Convergence analysis of substructuring Waveform Relaxation methods for space-time problems and their application to Optimal Control Problems

Mandal, Bankim

UNIVERSITÉ DE GENÈVE
Section de mathématiques

FACULTÉ DES SCIENCES
Martin J. Gander

Convergence Analysis of Substructuring Waveform Relaxation Methods for Space-time Problems and Their Application to Optimal Control Problems

Ph.D. THESIS

presented to the Faculty of Science, University of Geneva
for obtaning the degree of Doctor of Mathematics.

by

Bankim Chandra Mandal

of
India

Thèse N° 4745

Atelier de reproduction ReproMail de l'Université de Genève
Genève, 15 décembre, 2014

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

Doctorat ès sciences
Mention mathématiques

Thèse de ***Monsieur Bankim MANDAL***

intitulée :

"Convergence Analysis of Substructuring Waveform Relaxation Methods for Space-time Problems and Their Application to Optimal Control Problems"

La Faculté des sciences, sur le préavis de Monsieur M. GANDER, professeur ordinaire et directeur de thèse (Section de mathématiques), Monsieur F. KWOK, professeur et codirecteur de thèse (Section de mathématiques et Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, China), Monsieur E. HAIRER, professeur honoraire (Section de mathématiques) et Madame L. HALPERN, professeure (Universtié Paris 13, Laboratoire Analyse, Géométrie & Applications, Institut Galilée, Paris, France), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 15 décembre 2014

Thèse - 4745 -

Le Doyen

N.B. - La thèse doit porter la déclaration précédente et remplir les conditions énumérées dans les "Informations relatives aux thèses de doctorat à l'Université de Genève".

Abstract

This thesis contributes to develop a new class of methods for the numerical solution of partial differential equations (PDEs) using space-time domain decomposition algorithms to ensure the use of different time steps in different subdomains. It is motivated by the increasing demand of high resolution simulations of complex systems, as well as the increasing availability of computing clusters with thousands of cores.

We first introduce and analyze new types of Waveform Relaxation methods based on the Dirichlet-Neumann and Neumann-Neumann methods, for parabolic and hyperbolic problems. The algorithms, formally termed as Dirichlet-Neumann Waveform Relaxation (DNWR) and Neumann-Neumann Waveform Relaxation (NNWR), generalize the use of substructuring methods to the case of evolution problems in a natural way. Each of these methods is based on a non-overlapping spatial domain decomposition, and a series of subdomain problems is solved in space-time at each iteration with corresponding interface conditions, followed by a correction step. Both algorithms act as effective parallel solvers for space-time problems. Using Laplace transform techniques, we prove for the heat equation and for finite time intervals that, the DNWR and NNWR methods converge superlinearly for an optimal choice of the relaxation parameter. For any other choice of the relaxation parameter, convergence is only linear. In case of the wave equation, for a particular relaxation parameter, we show convergence of both the DNWR and NNWR methods in a finite number of steps for finite time intervals. But the number of steps depends on the subdomain size and the length of the time interval on which the algorithms are implemented. We illustrate the performance of the algorithms with numerical results, and show a comparison with classical and optimized Schwarz WR methods.

We finally propose an application of these methods for PDE-constrained Optimal Control Problems, solving the underlying forward and adjoint partial differential equations using a domain decomposition method. We apply and analyze the Dirichlet-Neumann and Neumann-Neumann methods on control problems and give the optimal choice of relaxation parameters for both the forward and adjoint problems in the steady as well as time-dependent case.

Keywords: Space-time domain decomposition, Waveform Relaxation, Dirichlet-Neumann Waveform Relaxation, Neumann-Neumann Waveform Relaxation, Optimized Schwarz Waveform Relaxation, Optimal Control Problems.

Résumé

Cette thèse contribue à développer une nouvelle classe de méthodes pour la résolution numérique d'équations aux dérivées partielles (EDP) en utilisant des méthodes de décomposition de domaine en espace-temps pour permettre l'utilisation de différents pas de temps dans les différents sous-domaines. Ceci est motivé par la demande croissante de simulations à haute résolution de systèmes complexes, ainsi que par la disponibilité croissante de clusters de calcul avec des milliers de cœurs.

Nous présentons et analysons de nouvelles méthodes de relaxation d'onde (WR) basées sur des méthodes de Dirichlet-Neumann et Neumann-Neumann pour des problèmes paraboliques et hyperboliques. Les algorithmes, officiellement désignées comme 'Dirichlet-Neumann Waveform Relaxation' (DNWR) et 'Neumann-Neumann Waveform Relaxation' (NNWR), généralisent de manière naturelle l'utilisation de méthodes de sous-structuration pour des problèmes d'évolution. Chacune de ces méthodes est basée sur une décomposition de domaine spatial sans chevauchement, et l'itération consiste à résoudre des sous-domaines dans l'espace-temps avec l'état correspondant de l'interface, suivie d'étape de correction. Les deux algorithmes agissent comme de solveurs parallèles efficaces pour les problèmes d'espace-temps. Nous démontrons en utilisant les techniques de la transformée de Laplace appliquée à l'équation de la chaleur que les méthodes de DNWR et NNWR convergent superlinéairement pour un choix optimal du paramètre de relaxation et pour des intervalles de temps finis. La convergence est linéaire pour tout autre choix du paramètre de relaxation. Dans le cas d'équation d'onde, nous prouvons la convergence à la fois du DNWR et NNWR méthodes dans un nombre fini d'itérations pour un paramètre de relaxation particulier et pour des intervalles de temps finis. Le nombre d'itérations dépend de la taille des sous-domaines et la longueur d'intervalle de temps sur lequel les algorithmes sont utilisés. Nous illustrons la performance des algorithmes avec des résultats numériques et nous montrons une comparaison avec les méthodes classiques et optimisés Schwarz WR.

