}_2^{n-1},$$

$$M\boldsymbol{x}_{2,\xi}^n - p\boldsymbol{x}_2^n = M\boldsymbol{x}_{1,\xi}^{n-1} - p\boldsymbol{x}_1^{n-1}.$$

This allows us to consider the entirety of the matrix $M$, rather than leaving out half as in iterations three and four. We also have consistent transmission conditions for both $x$ and $y$ variables, rather than using linear Robin conditions for $y$ and nonlinear versions for $x$.

Numerical results for the parallel classical Schwarz method, as well as all proposed optimized Schwarz methods, can be found in Section 7.4 (p. 172).

# Chapter 7

# Numerical Implementation and Results

In this chapter we describe how the various Schwarz DD iterations have been implemented for the 1D and 2D mesh equations. We then present numerical results which compare the various algorithms proposed and illustrate the types of meshes obtained.

## 7.1 Subdomain Specification

We decompose the domain $\Omega_c = [0, 1]$ into $S \geq 2$ subdomains,

$$\Omega_k = [\alpha_k, \beta_k], \quad \text{for} \quad k = 1, \ldots, S,$$

such that $\alpha_1 = 0$ and $\beta_S = 1$ — the endpoints of the original interval. To simplify the task of choosing appropriate $\alpha_k$ and $\beta_k$ for each of the $S$ intervals, we assume that each subdomain will have an equal number of mesh points. With this assumption, given a minimum number of mesh points to be used, $N_{\min}$, the number of subdomains to be used, $S \geq 2$, and the number of mesh points overlapping subdomains should share, $K \geq 1$; the number of points within a subdomain, $P$, as well as the location of its endpoints, can be determined. We determine $N$, the number of mesh points throughout $\Omega_c$, as follows. If there are $P$ points in a subdomain, then

$$\underbrace{P + (P - K) + (P - K) + \cdots + (P - K)}_{S \text{ Subdomains}} = N,$$

as each subdomain (excluding the first) shares exactly $K$ points with the previous subdomain. Thus

$$S(P - K) + K = N,$$

and by rearranging to solve for $P$, we find

$$P = \frac{N - K}{S} + K.$$

We require $P \in \mathbb{Z}$, which gives the restriction

$$\frac{N - K}{S} \in \mathbb{Z}.$$

To ensure this holds, we determine $N$ by

$$N = \left\lceil \frac{N_{\min} - K}{S} \right\rceil \cdot S + K.$$

This determines a correspondence between the subdomain endpoints: $[\alpha_i, \beta_i]$ for $i = 1, \ldots, S$, and the mesh points of $\Omega_c$, $\xi_j = \frac{j-1}{N-1}$ for $j = 1, \ldots, N$

$$\alpha_i = \xi_{(i-1)\cdot(P-K)+1},$$

$$\beta_i = \xi_{i\cdot(P-K)+K}.$$

Finally, to update the transmission conditions for $\Omega_k$, $k = 2, \ldots, S-1$, we must know the index of the point in $\Omega_{k-1}$ corresponding to $\alpha_k$, and the index of the point in $\Omega_{k+1}$ corresponding to $\beta_k$. Letting $X_k^n(i)$ denote the $i^{\text{th}}$ point of the numerical solution on the $k^{\text{th}}$ subdomain for the $n^{\text{th}}$ iteration, the desired relationships are

$$X_k^n(1) = X_{k-1}^{n-1}(P-K+1) \quad \text{and} \quad X_k^n(P) = X_{k+1}^{n-1}(K).$$

## 7.2 Implementing the Transmission Conditions

When implementing optimized or optimal Schwarz methods numerically, care needs to be taken when handling the transmission conditions due to the derivative approximations required. In this section we discuss how to handle both optimized and optimal cases.

### 7.2.1 Optimized Transmission Conditions

When handling the optimized Schwarz iteration (4.89), there are two steps to the numerical enforcement of the transmission conditions involving the operators specified in (4.90). We first must handle the extraction problem — calculating the right hand sides of these conditions, which we denote by $B_R$ and $B_L$;

$$B_R = M(x_2^{n-1}(\beta))\frac{dx_2^{n-1}(\beta)}{d\xi} + p\, x_2^{n-1}(\beta),$$
$$B_L = M(x_1^{n-1}(\alpha))\frac{dx_1^{n-1}(\alpha)}{d\xi} - p\, x_1^{n-1}(\alpha).$$

This involves numerically approximating solution derivatives from the previous iteration. If there are multiple points of overlap, we calculate $B_R$ and $B_L$ by central differences. However, if we use non-overlapping subdomains, there is a challenge in enforcing these conditions numerically with sufficient accuracy. We proceed recursively. Letting subscripts denote the subdomain and superscripts denote the iteration, and suppressing the argument (as $\alpha = \beta$ in this case), we initially have

$$M(x_1^1)\frac{dx_1^1}{d\xi} + px_1^1 = g_1^1 \quad \text{and} \quad M(x_2^1)\frac{dx_2^1}{d\xi} - px_2^1 = g_2^1,$$

where $g_1^1$ and $g_2^1$ are initial approximations to these boundary conditions, which we can take to be zero. For the next iteration we find

$$M(x_1^2)\frac{dx_1^2}{d\xi} + px_1^2 = M(x_2^1)\frac{dx_2^1}{d\xi} + px_2^1 \qquad M(x_2^2)\frac{dx_2^1}{d\xi} - px_2^1 = M(x_1^1)\frac{dx_1^1}{d\xi} - px_1^1$$

$$= (px_2^1 + g_2^1) + px_2^1 \qquad = (g_1^1 - px_1^1) - px_1^1$$

$$= 2px_2^1 \equiv g_1^2, \qquad = -2px_1^1 \equiv g_2^2.$$

Continuing in this fashion, we find that

$$M(x_1^n)\frac{dx_1^n}{d\xi} + px_1^n = 2px_2^{n-1} + g_2^{n-1} =: B_R,$$

$$M(x_2^n)\frac{dx_2^n}{d\xi} - px_2^n = -2px_1^{n-1} + g_1^{n-1} =: B_L.$$

By constructing the right hand sides of the transmission conditions recursively in this manner, we avoid approximating the derivatives required at the previous iteration.

The second step is the enforcement problem — discretizing the equations

$$M(x_1^n(\beta))\frac{dx_1^n(\beta)}{d\xi} + p\,x_1^n(\beta) = B_R, \tag{7.1}$$

$$M(x_2^n(\alpha))\frac{dx_2^n(\alpha)}{d\xi} - p\,x_2^n(\alpha) = B_L, \tag{7.2}$$

then combining them with the nonlinear system of equations to be solved on each subdomain. These conditions require the discretization of a spatial derivative at each end of a subdomain. To maintain the second order accuracy of the ODE discretization, we use central difference schemes with the "ghost point" method: introducing and then eliminating points outside the computational grid. We assume that a subdomain

has $N$ mesh points: $x_1, x_2, \ldots, x_N$, where the iteration index superscript is omitted. There are two cases to consider: a transmission condition may be imposed at the left or right endpoint of a subdomain.

### Discretization at the Left Endpoint

A centered difference discretization of (7.2) is

$$M(x_1)\left(\frac{x_2 - x_0}{2\Delta\xi}\right) - px_1 = B_L,$$

where $x_0$ is the "ghost" point. Solving for $x_0$,

$$x_0 = x_2 - \frac{2\Delta\xi}{M(x_1)}\left(B_L + px_1\right),$$

we can substitute this into a central difference equation for the first point in the subdomain, $x_1$, which we denote by $G(1)$,

$$G(1) = \frac{(M(x_2) + M(x_1))(x_2 - x_1) - (M(x_1) + M(x_0))(x_1 - x_0)}{2\Delta\xi^2}.$$

Treating $x_0$ as a function of $x_1$ and $x_2$, we find

$$\frac{\partial x_0}{\partial x_1} = -2\Delta\xi\left[\frac{pM(x_1) - (B_L + px_1)M'(x_1)}{(M(x_1))^2}\right] \quad \text{and} \quad \frac{\partial x_0}{\partial x_2} = 1.$$

Using these quantities, we can evaluate the Jacobian entries affected by this substitution of $x_0$,

$$\frac{\partial G(1)}{\partial x_1} = \frac{1}{2\Delta\xi^2}\Bigg[ M'(x_1)(x_2 - x_1) - (M(x_2) + M(x_1)) - \left(M'(x_1) + M'(x_0)\frac{\partial x_0}{\partial x_1}\right)(x_1 - x_0) - (M(x_1) + M(x_0))\left(1 - \frac{\partial x_0}{\partial x_1}\right)\Bigg],$$

$$\frac{\partial G(1)}{\partial x_2} = \frac{1}{2\Delta\xi^2}\Bigg[ M'(x_2)(x_2 - x_1) + (M(x_2) + M(x_1)) - M'(x_0)\frac{\partial x_0}{\partial x_2}(x_1 - x_0) + (M(x_1) + M(x_0))\frac{\partial x_0}{\partial x_2}\Bigg].$$

### Discretization at the Right Endpoint

Similarly, a central difference discretization of (7.1) is

$$M(x_N)\left(\frac{x_{N+1} - x_{N-1}}{2\Delta\xi}\right) - px_N = B_R,$$

where $x_{N+1}$ is the "ghost" value. Rearranging, we find

$$x_{N+1} = x_{N-1} + \frac{2\Delta\xi}{M(x_N)}\left(B_R - px_N\right),$$

which can be substituted into a central difference equation for the $N^{\text{th}}$ point in the subdomain

$$G(N) = \frac{(M(x_{N+1}) + M(x_N))(x_{N+1} - x_N) - (M(x_N) + M(x_{N-1}))(x_N - x_{N-1})}{2\Delta\xi^2}.$$

Treating $x_{N+1}$ as a function of $x_N$ and $x_{N-1}$, we calculate

$$\frac{\partial x_{N+1}}{\partial x_N} = -2\Delta\xi\left[\frac{pM(x_N) + (B_R + px_N)M'(x_N)}{(M(x_N))^2}\right] \quad \text{and} \quad \frac{\partial x_{N+1}}{\partial x_{N-1}} = 1.$$

Finally, we can use $x_{N+1}$ and its derivatives to evaluate the Jacobian entries

$$\begin{aligned}\frac{\partial G(N)}{\partial x_{N-1}} =& \frac{1}{2\Delta\xi^2}\left[M'(x_{N+1})\frac{\partial x_{N+1}}{\partial x_N}(x_{N+1}-x_N) - M'(x_{N-1})(x_N - x_{N-1})\right.\\ &\left.+(M(x_{N+1})+M(x_N))\frac{\partial x_{N+1}}{\partial x_N} + (M(x_N)+M(x_{N-1}))\right],\\ \frac{\partial G(N)}{\partial x_N} =& \frac{1}{2\Delta\xi^2}\left[\left(M'(x_{N+1})\frac{\partial x_{N+1}}{\partial x_{N-1}} + M'(x_N)\right)(x_{N+1}-x_N) - M'(x_N)(x_N - x_{N-1})\right.\\ &\left.+(M(x_{N+1})+M(x_{N+1}))\left(\frac{\partial x_{N+1}}{\partial x_{N-1}} - 1\right) - (M(x_N)+M(x_{N-1}))\right].\end{aligned}$$

### 7.2.2 Optimal Transmission Conditions

The transmission conditions of optimal iteration (4.63), the operators given by (4.64), require the integral of a function $M(x)$, which, in general, will not have an explicit antiderivative. To evaluate these conditions we use numerical quadrature, hence the optimal conditions can only be approximated in practice. Furthermore, due to the presence of derivatives, we have to address the extraction and enforcing problems, as in the case of the optimized iteration.

For extraction, we use central differences to approximate numerical derivative for overlapping subdomains, otherwise we once again proceed recursively. If we let

$$\begin{aligned}M(x_1^n)\frac{\partial x_1^n}{\partial \xi} - B_\alpha^1(x_1^n) &= P_n,\\ M(x_2^n)\frac{\partial x_2^n}{\partial \xi} - B_\alpha^2(x_2^n) &= Q_n,\end{aligned}$$

then the boundary conditions can be expressed as

$$M(x_1^n)\frac{\partial x_1^n}{\partial \xi} - B_\alpha^1(x_1^n) = B_\alpha^2(x_2^{n-1}) - B_\alpha^1(x_2^{n-1}) + Q_{n-1}, \tag{7.3}$$

$$M(x_2^n)\frac{\partial x_2^n}{\partial \xi} - B_\alpha^2(x_2^n) = B_\alpha^1(x_1^{n-1}) - B_\alpha^2(x_1^{n-1}) + P_{n-1}. \tag{7.4}$$

To enforce these boundary conditions we once again use central differences with "ghost" points introduced. Discretizing (7.4), denoting the known right hand side by $B_L$, we obtain

$$M(x_1)\left(\frac{x_2 - x_0}{2\Delta\xi}\right) - \frac{1}{\alpha}\int_0^{x_1} M(x)dx = B_L.$$

Rearranging, we find

$$x_0 = x_2 - \frac{2\Delta\xi}{M(x_1)}\left(B_L + \frac{1}{\alpha}\int_0^{x_1} M(x)dx\right),$$

from which we can compute the partial derivatives:

$$\frac{\partial x_0}{\partial x_1} = -\frac{2\Delta\xi}{\alpha\left(M(x_1)\right)^2}\left[\left(M(x_1)\right)^2 - \left(\alpha B_L + \int_0^{x_1} M(x)dx\right)M'(x_1)\right],$$

$$\frac{\partial x_0}{\partial x_2} = 1.$$

Similarly, denoting the right hand side of (7.3) by $B_R$, we discretize to find

$$M(x_N)\left(\frac{x_{N+1} - x_{N-1}}{2\Delta\xi}\right) - \frac{1}{1-\beta}\int_{x_N}^1 M(x)dx = B_R,$$

which can be rearranged to obtain

$$x_{N+1} = x_{N-1} + \frac{2\Delta\xi}{M(x_N)}\left(B_R + \frac{1}{1-\beta}\int_{x_N}^1 M(x)dx\right).$$

Finally, we compute the derivatives:

$$\frac{\partial x_{N+1}}{\partial x_{N-1}} = 1,$$

$$\frac{\partial x_{N+1}}{\partial x_N} = -\frac{2\Delta\xi}{(1-\beta)\left(M(x_N)\right)^2}\left[\left(M(x_N)\right)^2 + \left((1-\beta)B_R + \int_{x_N}^{1} M(x)dx\right)M'(x_N)\right].$$

With these values known, we can construct the Jacobian for both problems as in the optimized Schwarz case.

## 7.3 1D Numerical Results

We now present some numerical results to illustrate the various 1D Schwarz iterations proposed in Chapters 4 and 5. For steady problems we use the test function

$$u(x) = \tfrac{1}{2}(1 + \tanh(20x) - \tanh(12(x-0.4)) + \tanh(18(x-0.7))), \tag{7.5}$$

which we plot in Figure 7.1. As can be seen, this function has three regions of rapid changes in function values: near $x = 0$, near $x = 0.4$ and near $x = 0.7$. As such, we would expect mesh points to be clustered around these values of $x$ in an equidistributed mesh. Indeed, that is what we observe in Figure 7.2, where we show the location of mesh points as determined by the optimized Schwarz algorithm, beginning at the uniform initial mesh (bottom) and concluding at the equidistributed mesh (top).

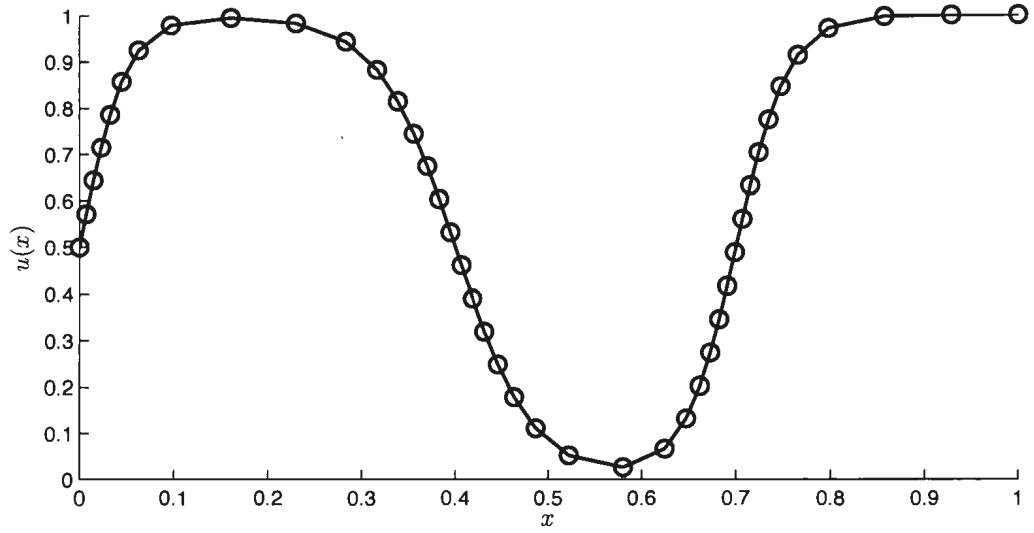


Figure 7.1: The test function (7.5) used for steady mesh generation.

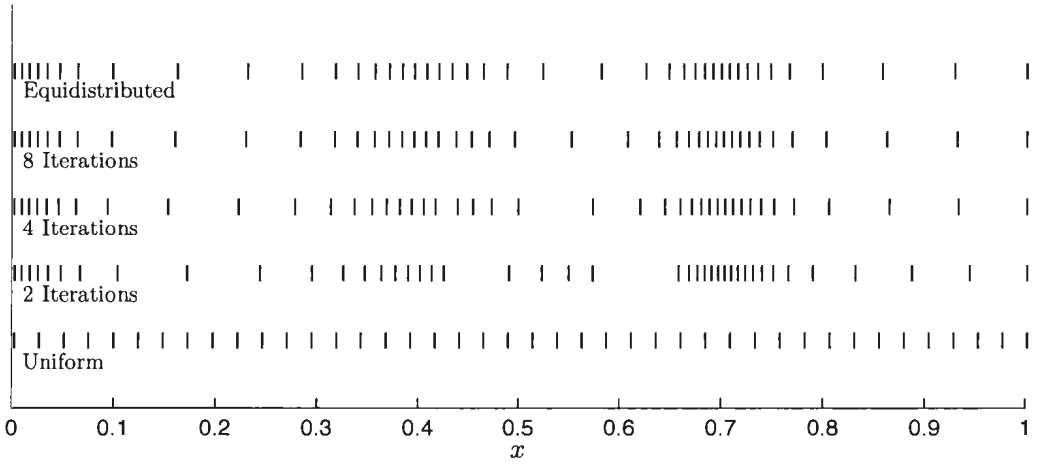


Figure 7.2: Location of mesh points as determined by optimized Schwarz, using parameter $p = 2$ and 40 mesh points, for the test problem (7.5).

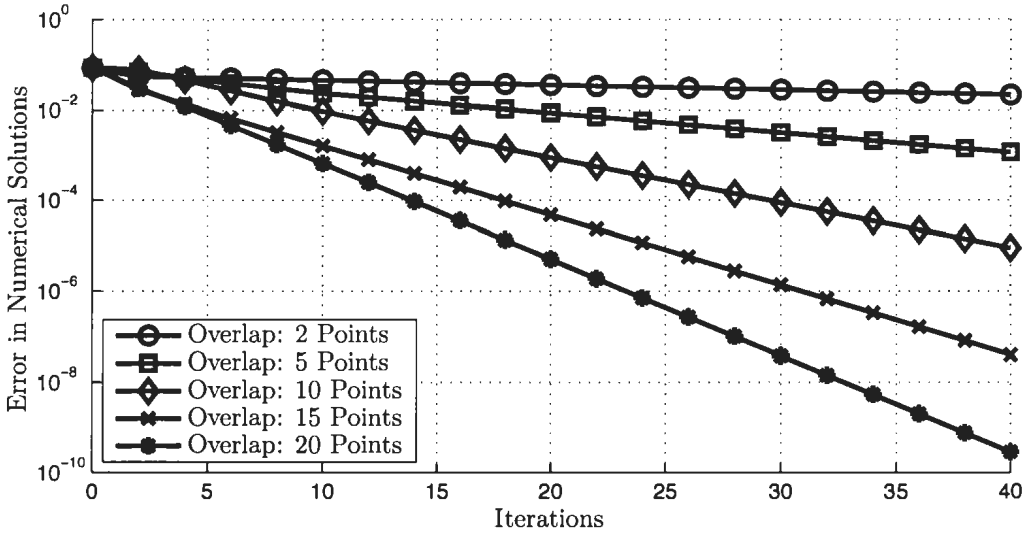


Figure 7.3: Convergence histories for parallel classical Schwarz with varying overlap on two subdomains and 80 mesh points total.

We begin by discussing the classical Schwarz iterations of Section 4.2 (p. 57). As noted in Theorem 4.7 (p. 58), the contraction factor $\rho$ depends on the extent of overlap between subdomains,

$$\rho := \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha}.$$

We illustrate this in Figure 7.3, where we show convergence histories for different amounts of overlap between subdomain, varying from two points, the fewest possible which will allow convergence, up to 20 points, which is 25% of the total number of mesh points. We note that in this, and all subsequent convergence histories, the error recorded is the maximum difference between the solution over the first subdomain

and the single domain numerical solution. The trend observed is that increasing the number of points shared between subdomains can greatly improve the rate of convergence, however this comes at the cost of making the subdomain problems larger, hence more costly to solve.