Nous proposons enfin une application des méthodes pour les problèmes de contrôle optimal avec contraintes PDE. Nous résolvons les équations aux dérivées partielles correspondant et adjointes en utilisant une méthode de décomposition de domaine. Nous appliquons et analysons les méthodes de Dirichlet-Neumann et Neumann-Neumann sur les problèmes de contrôle et donnons le choix optimal des paramètres de relaxation à la fois pour les problèmes correspondants et adjointes, pour des problèmes elliptiques et depandants du temps.

Mots-clés: Décomposition de domaines espace–temps, Relaxation d'Onde, Relaxation d'Onde de Dirichlet-Neumann, Relaxation d'Onde de Neumann-Neumann, Relaxation d'Onde de Schwarz Optimisée, Problèmes de contrôle optimal.

To my *Maa*, *Baba*, *Dada & Madhuparna*

Acknowledgements

It is a great pleasure to express my gratitude and everlasting indebtedness to my research supervisor, Prof. Dr. Martin Jacob Gander. I am grateful to him for his advice and continuous encouragement throughout my research work. Without his active guidance, it would not have been possible for me to complete this work.

I also wish to express my gratitude to my co-supervisor Prof. Dr. Felix Kwok, who provided me with most important academic inputs to carry out my doctoral study. I acknowledge him as a mentor for offering his immense knowledge at every stage of my research. Most importantly, he always gave moral support and showed trust in my work.

In addition, my stay at the Mathematics section has been smoothened by a series of meetings that have had their deep impact on me. I would like to express my thankfulness to all the faculty members and all my fellow colleagues of the department for their valuable suggestions and encouragement. It also gives me great pleasure in acknowledging the secretaries of the department for making all the administrative works in one-go. I also want to add a token of gratitude to Pierre-Henri, Hui, Shu-Lin, Jérôme, Erwin and Soheil, who supported my effort with their help and knowledge, contributing to the quality of my thesis.

My eternal gratitude goes to my parents, Mrs. Rekha Mandal and Mr. Kartick Chandra Mandal, and other family members including my elder brother Dr. Iswar Chandra Mandal for their support and continuous motivation. I also thank my roommate Harekrishna Ghosh, who shared a lot of things with me to keep the tempo high. Special thanks to Prof. Krishnendra Shekhawat, Mrs. Komal Rathore, Dr. Pavel Chakraborty and Mrs. Kranti Biswas, for whom I feel at home while staying in Geneva at the beginning. I am privileged to mention the name of Swami Mahamedhananda for providing philosophical and spiritual strength throughout my career. I am also very grateful to my M.Sc. thesis supervisor Prof. S. Baskar from IIT Bombay for his support. Special thanks to my sister Corrin Cynthia and friends Abhishek, Mahesh, Sumit, Gouranga for their encouragement. Lastly happy to acknowledge my friend Madhuparna Sengupta, who is ever encouraging in all aspects of my life. In these acknowledgments, I could not unfortunately cite a large number of people that I met daily or occasionally.

Finally, I bow down before the Almighty for always being there!

Contents

1 Introduction

Our work is primarily focused on one of the active topics of recent years, namely space-time parallel methods for the numerical solution of partial differential equations (PDEs). This is driven by the increasing demand of high resolution simulation of complex systems, as well as the increasing availability of multiprocessor supercomputers with thousands of cores that help to reduce the computational cost significantly. In other words, it is important to use robust, fast and mathematically sound linear or nonlinear solvers that combine significant physics of the problem with efficient numerical techniques to achieve maximum possible efficiency and sufficient accuracy. One such attractive way of speeding up the computation is to parallelize the solution process using domain decomposition (DD) methods. In these methods, the computational domain is subdivided into several smaller subdomains. One then decouples the problem by making an initial guess along subdomain interfaces and solves the subdomain problems in parallel; then checks for discrepancies (e.g., non-smoothness) along subdomain interfaces at the end of each iteration, and continuously iterates the process until a smooth solution on the entire domain is achieved. These classical DD methods have their origin in the work of Schwarz [80], where he showed the existence of a solution to the Laplace equation using an iterative process on an irregular domain which is a combination of a rectangle and a disk. Later the concept was utilized and extended by Lions [57], and then by others in parallel computing to solve large-scale PDEs [76, 82]. Various other DD methods have been formulated since then to improve the performance of the classical DD method. Unlike the alternating Schwarz method [57, 58, 59], or the additive Schwarz method [20, 19, 21], which are formulated for overlapping subdomains, iterative substructuring methods are an important different class of methods, which are developed as non-overlapping domain decomposition algorithms. The convergence behavior of such methods is well understood for many types of boundary value problems, see [82, 76]. In particular, the Dirichlet-Neumann and the Neumann-Neumann methods belong to this class of substructuring methods for solving elliptic PDEs.