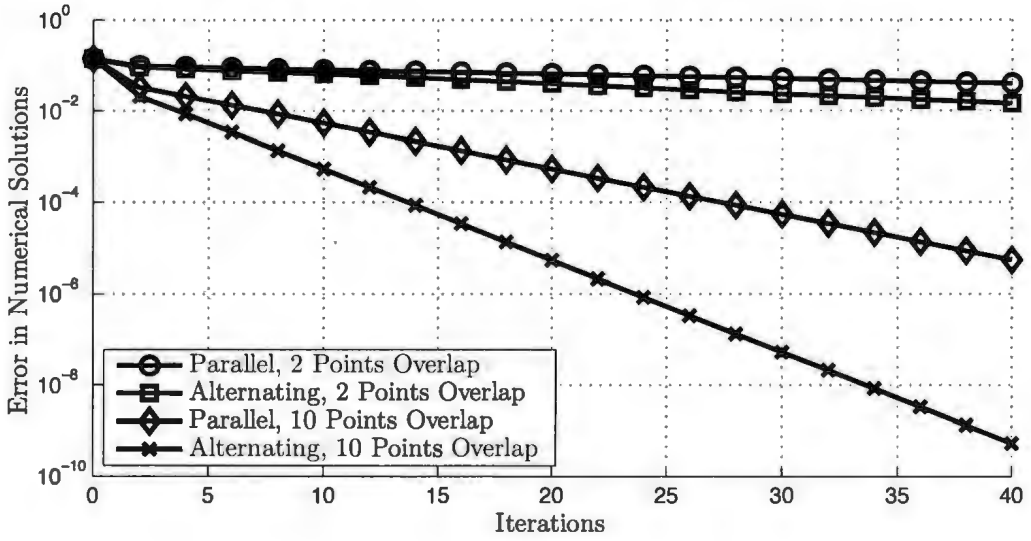


Figure 7.4: Convergence histories for parallel and alternating classical Schwarz on two subdomains and 80 mesh points total.

In Figure 7.4 we highlight the potential benefit of using alternating methods by comparing the parallel and alternating classical Schwarz iterations for two different amounts of overlap. In both cases we see that the alternating method outperforms the parallel method, with the extent of the improvement increasing with the amount of overlap. The reason for this can be seen by comparing (4.15) in Theorem 4.7 (p. 58)

and (4.31) in Theorem 4.9 (p. 63): the right hand side of (4.15) gains a factor of $\rho$ every two iterations, whereas the bound of (4.31) is multiplied by $\rho$ at every iteration, hence the improvement due to increased overlap is compounded twice as often. The obvious drawback, as previously mentioned, is the loss of obvious parallelization.

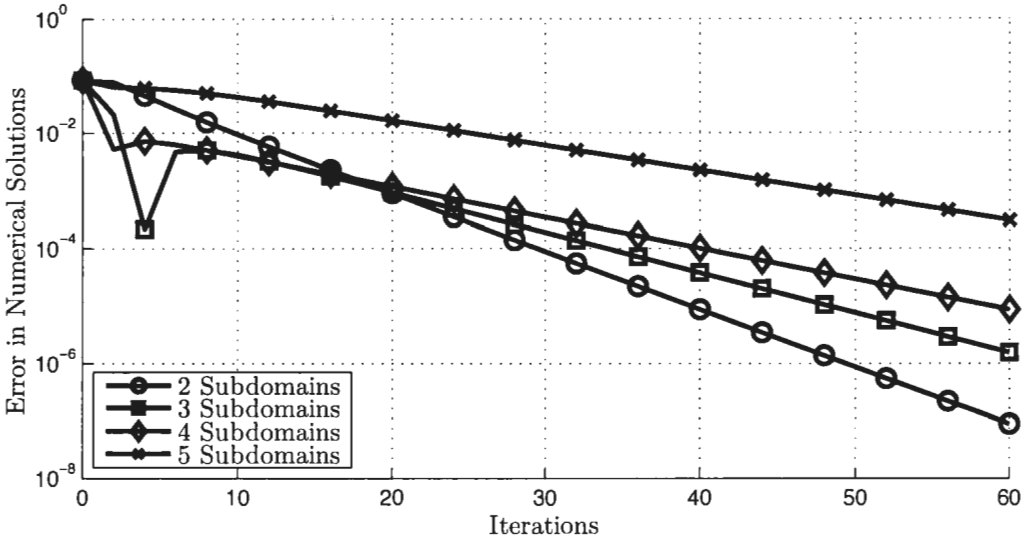


Figure 7.5: Convergence histories for parallel classical Schwarz with varying numbers of subdomains. Each case uses a total of 80 Mesh Points and 10 points of overlap between adjacent subdomains.

In addition to overlap, the number of subdomains also affects the rate of convergence of a DD algorithm. In Figure 7.5 we show convergence histories for parallel classical Schwarz iterations with varying numbers of subdomains. Following some initial transient sharp changes in the error recorded, we see that the rate of convergence

decreases as the number of subdomains used increases. This agrees with the estimates of Theorem 4.14 (p. 70): the error bounds involve the contraction rate estimate,

$$\left(1 - 4r(1-r)\sin^2\frac{\pi}{2(S+1)}\right),$$

and as $S$ increases this quantity approaches 1, resulting in slower convergence.

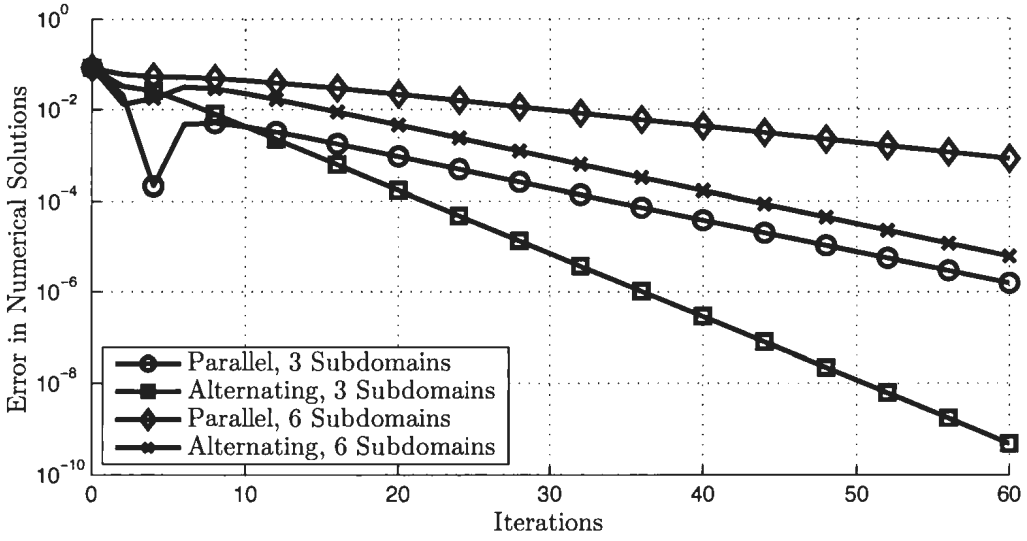


Figure 7.6: Convergence histories for parallel and alternating classical Schwarz. Each case uses 80 mesh points in total and 10 points of overlap between adjacent subdomains.

In Figure 7.6 we again compare the parallel multidomain method of Section 4.2.2 to the alternating multidomain method of Section 4.2.3, this time for three and six subdomains. We see that the trends previously observed in Figures 7.4 and 7.5

continue: alternating methods outperform parallel methods when considered in terms of DD iterations, and more subdomains require more iterations to reach a given level of accuracy.

In Figure 7.7 we include the other multidomain classical Schwarz iteration — the red-black method of Section 4.2.4. As promised in the discussion, we see that this red-black iteration converges at approximately the same rate as the alternating iteration, with the added benefit of being able to implement this iteration in parallel.

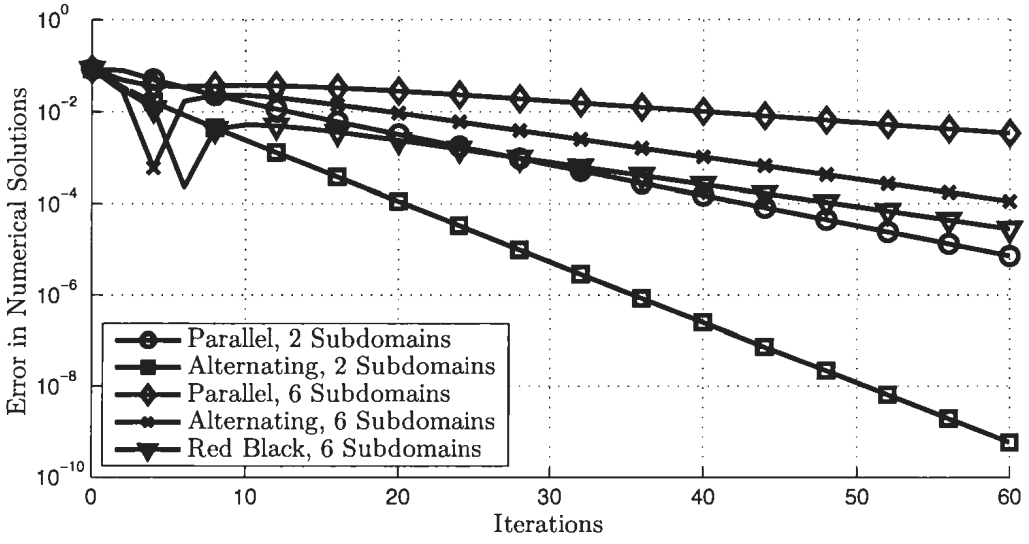


Figure 7.7: Convergence histories comparing parallel, alternating, and red-black iterations for two and six subdomains (recall that there is no difference between the alternating and red-black iterations for two subdomains). Each case uses 120 mesh points in total and 10 points of overlap between adjacent subdomains.

We now turn to the optimal Schwarz iterations proposed in Section 4.3 (p. 91). We plot convergence histories for the optimal iteration in Figure 7.8, including a plot for parallel classical Schwarz to facilitate comparison. We observe that, while these iterations fail to achieve convergence in two iterations as in the continuous case, they vastly outperform the classical iteration. Furthermore, by increasing the number of mesh points used throughout the domain, we can significantly improve the convergence of the optimal iteration, with convergence in two iterations being the limiting case.

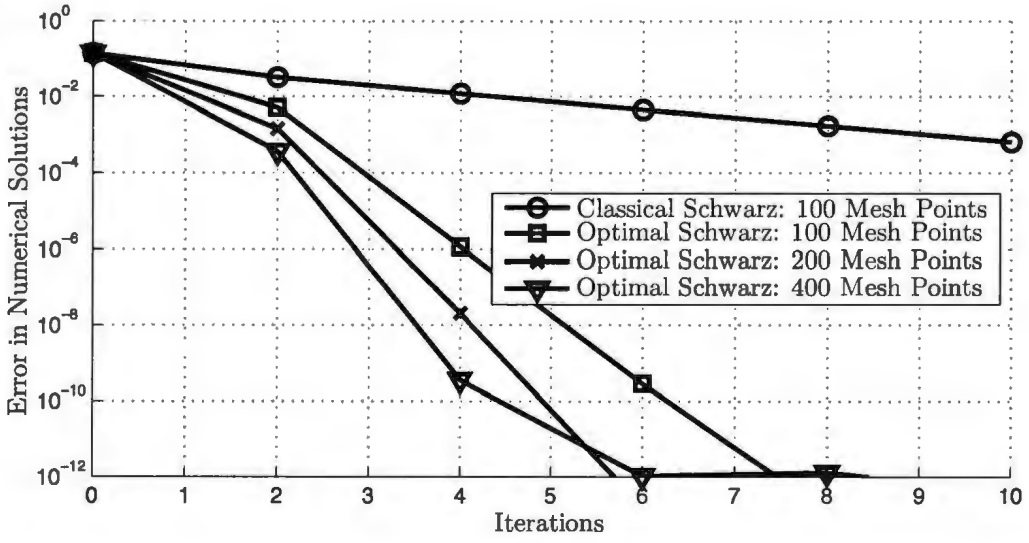


Figure 7.8: Convergence histories for non-overlapping parallel optimal Schwarz for varying numbers of mesh points. Classical Schwarz is plotted using 15 points of overlap between subdomains.

As discussed in Sections 4.3.3 (p. 100) and 4.3.4 (p. 104), we can extend the optimal Schwarz iteration to three subdomains and maintain the optimal convergence, however the optimality is lost for four or more subdomains. In Figure 7.9 we compare these optimal iterations to the parallel classical Schwarz iterations on the same number of subdomains. We observe that while there is a significant drop off in the rate of convergence when comparing the plots for three and four subdomains, in all cases the optimal iterations do much better than classical Schwarz, suggesting that they can offer a significant improvement for numerical calculations.

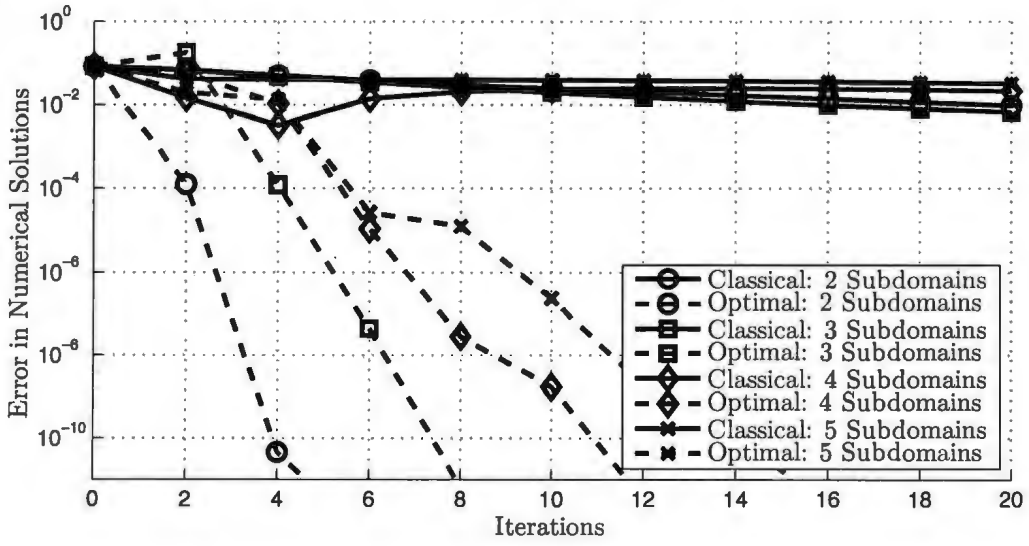


Figure 7.9: Optimal multidomain iterations. Each plot is generated using 500 mesh points in total, with 25 points of overlap between subdomains for classical Schwarz and one shared point between subdomains otherwise.

Following optimal Schwarz, we discussed optimized Schwarz in Section 4.4 (p. 113). In Figure 7.10 we have convergence histories for optimized Schwarz where we vary the transmission parameter, $p$. By doing so, we see that by increasing $p$ from 4 to 12 we first observe improved convergence for $p = 6$ and $p = 8$, little change from $p = 8$ to $p = 10$, then slower convergence for $p = 12$. This is a common observation for optimized Schwarz, with the best convergence attained for some $p^* > 0$ and convergence deteriorating as $p$ moves away from this value.

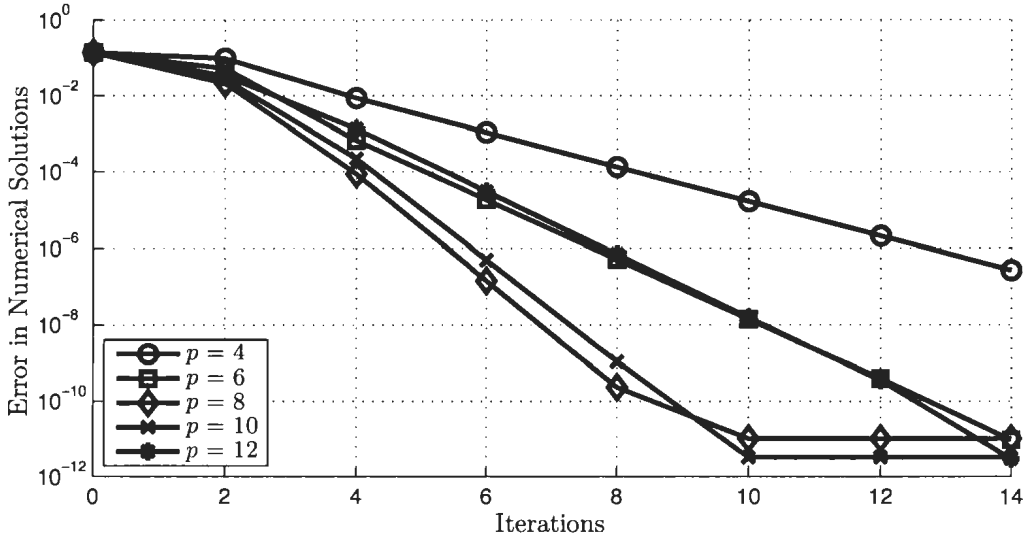


Figure 7.10: Convergence histories for parallel optimized Schwarz with different values of the transmission parameter $p$. In each case we use two subdomains and 80 mesh points in total.

We compare the three types of iterations described thus far in Figure 7.11. In this

plot we observe: optimal Schwarz converges the fastest, classical Schwarz converges the slowest, and optimized Schwarz will fall somewhere in between. Furthermore, by adjusting the transmission parameter $p$, optimized Schwarz can closely approximate the convergence results of optimal Schwarz.

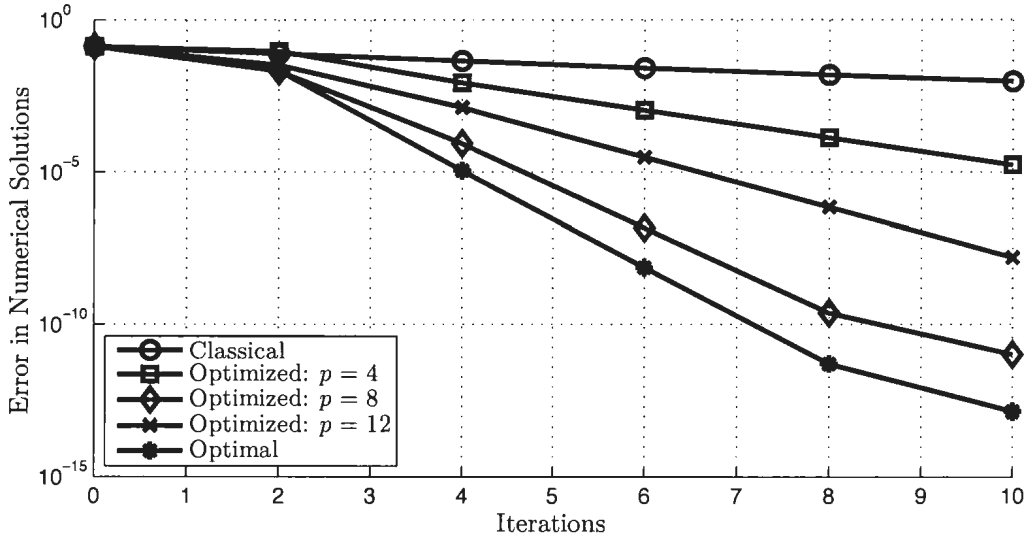


Figure 7.11: Convergence histories comparing classical Schwarz, optimal Schwarz, and optimized Schwarz. Each case used 2 subdomains with 80 mesh points in total, and 2 points of overlap except for classical with 10 points.

As a modification to the classical Schwarz iteration, which requires the solution of a nonlinear system on each subdomain, in Section 4.5 we discuss linearized classical Schwarz iterations, in which we freeze the argument of the mesh density function $M(x)$ at the previous iteration to obtain a linear system. We compare the convergence of

both parallel and alternating linearized methods versus their nonlinear counterparts in Figures 7.12 and 7.13.

In Figure 7.12 we plot convergence histories versus the number of DD iterations. The convergence of the linearized iterations is noticeably slower than the corresponding nonlinear iteration. We also note that alternating methods again outperform parallel methods.

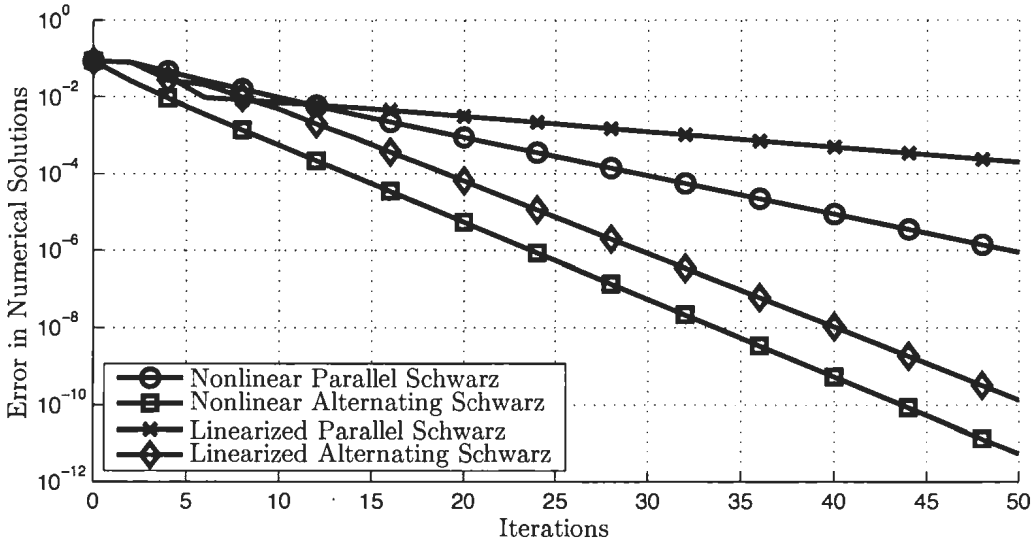


Figure 7.12: Convergence histories for parallel and alternating versions of both linearized classical Schwarz and standard (nonlinear) classical Schwarz, showing error versus number of DD iterations. Each case generated using two subdomains with 10 points of overlap and 80 mesh points in total.