1.1 Dirichlet-Neumann method

This type of algorithm was first considered by Bjørstad & Widlund [11] and further studied in [14], [66] and [67]. The method is based on a non-overlapping domain decom-

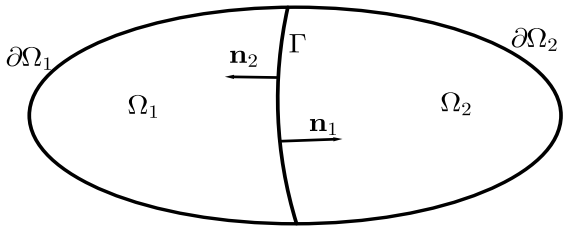


Figure 1.1.1: Splitting Ω into two disjoint subdomains

position in space, and the iteration requires subdomain solves with Dirichlet boundary conditions followed by subdomain solves with Neumann boundary conditions. We may look at the algorithm in the following way: suppose we want to solve the following Poisson equation in $\Omega \subset \mathbb{R}^d, d = 1, 2, 3$

$$\begin{aligned} -\triangle u &= f, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega. \end{aligned} \tag{1.1.1}$$

Then partitioning the domain Ω into two disjoint subdomains Ω_1, Ω_2 as in Figure 1.1.1, we can reformulate (1.1.1) according to the following equivalence result, see [76].

Theorem. *If u is a solution of the model problem (1.1.1) with $f \in L^2(\Omega)$, then for $i = 1, 2$, $u_i := u|_{\Omega_i}$ satisfies*

$$\begin{aligned} -\triangle u_i &= f, \quad \textit{in } \Omega_i, \\ u_i &= 0, \quad \textit{on } \partial\Omega_i \cap \partial\Omega, \end{aligned}$$

with the coupling conditions

$$u_1 = u_2 \quad \textit{and} \quad \partial_{\boldsymbol{n}_1} u_1 = -\partial_{\boldsymbol{n}_2} u_2, \quad \textit{on } \Gamma,$$

where $\boldsymbol{n}_i$ is the unit outward normal for Ω_i on the interface $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$.

Using this equivalence theorem one can introduce an iterative method to generate two sequences of functions $\left\{u_1^k\right\}_k, \left\{u_2^k\right\}_k$ starting from an initial guess u_2^0 as follows: for $k = 1, 2, \ldots$

$$\begin{cases} -\triangle u_1^k = f, & \text{in } \Omega_1, \\ u_1^k = 0, & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ u_1^k = u_2^{k-1}, & \text{on } \Gamma, \end{cases} \qquad \begin{cases} -\triangle u_2^k = f, & \text{in } \Omega_2, \\ \partial_{\boldsymbol{n}_2} u_2^k = -\partial_{\boldsymbol{n}_1} u_1^k, & \text{on } \Gamma, \\ u_2^k = 0, & \text{on } \partial\Omega_2 \cap \partial\Omega. \end{cases} \tag{1.1.2}$$

Such particular interface conditions explain the reason behind the name Dirichlet-Neumann (DN) method. If the sequences of the solutions to these subproblems converge, their limit will converge to u_1 and u_2 respectively, for references see [66, 67]. But the convergence is not always guaranteed. We plot the sequences $\left\{u_1^k\right\}_k, \left\{u_2^k\right\}_k$ from (1.1.2) in Figure 1.1.2 for the problem $-\frac{d^2u}{dx^2} = 0$ in $\Omega = (-3, 2)$, and check whether

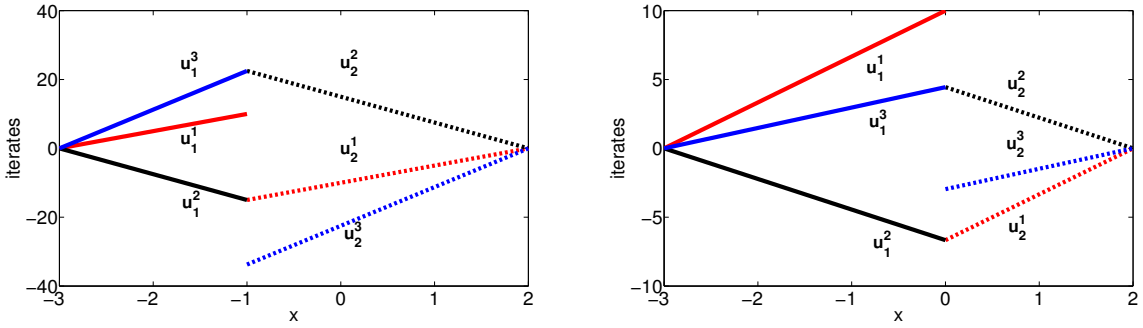


Figure 1.1.2: DN iterations: divergence for smaller Dirichlet subdomain Ω_1 on the left, and convergence for larger Dirichlet subdomain Ω_1 on the right

the iterative sequences converge to the zero solution. In the first experiment we take $\Omega_1 = (-3, -1), \Omega_2 = (-1, 2)$ and for the second experiment $\Omega_1 = (-3, 0), \Omega_2 = (0, 2)$.