The true benefit of linearized iterations is seen when we plot the error versus the

number of linear systems solved in Figure 7.13. Here we see that the linearized results do much better per linear solve than the nonlinear iterations. This is not surprising, as the nonlinear subdomain problems will require multiple linear solves in each DD iteration, whereas linearized Schwarz methods only require a single linear system to be solved at each DD iteration.

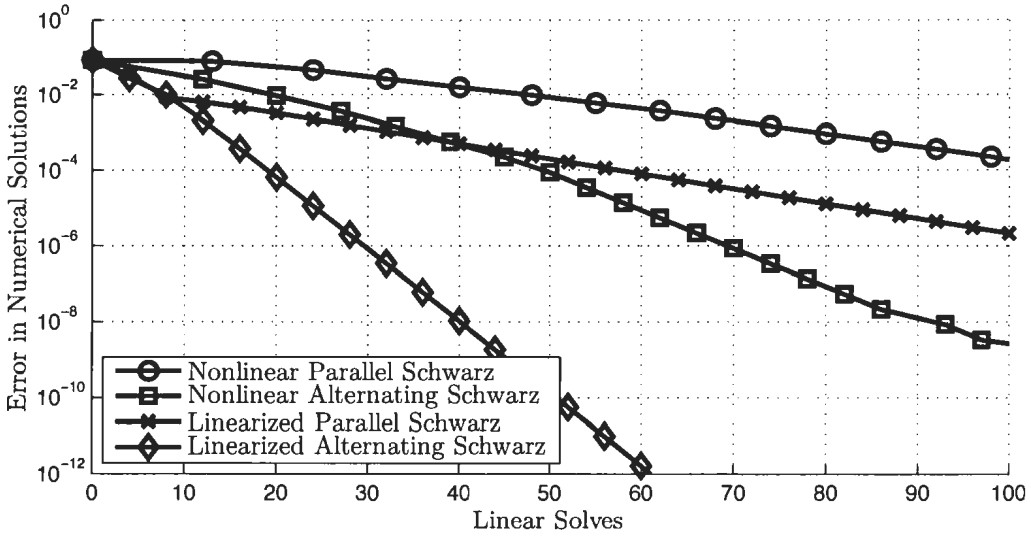


Figure 7.13: Convergence histories for parallel and alternating versions of both linearized classical Schwarz and standard (nonlinear) classical Schwarz, showing error versus number of linear solves.

In Section 4.5 we also introduced the idea of linearized iterations for optimized Schwarz iterations (p. 122). We compare linearized and nonlinear versions of both classical and optimized Schwarz iterations in Figure 7.14, which shows that just as

optimized Schwarz outperforms classical Schwarz, linearized optimized Schwarz outperforms linearized classical Schwarz.

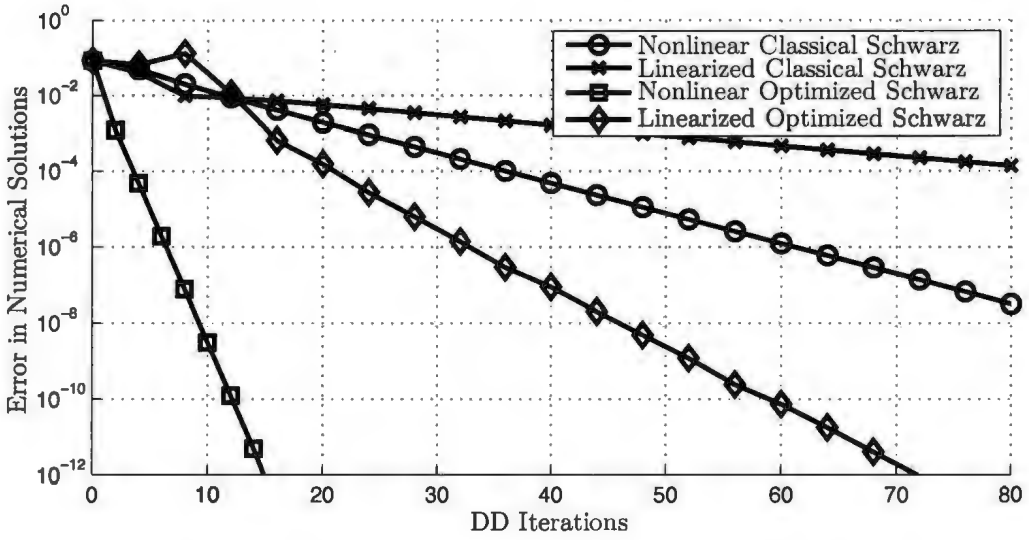


Figure 7.14: Convergence histories for linearized and nonlinear classical Schwarz and optimized Schwarz iterations, showing error versus number of DD iterations. In each case we use 100 mesh points total, 10 points of overlap between subdomains, and transmission parameter $p = 8$.

In addition to convergence histories, we can consider the interpolation error resulting from the use of equidistributed meshes. For interpolation error we report the maximum error between the function $u(x)$ and the linear interpolating polynomial formed using the values of $u(x)$ for a given mesh. We report interpolation errors for the function $u(x)$ given by (7.5) in Table 7.1 after the stated number of DD itera-

tions. In the column headed by 0 we report the interpolation error for a uniform mesh, and in the column headed by $\infty$ we give interpolation error for the single domain equidistributed mesh.

Table 7.1: Interpolation errors for grids obtained by various Schwarz methods at given iterations. For each case we use 100 mesh points, 10 points of overlap for classical Schwarz methods, 1 point of overlap otherwise, and $p = 8$ for the optimized transmission parameter.

| Iterations | 0 | 2 | 4 | 6 | 8 | 10 | $\infty$ |
|---|---|---|---|---|---|---|---|
| Classical (Par) | 0.1993 | 0.0627 | 0.0614 | 0.0605 | 0.0598 | 0.0594 | 0.0584 |
| Classical (Alt) | 0.1993 | 0.0605 | 0.0594 | 0.0589 | 0.0586 | 0.0585 | 0.0584 |
| Linearized (Par) | 0.1993 | 0.1287 | 0.0810 | 0.0730 | 0.0709 | 0.0684 | 0.0584 |
| Linearized (Alt) | 0.1993 | 0.1282 | 0.0883 | 0.0577 | 0.0581 | 0.0582 | 0.0584 |
| Optimal (Par) | 0.1993 | 0.0579 | 0.0578 | 0.0578 | 0.0578 | 0.0578 | 0.0584 |
| Optimal (Alt) | 0.1993 | 0.0578 | 0.0578 | 0.0578 | 0.0578 | 0.0578 | 0.0584 |
| Optimized (Par) | 0.1993 | 0.0607 | 0.0584 | 0.0584 | 0.0584 | 0.0584 | 0.0584 |
| Optimized (Alt) | 0.1993 | 0.0584 | 0.0584 | 0.0584 | 0.0584 | 0.0584 | 0.0584 |

In almost all cases, the mesh obtained by DD would give almost exactly the same interpolation error as the single domain mesh after 10 iterations. Indeed, the optimal Schwarz and optimized Schwarz methods result in meshes of almost identical quality after the initial two iterations. As such, rather than iterating to meet a strict tolerance, the mesh obtained after 2–4 DD iterations will often be sufficient for most computational purposes.

As the final 1D example, we consider the time dependent case of Chapter 5. In

the convergence results of Theorems 5.3 (p. 126) and 5.6 (p. 135) we have bounds on the contraction rate of

$$\frac{\sinh(\sqrt{\theta}\alpha)}{\sinh(\sqrt{\theta}\beta)}\frac{\sinh(\sqrt{\theta}(1-\beta))}{\sinh(\sqrt{\theta}(1-\alpha))}, \quad \text{where} \quad \theta = \frac{\tau}{\Delta t}\frac{1}{\hat{m}}.$$

By decreasing the time step $\Delta t$, we cause $\theta$ to grow, resulting in a lower bound for the contraction rate, hence we can expect faster convergence. In Figure 7.15 we plot both parallel and alternating classical Schwarz iterations for the time dependent problem for several different time steps using the test problem

$$u(x,t) = \frac{1}{2}[1 - \tanh(c(t)[x - t - 0.4]), \quad c(t) = 1 + \frac{199}{2}[1 + \tanh(50t - 2.5)].$$

We observe that decreasing the time step does result in greater accuracy after a given number of DD iterations. Furthermore, we note that, as in the steady case, the alternating algorithm fares better than the parallel algorithm in each case. Further results also carry over from the steady case: we can say that, in general, by increasing overlap we improve the rate of convergence, and that by increasing the number of subdomains used the convergence of the iteration will slow.

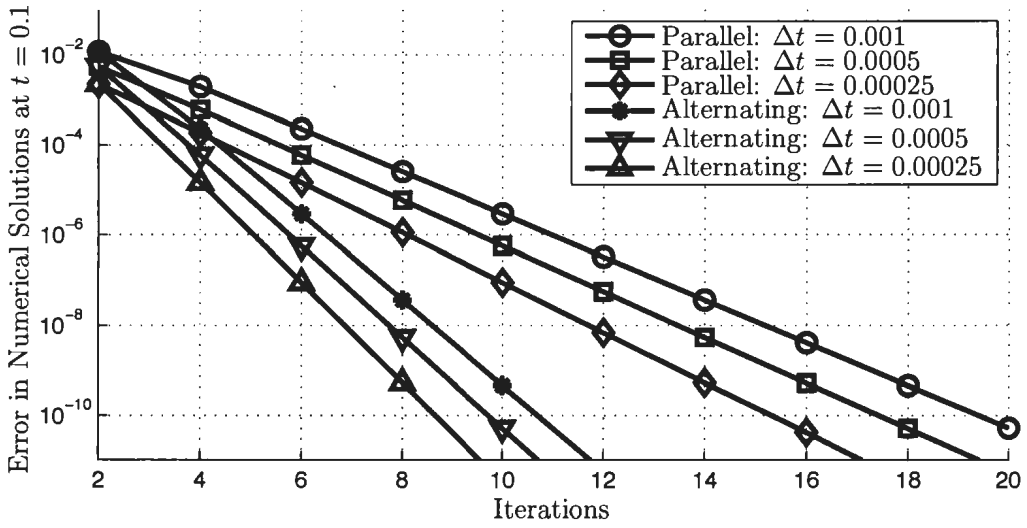


Figure 7.15: Convergence histories for parallel and alternating classical Schwarz iterations at $t = 0.1$ for several time steps $\Delta t$. All plots obtained using 50 mesh points in total, with 20 points of overlap between two subdomains.

## 7.4 2D Numerical Implementation

To approximate the solution of the system of equations enforcing local equidistribution,

$$\begin{aligned} \left[ \begin{pmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \end{pmatrix}^T M \begin{pmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \end{pmatrix} \right]^{1/2} &= c_1(\eta), \\ \left[ \begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \end{pmatrix}^T M \begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \end{pmatrix} \right]^{1/2} &= c_2(\xi), \end{aligned} \tag{7.6}$$

the physical boundary conditions and transmission conditions are discretized using standard second order finite differences on a uniform grid in the computational $(\xi, \eta)$ variables as described in Section 2.4.1 (p. 30).

To discretize transmission conditions we use centered differences, introducing "ghost" values for $x$ and $y$ adjacent to the appropriate boundary; but unlike for the 1D equations, these ghost values cannot be eliminated. As such, the ghost values and the actual mesh are determined simultaneously, resulting in an additional $2N_\eta$ equations to be solved, $N_\eta$ being the number of mesh lines in the $\eta$ direction, with the ghost values determined from the transmission conditions and the mesh by solving the original system of equations. We note that the nonlinear transmission conditions of Case Two through Case Four (p. 143) result in additional nonlinear equations to be solved at the interface. However, this is not a significant disadvantage, as the original system (7.6) is nonlinear, hence must be solved by a Newton iteration or some other rootfinding method for nonlinear systems. Furthermore, when equidistributing boundary conditions are used, we must transmit two additional pieces of data: the $x$-coordinate of the mesh point adjacent to the boundary in the opposite subdomain, so that the 1D equidistributing principle can be solved using central differences throughout.

A final note on the implementation of the possible optimized conditions of Sec-

tion 6.2. In theory these conditions can be used for non-overlapping subdomains, in practice this presents a challenge. If we use forward and backward finite differences for numerical approximation of derivatives in the transmission conditions, the convergence of the iteration will suffer, typically failing to agree with the single domain result with sufficient accuracy. Attempts at extending the recursive approach to handle the extraction problem in 1D have not been successful. As such, in the following 2D numerical results, we will restrict ourselves to considering overlapping subdomains for all experiments.

## 7.5 2D Numerical Results

To illustrate the 2D mesh iterations, we use the test function $u(x, y)$ given by

$$u(x, y) = (1 - e^{(R(x-1))}) \sin(\omega y), \tag{7.7}$$

where $R$ and $\omega$ are user defined parameters. In Figure 7.16 we show the mesh determined by classical Schwarz for $u(x, y)$ using parameters $R = 15$ and $\omega = 1.5\pi$, the mesh on $\Omega_1$ colored red, the mesh on $\Omega_2$ colored blue, and the mesh on the overlapping region colored purple.

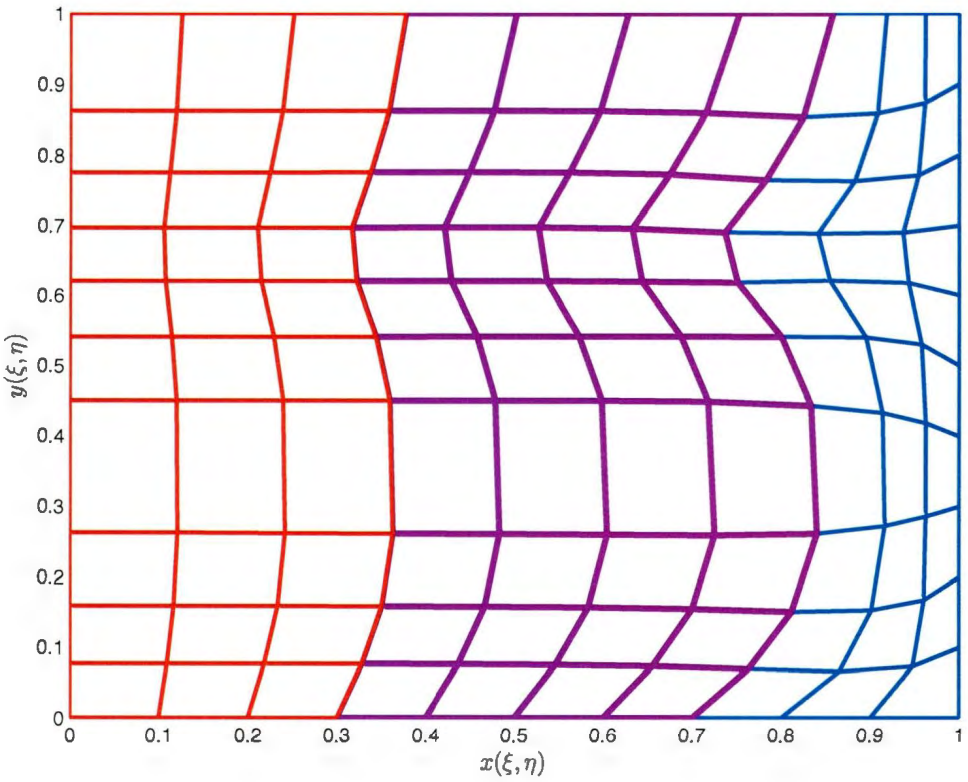


Figure 7.16: The mesh determined for $u(x, y)$ given by (7.7). We use classical Schwarz with an $11 \times 11$ mesh for the entire domain and 5 lines of overlap between subdomains. The mesh parameters are $a = 0.75$ and $b = 0.05$.

Comparing to Figure 7.17, which shows the function $u(x, y)$ plotted using the mesh of Figure 7.16, we see that mesh points are concentrated in a vertical strip for $x$ near 1 due to the exponential term and in horizontal strips near $y$ values for which the sine function is near zero.

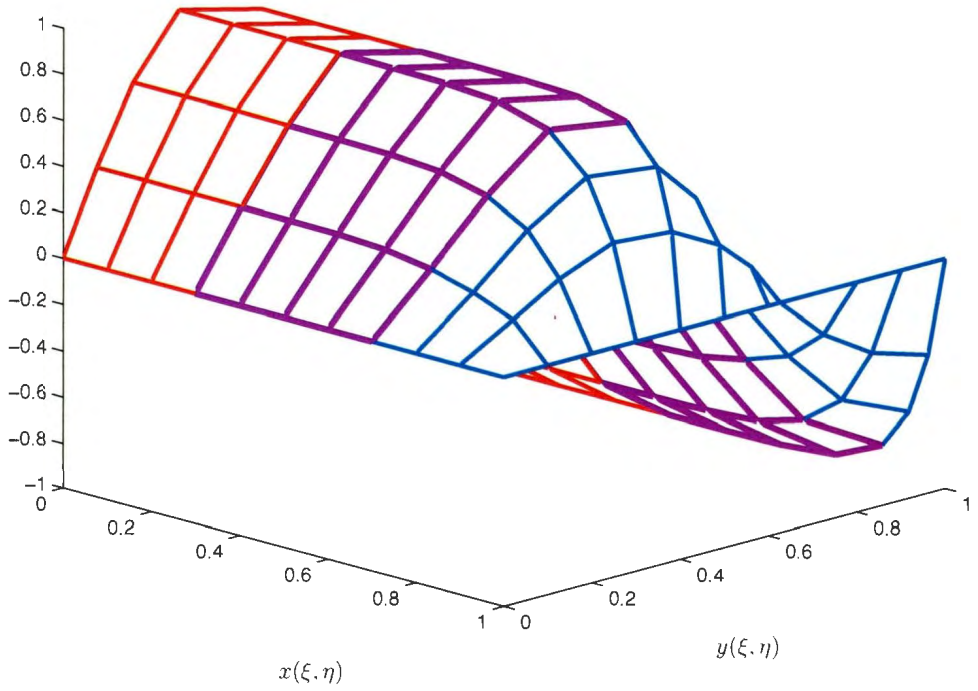


Figure 7.17: The function $u(x, y)$ given by (7.7) plotted using its locally equidistrbuted mesh as shown in Figure 7.16.

To compare the various Schwarz iterations proposed, we once again make use of convergence histories. Once again solving the mesh equations with parameters $R = 15$ and $\omega = 1.5\pi$ for (7.7), we plot convergence histories for the classical Schwarz iteration of Section 6.1 (p. 141) for varying amounts of overlap. We use $x_1$, the $x$ component for the first subdomain, as a representative of the overall solution to calculate error

with respect to the single domain numerical solution. As in the 1D case, we see that convergence steadily improves with the amount of overlap, the downside being that the subdomain problems become more computationally expensive as a result.

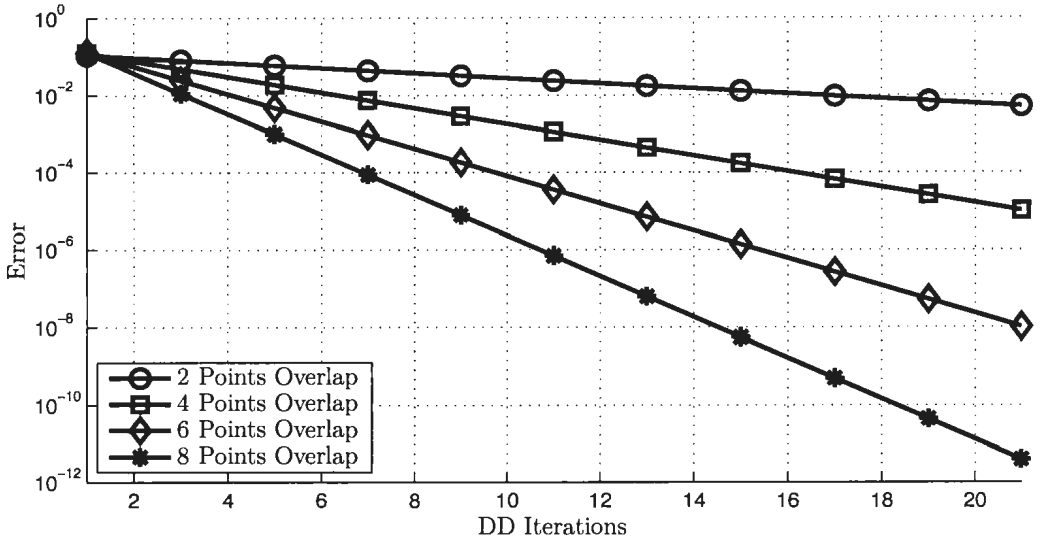


Figure 7.18: Convergence histories for classical Schwarz with varying amounts of subdomain overlap. The iteration results in a $14 \times 14$ mesh for the entire domain, and used mesh parameters $a = 0.8$ and $b = 0.1$.

The decision to focus only on the convergence results for $x_1$ is not unreasonable, as we show in Figure 7.19 by plotting the error in both $x$ and $y$ components for each subdomain. We see that the error of each solution component decreases at approximately the same rate, hence any one of them can be used to illustrate the general results for all.