A variant of this DN method can be set by replacing the Dirichlet condition in the first subdomain by $u_1^k = \lambda^{k-1}$ on Γ with λ^0 as an initial guess, and after each complete iteration the Dirichlet data is updated by a relaxation procedure,

$$\lambda^k = \theta \, u_2^k\big|_\Gamma + (1-\theta)\lambda^{k-1}.$$

Here a new variable $\theta \in (0, 1]$ is introduced and it is called the relaxation parameter. Convergence properties of such DN algorithm are studied for a particular model problem in Section 2.2 of Chapter 2. However, the performance of the algorithm for elliptic problems is now well understood, see for example the books [76, 82] and the references therein.

1.2 Neumann-Neumann method

Like the DN method, the Neumann-Neumann (NN) method is also a substructuring type algorithm for solving steady problems. The algorithm was first introduced by Bourgat et al. [13], see also [78] and [81]. One step of this method consists of solving the subdomain problems using Dirichlet interface conditions, then a correction step involving Neumann interface conditions. This method is analyzed for a model steady problem in Section 3.2 of Chapter 3. The NN method for the elliptic problems exhibit close similarity with FETI methods, which are among the most commonly used and well-understood DD methods in existence; see [23]. In addition, if a coarse grid is introduced, both NN and FETI methods can be scaled better for elliptic problems in the mesh size and in the number of subdomains: in [82] it is illustrated that the NN-preconditioned system matrix has a condition number proportional to $H^{-2}\left(1 + \log(H/h)\right)^2$ without coarse grid and to $\left(1 + \log(H/h)\right)^2$ for a coarse grid correction, where h is the mesh size and H is the size of the subdomains. Later, a technique called Balancing Domain Decomposition (BDD) was introduced for a decomposition into multiple subdomains, see in [63, 64]. This iterative method is in general aimed to solve a system of linear algebraic equations arising from the finite element method. It combines in each iteration the solution of local subproblems on non-overlapping subdomains with a coarse problem. For further details, see [76, 82].

On the other hand, when the models are described by time-dependent PDEs (evolution problems), the following three possible classes of DD techniques exist:

- *Discretization in time*: this approach consists of discretizing the problem uniformly in time with an implicit scheme to obtain a sequence of elliptic problems, which are in turn solved by DD methods; this is sometimes known as Rothe's method, in honor of the German analyst Erich Rothe. For this kind of technique, we refer to [15, 16]. One of the shortcomings of this approach is that it is by nature sequential in time. It means, one cannot obtain the solution of a later time step before solving the underlying problem at the earlier time steps. Another drawback is that, uniform time step across the whole domain need to be enforced, which is very restrictive for problems with variable coefficients or multiple time scales. So this method is expensive for parallel computation, since one needs to exchange information at each time step of the discretization.

- *Discretization in space*: in this approach the equation is first discretized in space, which is called the method of lines, and then one applies a waveform relaxation algorithm to solve the large system of ordinary differential equations (ODEs) obtained from the space-discretization process. Multigrid dynamic iteration [60, 49] and multi-splitting algorithms [50] are some particular examples of this approach. Therefore the same limitation of exchanging information along the interfaces applies inherently in this case. Also, the connectivity of the subsystems in the large system of ODEs or the information about physical overlap are difficult to interpret for this type of method.

- *Space-time Domain Decomposition*: in contrast to the two classical techniques above, there exist space-time domain decomposition methods, formulated at the continuous level. Here, instead of discretizing in time or in space, one decomposes the original spatial domain into smaller subdomains and considers each subdomain problem as posed in both space and time; then the subproblems are solved iteratively communicating information at the interfaces between subdomains. This permits the use of different numerical methods in different subdomains, and saves communication time in parallel computing. At each iteration, one solves the space-time subproblem over the entire time interval of interest, before communicating interface data across subdomains. For this approach, see [40, 41, 26, 29, 65] for parabolic problems, and [32, 28, 30] for hyperbolic problems. In a practical note, the convergence of space-time domain decomposition methods can be improved further by introducing smaller time windows, where the long time interval is subdivided into several smaller subintervals, and the problem is then solved sequentially in those time windows.

The work in this thesis is focused on the last of these classes as it gives a natural way to efficiently deal with problems with strong heterogeneities. These types of methods are independent of the grid size. In contrast, the convergence rate depends mainly on which transmission conditions between subdomains are used. Appropriate transmission conditions can lead to convergence in few iterations, which is the most important criterion if the subdomain problems are expensive to solve. However, these optimal transmission conditions are highly problem-dependent and are known only for certain model problems. They are also expensive to compute in general. So our focus here is

to search for space-time DD methods having relatively cheap transmisson conditions, and to get, if not better, similar convergence rates as like the known methods. In this thesis, we systematically introduce and analyze two new types of space-time domain decomposition methods to solve general space-time problems. These iterative algorithms are also time parallel methods, and they are categorized under the name Waveform Relaxation (WR).

1.3 Waveform Relaxation methods

WR methods have their origin in the work of Picard [74] and Lindelöf [55] in the late 19th century. Picard considered systems of ODEs of the form

$$\frac{dy}{dx} = f(x, y), \text{ with the initial condition } y(0) = y_0,$$

and proved existence of solutions using a successive approximation method. In this method a vector y^n, $n = 1, 2, \ldots$ is defined by

$$\frac{dy^n}{dx} = f(x, y^{n-1}), \text{ with } y^n(0) = y_0,$$

with y^0 as an initial guess. After integrating both sides, the above equation takes the form

$$y^n(x) = y_0 + \int_0^x f(\tau, y^{n-1}(\tau))\, d\tau. \tag{1.3.1}$$

So the problem of solving the ODEs is transformed into a relatively easier problem that mainly deals with integration. Picard proved convergence of the sequence $\{y^n\}_n$ to show the existence of solutions for ODEs. Later Lindelöf proved the following convergence estimate:

Theorem. *If f is continuous with respect to its two variables, and uniformly Lipschitz with respect to y with Lipschitz constant L, then the sequence in (1.3.1) is convergent, and the error satisfies following superlinear estimate*

$$\|y^n - y\|_\infty \leq \frac{(LT)^n}{n!} \|y - y_0\|_\infty ,$$

where $\|\cdot\|_\infty$ is the supremum norm in the bounded interval $[0, T]$.

But this iterative method was not effective for practical computations. Much later in 1982 Lelarasmee, Ruehli and Sangiovanni-Vincentelli introduced WR as a practical parallel method for solving systems of ODEs originating from the analysis of large-scale integrated circuits in [54]:

'The Waveform Relaxation (WR) method is an iterative method for analyzing nonlinear dynamical systems in the time domain. The method, at each iteration, decomposes the system into several dynamical subsystems each of which is analyzed for the entire given time interval.'

We now summarize their idea for one of the original models (taken from [54]) of a MOS ring oscillator circuit plotted on the left of Figure 1.3.1. The behavior of the

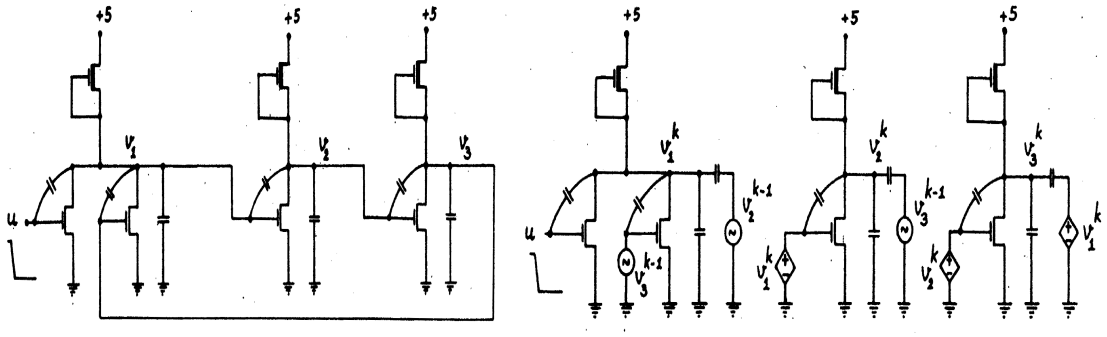


Figure 1.3.1: A MOS ring oscillator from [54] on the left, and the decomposition of the circuit on the right

circuit is governed by the following system of ODEs, formulated by treating the nodal voltages v_1, v_2, v_3 as variables

$$\frac{dv_1}{dt} = F_1\left(v_1, v_2, v_3\right), \frac{dv_2}{dt} = F_2\left(v_1, v_2, v_3\right), \frac{dv_3}{dt} = F_3\left(v_1, v_2, v_3\right),$$

with the intial state of the circuit as an initial condition. The authors decompose the circuit into subcircuits as shown in the right panel of Figure 1.3.1, and then use an iterative method, e.g. a Gauss-Seidal relaxation algorithm, to solve the system sequentially for $k = 1, 2, \ldots$

$$\frac{dv_1^k}{dt} = F_1\left(v_1^k, v_2^{k-1}, v_3^{k-1}\right), \frac{dv_2^k}{dt} = F_2\left(v_1^k, v_2^k, v_3^{k-1}\right), \frac{dv_3^k}{dt} = F_3\left(v_1^k, v_2^k, v_3^k\right),$$

together with the initial conditions $v_i^k(0) = v_i(0)$. One can also write the corresponding iterations for a Jacobi algorithm. The name of the method, Waveform Relaxation, comes from the fact that, the signals in circuit simulation are called 'waveforms'.