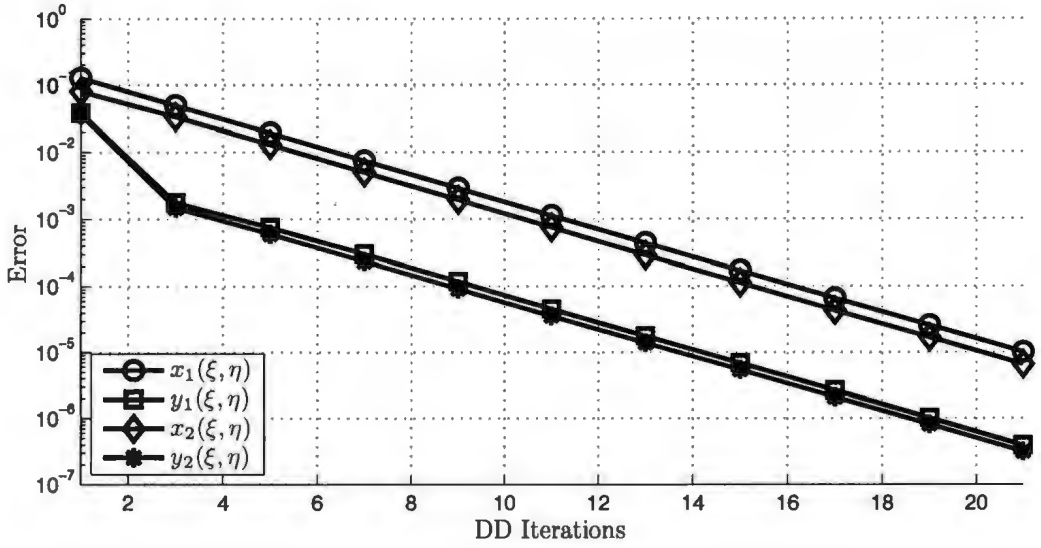


Figure 7.19: Convergence histories for classical Schwarz showing the error for each solution component. This is the case for 4 lines of overlap from Figure 7.18.

To illustrate the various possible optimized Schwarz transmission conditions, we first begin with a comparison of the methods outlined in Section 6.2 (p. 143). We once again use $u(x, y)$ as described in (7.7), changing the parameters to $R = 15$ and $\omega = \pi$. As the Figure shows, all optimized methods vastly outperform the classical iteration, though there is some variation between the convergence attained between any given pair of optimized methods. As such, the actual choice of transmission conditions to be used will be a question of which allows theoretical convergence results to be established in the future.

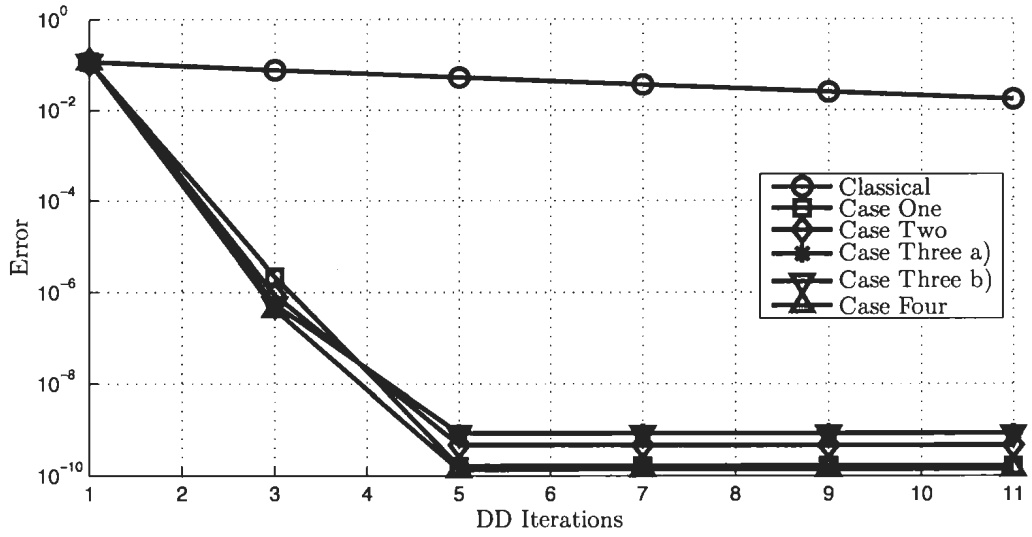


Figure 7.20: Comparison of the possible 2D optimized Schwarz iterations. In each case we use two lines of overlap between subdomains to obtain a $12 \times 12$ mesh over the entire domain. We use transmission parameter $p = 2$ and mesh parameters $a = 0.7$ and $b = 0.05$.

To illustrate the effect of the transmission parameter $p$, we restrict ourselves to considering Case One of Section 6.2, which are simple linear Robin conditions. In Figure 7.21 we plot convergence histories for varying values of $p$, generating the mesh for (7.7) with parameters $R = 25$ and $\omega = 1.5\pi$. As in the 1D example of Figure 7.10, by using an appropriate value of $p$ we can optimize the convergence of this iteration.

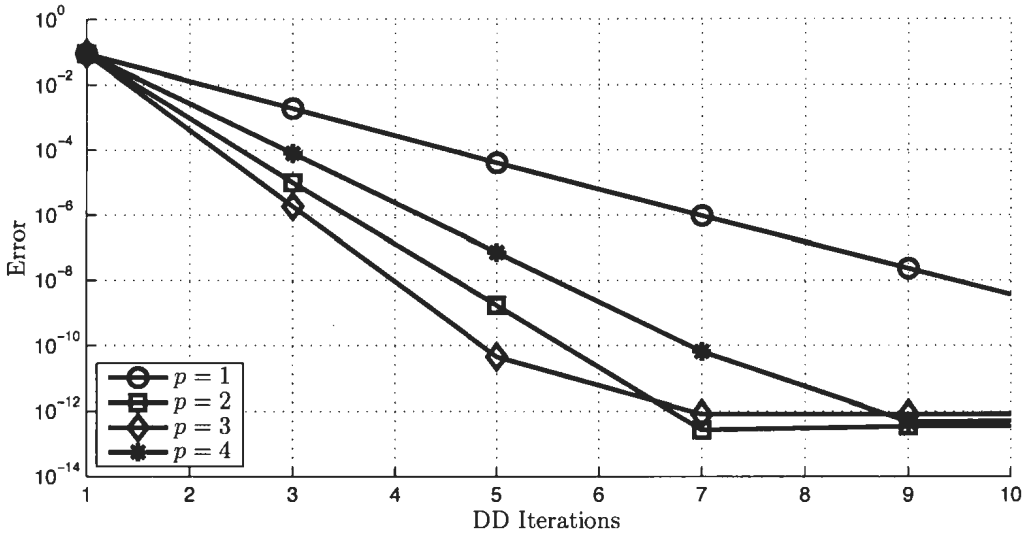


Figure 7.21: Convergence histories for the optimized Schwarz iteration using linear Robin conditions (Case One). We use three lines of overlap between subdomains to obtain an $11 \times 11$ mesh for the entire domain. We use mesh parameters $a = 0.75$ and $b = 0.1$.

Another way to assess the mesh obtained from a DD iteration is to compute a mesh quality measure. In particular, we consider the equidistribution quality measure for each element $K$ of the grid, $Q_{eq}(K)$, as presented in [39]. The maximum of $Q_{eq}$ over all elements is 1 if and only if the equidistribution condition is satisfied exactly. The larger the value of $\max_K Q_{eq}(K)$ the farther the mesh is from equidistributing the monitor matrix $M$.

For the function $u(x, y)$ of (7.7) with parameters $R = 25$ and $\omega = \pi$, we record the

values of $Q_{eq}(K)$ after a given number of iterations in Table 7.2. As for interpolation error in the 1D case, the zero column corresponds to a uniform mesh and the $\infty$ column with the single domain equidistributing mesh. Much like with the interpolation errors recorded in Table 7.1 for the 1D case, we see that after a few iterations the mesh produced by optimized methods are of approximately the same quality as the single domain equidistributing mesh. Similarly, the classical iteration succeeds in providing a reasonable approximation after the fifth iteration. As such, the mesh obtained by stopping any of DD algorithms after a small number of iterations will be as good as a mesh obtained by letting the iteration reach an arbitrary level of accuracy.

Table 7.2: Mesh quality measures for the grids obtained by the proposed Schwarz iterations of Chapter 6. Each iteration uses two subdomains with two lines of overlap to produce a $12 \times 12$ mesh over the entire domain. Mesh parameters are $a = 0.8$ and $b = 0.05$.

| Iterations | 0 | 1 | 2 | 3 | 4 | 5 | $\infty$ |
|---|---|---|---|---|---|---|---|
| Classical | 1.7910 | 1.4245 | 1.5699 | 1.3496 | 1.4082 | 1.2931 | 1.2116 |
| Case One | 1.7910 | 1.9786 | 1.2116 | 1.2116 | 1.2116 | 1.2116 | 1.2116 |
| Case Two | 1.7910 | 1.9852 | 1.2116 | 1.2116 | 1.2116 | 1.2116 | 1.2116 |

# Chapter 8

# Summary and Future Work

In this thesis we explored methods which combine the techniques of mesh adaptation and domain decomposition for boundary value problems. In Chapter 2 we discussed the idea of moving mesh methods derived from the equidistribution principle for both steady and time dependent problems and in one or more spatial dimensions, providing theoretical details on the derivation of governing equations as well as information on how to implement such methods numerically. Chapter 3 introduced the basics of domain decomposition, exploring the historical origin and development of what are known as Schwarz domain decomposition methods. We discussed the connection between domain decomposition for continuous and discrete problems, and described ways in which the convergence of the domain decomposition algorithms can be im-

proved.

In Chapter 4 we presented results for the combination of mesh adaptation and domain decomposition for one dimensional steady problems, some results from earlier papers and others appearing for the first time. We established the convergence of three general types of methods for the mesh generation problem: classical Schwarz, optimal Schwarz, and optimized Schwarz, each type corresponding to a different type of transmission condition used between subdomains. In all cases we established convergence of both parallel and alternating iterations, and for classical Schwarz we proved the convergence of several possible iterations for arbitrarily many subdomains. We also considered the idea of linearized methods for classical Schwarz and optimized Schwarz iterations, which eliminate the need to solve nonlinear systems of equations.

In Chapter 5 we extended the classical Schwarz iterations proposed in Chapter 4 to handle time dependent mesh equations, allowing this combined mesh adaptation and domain decomposition approach to be used for parabolic problems, in addition to elliptic problems. Chapter 6 explores the further extension of the classical and optimized domain decomposition algorithms to two dimensional mesh generation. Finally, in Chapter 7 we provided numerous numerical results illustrating the iterations discussed in the preceding chapters.