In the literature we find the following two convergence results for such WR method applied to ODEs:

- converge linearly on unbounded time intervals, provided some dissipation assumptions on the splitting; see [70, 71, 50, 68];
- converge superlinearly for nonlinear systems (also true for linear systems) on bounded time intervals, assuming a Lipschitz condition on the splitting function; see [70, 71, 3, 9].

Identification with domain decomposition methods for PDEs was made for hyperbolic systems by Bjørhus [10], and for the heat equation by Gander [25]. These algorithms are called Schwarz Waveform Relaxation, which is due to the domain decomposition with overlap as in the classical Schwarz method for steady problems, and iterative solves of time dependent problems like in waveform relaxation. Gander and Stuart showed for the heat equation in [40] a linear convergence of the overlapping Schwarz WR method on unbounded time intervals, with convergence rate depending on the size of the overlap. In [41] Giladi and Keller analyzed the convergence of the overlapping Schwarz WR method for the convection-diffusion equation and proved superlinear convergence

on bounded time intervals. But one needs to consider short time windows for faster convergence, since the convergence rate of WR algorithms in general is very slow. To address this issue, one can use optimized transmission conditions, which yield much faster algorithms. For such techniques, we refer to [29, 8] for parabolic problems, and [32, 28] for hyperbolic problems. This is also very similar for elliptic problems, where the introduction of optimized transmission conditions results to better convergence behavior. In fact, it is very natural to extend any domain decomposition method designed for elliptic problems into one for solving time-dependent problems based on the WR approach. As discussed earlier, such systematic extension of the classical Schwarz method to time-dependent problems was done independently in [40, 41].

However, no substructuring-type analogue of the WR method has been proposed so far. In this work, we propose the Dirichlet-Neumann Waveform Relaxation and the Neumann-Neumann Waveform Relaxation methods, which systematically generalize the use of substructuring methods to the case of time-dependent problems in a natural way. We define and analyze these methods in the continuous setting to understand the asymptotic behavior of the methods in the case of fine grids. Both methods solve the PDE over smaller subdomains over the whole time window at each iteration. So, these are of substructuring domain decomposition type, for the nature of domain decomposition, and of WR type due to the fact that they solve each subproblem for the whole time window. We develop two new space-time domain decomposition methods as follows:

i) The first method, named Dirichlet-Neumann Waveform Relaxation (DNWR), is of substructuring-type but global with respect to time in nature. More precisely, this is a Waveform Relaxation (WR) version of the Dirichlet-Neumann algorithm for solving space-time problems in parallel. It is based on a non-overlapping spatial domain decomposition, and each iteration involves subdomain solves in space time with corresponding interface condition and ends with a correction step. We apply this algorithm to both parabolic and hyperbolic problems, and using a Laplace transform argument, we analyze the convergence for one dimensional model problems. We prove for the heat equation that, the DNWR method converges superlinearly for a particular choice of the relaxation parameter in case of finite time intervals, and we show finite step convergence for the wave equation. The number of steps depends on the subdomain size and the length of the time interval on which the algorithms are implemented. One can also consider coarse grid corrections for parabolic problems to ensure a convergence rate independent of the number of subdomains.

ii) The second method is the Neumann-Neumann Waveform Relaxation (NNWR) method, which generalizes the Neumann-Neumann method for elliptic problems in a natural way. The first step of the Neumann-Neumann method consists of solving the subdomain problems using Dirichlet interface conditions, and then a correction step that takes Neumann boundary conditions along the interfaces. Now, each subproblem is in space-time, and the interface boundary conditions are also time-dependent. We introduce and analyze the algorithm at the continuous level, because it makes easier to understand the asymptotic behavior of the method for fine grids, without requiring any information about the discretisation schemes.

Parabolic initial value problems are in general highly non-normal in time, and existing techniques for estimating the condition number using abstract Schwarz theory [82] can not be employed to analyze such methods. Therefore we focus on the following:

- analyzing the convergence behavior of these two methods using Fourier-Laplace techniques for suitable model problems in 1D, and in 2D for simple geometric regions;
- illustrating the convergence behavior of the methods numerically for problems with spatially varying coefficients and more complex geometric regions.

Using the one and two dimensional heat equation as the model problems, we show that the NNWR method also converges superlinearly for finite time intervals. In addition, for two subdomain splitting we derive a linear bound which is also valid for unbounded time intervals. For 1D and 2D wave equations, we prove finite step convergence, even faster than the DNWR method, to the exact solution.

Certain parts of this thesis are covered in some recent publications. The results of the DNWR and NNWR algorithms for parabolic problems are presented in [62, 37, 36]. For the convergence results of these methods applied to hyperbolic problems, we refer to [35, 61, 36].

1.4 PDE-constrained Optimal Control Problems

Technically speaking, PDEs are not originated or solved for their own sake, but the solution is used to fulfill some other requirements. We focus here on some applications of the substructuring methods for the numerical solution of PDEs, both stationary and non-stationary, namely optimal control problems.

An optimal control problem generally consists of an objective or cost function to be minimized, a control variable and a state equation which relates the state variable for the control. There are numerous interesting instances where the state equations or the constraints are given by some PDEs, see [83, 47, 77]. Here we give some examples of optimal control problems subject to some linear elliptic or parabolic or hyperbolic PDEs.

Example 1. Stationary optimal heating problem.

Suppose a body Ω, as in Figure 1.4.1, is heated by a controlled heat source u (e.g. electromagnetic induction or microwaves) to reach the target temperature $\widetilde{y}$. If we assume that the temperature at the boundary vanishes, then the complete optimal control problem will be as follows:

$$\text{Minimize } J(y,u) = \tfrac{1}{2}\int_\Omega \left(y(x)-\widetilde{y}(x)\right)^2 dx + \tfrac{\lambda}{2}\int_\Omega u^2(x)dx,$$
$$\text{subject to } \begin{cases} -\text{div}\left(\kappa\nabla y(x)\right) = u(x), & x\in\Omega, \\ y(x) = 0, & x\in\partial\Omega, \end{cases}$$

where λ is a regularization parameter and κ is the thermal conductivity of the body. Similar kind of elliptic state equations are obtained in various other examples, such as control of current in a cathodic protection process or in scattering and radiation problems.

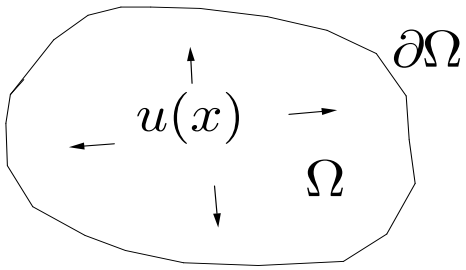


Figure 1.4.1: Spatial domain for the distributed control problem

Example 2. Transient optimal heating problem.

Now if the temperature changes with time in the above example, then the state equation is given by a heat equation. In that case, the temperature $y(x,t)$ and the heat source $u(x,t)$ depend on both space and time. Suppose our goal is to approximate the desired temperature $\widetilde{y}(x)$ at the final time T. Then the control problem consists in minimizing the objective functional

$$J(y,u) = \frac{1}{2}\int_\Omega \left(y(x,T) - \widetilde{y}(x)\right)^2 dx + \frac{\lambda}{2}\int_0^T \int_\Omega u^2(x,t)\, dxdt,$$

subject to the constraints:

$$\begin{cases} \partial_t y - \operatorname{div}(\kappa \nabla y) = u, & x \in \Omega, t \in (0,T), \\ y(x,0) = y_0(x), & x \in \Omega, \\ y(x,t) = g(x,t), & x \in \partial\Omega \times (0,T). \end{cases}$$

This is an example of a linear quadratic parabolic optimal control problem.

Example 3. Identification of a source of pollution.

Suppose a lake is polluted, and the problem is to find the source of pollution (unknown position q), that pollutes at a rate u. Then the concentration of pollutant $y(x,t)$ satisfies:

$$\begin{cases} \partial_t y - \nu\Delta y = u(t)\delta(x-q), & x \in \Omega, t \in (0,T), \\ y(x,0) = y_0(x), & x \in \Omega, \\ y(x,t) = g(x,t), & x \in \partial\Omega \times (0,T), \end{cases}$$

where δ is the dirac delta function given by

$$\delta(x-q) = \begin{cases} 0, & \text{if } x \neq q, \\ \infty, & \text{if } x = q. \end{cases}$$

Thus the optimal control problem is to find q which minimizes

$$\int_0^T \int_\Omega (y - \hat{y})^2\, dxdt.$$

This is also an example of linear quadratic parabolic optimal control problem.

Example 4. Optimal vibration problem.

Consider the problem of jumping on a river bridge by a group of people while crossing. Suppose $y(x,t)$ is the transversal displacement of the bridge and $u(x,t)$ is the vertical force acting on it. Then we get the following optimal control problem:

$$\text{Minimize } J(y,u) = \frac{1}{2}\int_0^T \int_\Omega (y(x,t) - \widetilde{y}(x,t))^2 \, dxdt + \frac{\lambda}{2}\int_0^T \int_\Omega u^2(x,t)\, dxdt,$$

subject to the constraints:

$$\begin{cases} \partial_{tt} y - \Delta y = u, & x \in \Omega, t \in (0,T), \\ y(x,0) = y_0(x), & x \in \Omega, \\ \partial_t y(x,0) = \bar{y}_0(x), & x \in \Omega, \\ y(x,t) = g(x,t), & x \in \partial\Omega \times (0,T). \end{cases}$$

The optimality conditions of these types of problems come from the theory of control problems with PDE constraints, which can be found in [56, 53, 83, 47, 77, 38]. These conditions yield a coupled problem consisting of two PDEs for steady problems, and for a time-dependent problem a system of equations containing the (forward) state equation and the (backward) adjoint equation which are coupled by an optimality condition. So, one needs to focus on the development of fast and effective tools for their numerical solution. DD methods are often used as powerful parallel computing tools for large-scale optimization problems governed by PDEs. For DD methods applied to linear-quadratic elliptic optimal control problems see [45, 46]. On the other hand, for time dependent problems, one can consider either time DD methods [52, 43] or spatial DD methods [44, 4, 12].