Future steps to be taken include establishing the convergence of optimized Schwarz

algorithms for overlapping subdomains and multiple domains in the 1D mesh generation problem, both of which have been observed numerically. It also remains to prove convergence for optimized Schwarz methods for the 1D time dependent mesh problem. Finally, while convergence of the classical Schwarz and optimized Schwarz iterations proposed for 2D mesh problems has been observed numerically, it remains to prove a general convergence result for these methods.

# Bibliography


[1] Uri M. Ascher. DAEs that should not be solved. In *Dynamics of algorithms (Minneapolis, MN, 1997)*, volume 118 of *IMA Vol. Math. Appl.*, pages 55–67. Springer, New York, 2000.

[2] I. Babuška and Werner C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15(4):736–754, 1978.

[3] George Beckett, John A. Mackenzie, Alison Ramage, and David M. Sloan. Computational solution of two-dimensional unsteady PDEs using moving mesh methods. *J. Comput. Phys.*, 182(2):478–495, 2002.

[4] Daniel Bennequin, Martin J. Gander, and Laurence Halpern. A homographic best approximation problem with application to optimized Schwarz waveform relaxation. *Math. of Comp.*, 78(265):185–232, 2009.

[5] Morton Bjorhus. *On Domain Decomposition, Subdomain Iteration, and Waveform Relaxation.* PhD thesis, Norwegian Institute of Technology, The University of Trondheim, 1995.

[6] Igor P. Boglaev. Iterative algorithms of domain decomposition for the solution of a quasilinear elliptic problem. *J. Comput. Appl. Math.*, 80(2):299–316, 1997.

[7] Chris J. Budd, Weizhang Huang, and Robert D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009.

[8] Xiao-Chuan Cai. Additive Schwarz algorithms for parabolic convection-diffusion equations. *Numer. Math.*, 60(1):41–61, 1991.

[9] Xiao-Chuan Cai. Multiplicative Schwarz methods for parabolic problems. *SIAM J. Sci. Comput.*, 15(3):587–603, 1994. Iterative methods in numerical linear algebra (Copper Mountain Resort, CO, 1992).

[10] Xiao-Chuan Cai and Maksymilian Dryja. Domain decomposition methods for monotone nonlinear elliptic problems. In *Domain decomposition methods in scientific and engineering computing (University Park, PA, 1993)*, volume 180 of *Contemp. Math.*, pages 21–27. Amer. Math. Soc., Providence, RI, 1994.

[11] Carl de Boor. Good approximation by splines with variable knots. In *Spline functions and approximation theory (Proc. Sympos., Univ. Alberta, Edmonton, Alta., 1972)*, pages 57–72. Internat. Ser. Numer. Math., Vol. 21. Birkhäuser, Basel, 1973.

[12] Carl de Boor. Good approximation by splines with variable knots. II. In *Conference on the Numerical Solution of Differential Equations*, volume 363 of *Lecture Notes in Mathematics*, pages 12–20. Springer Berlin / Heidelberg, 1974. 10.1007/BFb0069121.

[13] James W. Demmel. *Applied Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[14] Yana Di, Ruo Li, Tao Tang, and Pingwen Zhang. Moving mesh finite element methods for the incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 26(3):1036–1056, 2005.

[15] Maksymilian Dryja and Wolfgang Hackbusch. On the nonlinear domain decomposition method. *BIT*, 37(2):296–311, 1997.

[16] Martin J. Gander. A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations. *Numerical Linear Algebra with Applications*, 6:125–145, 1998.

[17] Martin J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.

[18] Martin J. Gander. Schwarz methods over the course of time. *Electron. Trans. Numer. Anal.*, 31:228–255, 2008.

[19] Martin J. Gander and Laurence Halpern. Absorbing boundary conditions for the wave equation and parallel computing. *Math. of Comp.*, 74(249):153–176, 2004.

[20] Martin J. Gander and Laurence Halpern. Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. *SIAM J. Numer. Anal.*, 45(2):666–697, 2007.

[21] Martin J. Gander, Laurence Halpern, and Frédéric Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. *SIAM Journal of Numerical Analysis*, 41(5):1643–1681, 2003.

[22] Martin J. Gander and Ronald D. Haynes. Domain decomposition approaches for mesh generation via the equidistribution principle. *SIAM Journal on Numerical Analysis*, 50(4):2111–2135, 2012.

[23] Martin J. Gander, Ronald D. Haynes, and Alexander J.M. Howse. Alternating and linearized alternating Schwarz methods for equidistributing grids, 2012. In Press, 8 pages.

[24] Martin J. Gander and Christian Rohde. Overlapping Schwarz waveform relaxation for convection-dominated nonlinear conservation laws. *SIAM J. Sci. Comput.*, 27(2):415–439, 2005.

[25] Martin J. Gander and Andrew M. Stuart. Space–time continuous analysis of waveform relaxation for the heat equation. *SIAM J. Sci. Comput.*, 19(6):2014–2031, 1998.

[26] Eldar Giladi and Herbert B. Keller. Space-time domain decomposition for parabolic problems. *Numer. Math.*, 93(2):279–313, 2002.

[27] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order.* Classics in Mathematics. Springer-Verlag, Berlin, 2001. Reprint of the 1998 edition.

[28] Edgar G. Goodaire and Michael M. Parmenter. *Discrete Mathematics with Graph Theory.* Pearson Prentice Hall, 2006.

[29] Ronald D. Haynes. Recent advances in Schwarz waveform relaxation moving mesh methods – a new moving subdomain method. In *Domain decomposition methods in science and engineering XIX*, volume 78 of *Lect. Notes Comput. Sci. Eng.*, pages 253–260. Springer, Berlin, 2011.

[30] Ronald D. Haynes and Alexander J.M. Howse. Alternating Schwarz methods for mesh equidistribution. *Electron. Trans. Numer. Anal.*, Submitted Oct 5, 2012.

[31] Ronald D. Haynes, Weizhang Huang, and Robert D. Russell. A moving mesh method for time-dependent problems based on Schwarz waveform relaxation. In *Domain decomposition methods in science and engineering XVII*, volume 60 of *Lect. Notes Comput. Sci. Eng.*, pages 229–236. Springer, Berlin, 2008.

[32] Ronald D. Haynes and Robert D. Russell. A Schwarz waveform moving mesh method. *SIAM J. Sci. Comput.*, 29(2):656–673, 2007.

[33] Weizhang Huang. Practical aspects of formulation and solution of moving mesh partial differential equations. *J. Comput. Phys.*, 171(2):753–775, 2001.

[34] Weizhang Huang, Yuhe Ren, and Robert D. Russell. Moving mesh methods based on moving mesh partial differential equations. *Journal of Computational Physics*, 113(2):279 – 290, 1994.

[35] Weizhang Huang, Yuhe Ren, and Robert D. Russell. Moving mesh partial differential equations (MMPDES) based on the equidistribution principle. *SIAM J. Numer. Anal.*, 31(3):709–730, 1994.

[36] Weizhang Huang and Robert D. Russell. A moving collocation method for solving time dependent partial differential equations. *Appl. Numer. Math.*, 20(1-2):101–116, 1996. Workshop on the method of lines for time-dependent problems (Lexington, KY, 1995).

[37] Weizhang Huang and Robert D. Russell. A high-dimensional moving mesh strategy. In *Proceedings of the International Centre for Mathematical Sciences Conference on Grid Adaptation in Computational PDEs: Theory and Applications (Edinburgh, 1996)*, volume 26, pages 63–76, 1998.

[38] Weizhang Huang and Robert D. Russell. Adaptive mesh movement — the MMPDE approach and its applications. *Journal of Computational and Applied Mathematics*, 128(1—2):383 – 398, 2001. ¡ce:title¿Numerical Analysis 2000. Vol. VII: Partial Differential Equations¡/ce:title¿.

[39] Weizhang Huang and Robert D. Russell. *Adaptive Moving Mesh Methods*, volume 174 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin, 2011.

[40] Weizhang Huang and David M. Sloan. A simple adaptive grid method in two dimensions. *SIAM J. Sci. Comput.*, 15:776–797, July 1994.

[41] R. Bruce Kellogg. A nonlinear alternating direction method. *Math. Comp.*, 23:23–27, 1969.

[42] Pierre-Louis Lions. On the Schwarz alternating method. I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987)*, pages 1–42. SIAM, Philadelphia, PA, 1988.

[43] Pierre-Louis Lions. On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 202–223. SIAM, Philadelphia, PA, 1990.

[44] Shiu-Hong Lui. On monotone and Schwarz alternating methods for nonlinear elliptic PDEs. *M2AN Math. Model. Numer. Anal.*, 35(1):1–15, 2001.

[45] Shiu-Hong Lui. On linear monotone iteration and Schwarz methods for nonlinear elliptic PDEs. *Numer. Math.*, 93(1):109–129, 2002.

[46] John A. Mackenzie. The efficient generation of simple two-dimensional adaptive grids. *SIAM J. Sci. Comput.*, 19(4):1340–1365 (electronic), 1998.

[47] Tarek P. A. Mathew. *Domain decomposition methods for the numerical solution of partial differential equations*, volume 61 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2008.

[48] Frédéric Nataf, Francois Rogier, and Eric de Sturler. Optimal interface conditions for domain decomposition methods. *CMAP (Ecole Polytechnique)*, 1994.

[49] James M. Ortega and Werner. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1970 original.

[50] Donald W. Peaceman and Henry H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3:28–41, 1955.

[51] Linda R. Petzold. A description of DASSL: a differential/algebraic system solver. In *Scientific computing (Montreal, Que., 1982)*, IMACS Trans. Sci. Comput., I, pages 65–68. IMACS, New Brunswick, NJ, 1983.

[52] Ali Reza Soheili and John M. Stockie. A moving mesh method with variable mesh relaxation time. *Applied Numerical Mathematics*, 58(3):249 – 263, 2008.

[53] Xue-Cheng Tai and Magne Espedal. Rate of convergence of some space decomposition methods for linear and nonlinear problems. *SIAM J. Numer. Anal.*, 35(4):1558–1570, 1998.

[54] Tao Tang. Moving mesh methods for computational fluid dynamics. In *Recent advances in adaptive computation*, volume 383 of *Contemp. Math.*, pages 141–173. Amer. Math. Soc., Providence, RI, 2005.

[55] Andrea Toselli and Olof B. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Springer series in computational mathematics, 34. Springer, 1 edition, November 2005.

[56] Arthur van Dam and Paul A. Zegeling. A robust moving mesh finite volume method applied to 1D hyperbolic conservation laws from magnetohydrodynamics. *J. Comput. Phys.*, 216(2):526–546, 2006.

[57] Arthur van Dam and Paul A. Zegeling. Balanced monitoring of flow phenomena in moving mesh methods. *Commun. Comput. Phys.*, 7(1):138–170, 2010.

[58] Baodong Zheng and Liancheng Wang. Spectral radius and infinity norm of matrices. *J. Math. Anal. Appl.*, 346(1):243–250, 2008.