Heinkenschloss et al. [44] presented the numerical behavior of a non-overlapping spatial DD method for the solution of linear-quadratic parabolic optimal control problems, that arose in the determination of groundwater pollutant sources from measurements of concentrations, and identification of sources, or cooling processes in metallurgy. But there is to our knowledge no systematic statement and analysis of the DN and NN methods for optimal control problems in the literature, not even in the simple case of the heat equation in 1D or in regular 2D domains. In this thesis, we analyze the behavior of the DN and the NN methods for a model elliptic control problem, and show convergence in at most three iterations for a particular set of relaxation parameters. We also systematically introduce the DNWR and the NNWR algorithms for a model parabolic optimal control problem, and present relevant numerical results.

Main contribution of the thesis

Our work is the first of its kind to systematically introduce and analyze new types of Waveform Relaxation (WR) methods based on the substructuring methods, namely Dirichlet-Neumann and Neumann-Neumann algorithms for steady problems. However, for well-posedness of these methods, we refer to [48]. These newly proposed DNWR and NNWR algorithms are iterative methods, which enjoy all the advantages of a WR method for solving time-dependent PDEs on a parallel computer. Moreover, they are

based on non-overlapping domain decompositions that guarantee the use of different time steps in different subdomains. The main aspects covered in this thesis are as follows:

- a detailed analysis to show superlinear convergence of DNWR and NNWR methods for parabolic problems, for particular choices of the relaxation parameters;
- a detailed analysis to show that the DNWR and the NNWR algorithms are two-step convergence methods for hyperbolic problems, for particular choices of the relaxation parameters;
- a systematic comparison of the convergence results with other known WR methods, especially the Schwarz WR methods for heat and wave equations;
- analysis of both DN and NN methods, if applied to solve optimal control problems with steady equation as PDE-constraints, together with the detailed expressions of the relaxation parameters to achieve three-step convergence of both these methods. Numerical experiments for the DNWR and the NNWR methods applied to linear quadratic parabolic optimal control problems are presented.

2 Dirichlet-Neumann Waveform Relaxation Methods

2.1 Introduction

We introduce and analyze a waveform relaxation version of the Dirichlet-Neumann method for space-time problems. Like the Dirichlet-Neumann method for steady problems, the method is based on a non-overlapping spatial domain decomposition, and the iteration involves subdomain solves with Dirichlet boundary conditions followed by subdomain solves with Neumann boundary conditions. Each subdomain problem is now in space and time, and the interface conditions are also time-dependent. We formally call the algorithm Dirichlet-Neumann Waveform Relaxation (DNWR) method. We can identify this algorithm as a generalization of a substructuring method to time-dependent problems. We begin with the analysis of the Dirichlet-Neumann algorithm for a second order steady problem in Section 2.2. In Section 2.3, we extend this idea of Dirichlet-Neumann algorithm for Parabolic problems, and analyze for both two and multiple subdomains. These results can be found in [62, 37, 36]. Finally we present the convergence analysis for Hyperbolic problems in Section 2.4; see [35, 61].

2.2 Steady-state analysis

Suppose we want to solve the steady equation: $-\triangle u = f$ with boundary conditions $u = g$ with a Dirichlet-Neumann (DN) algorithm over a domain $\Omega = (-a, b)$. We divide Ω into two non-overlapping subdomains $\Omega_1 = (-a, 0)$ and $\Omega_2 = (0, b)$ to formulate the Dirichlet-Neumann algorithm: given an initial guess λ^0 along the interface $\{x = 0\}$, compute for $k = 1, 2, \ldots$

$$\begin{cases} -\triangle u_1^k = f, & \text{in } \Omega_1, \\ u_1^k(-a) = g(-a), \\ u_1^k(0) = \lambda^{k-1}, \end{cases} \qquad \begin{cases} -\triangle u_2^k = f, & \text{in } \Omega_2, \\ \partial_x u_2^k(0) = \partial_x u_1^k(0), \\ u_2^k(b) = g(b), \end{cases} \tag{2.2.1}$$

with the update condition

$$\lambda^k = \theta u_2^k(0) + (1 - \theta)\lambda^{k-1},$$

where $\theta \in (0,1]$ is a relaxation parameter. Now by linearity it suffices to consider only the error equations, $f = g = 0$. Solving both the error equations in (2.2.1) simultaneously, we obtain the above updating condition as

$$\lambda^k = \left(1 - \theta\frac{a+b}{a}\right)^k \lambda^0, \ k = 1, 2, \ldots. \tag{2.2.2}$$

Lemma 5 (Convergence of DN). *The Dirichlet-Neumann algorithm (2.2.1) converges linearly for* $0 < \theta < \min\left\{1, \frac{2a}{a+b}\right\}$, $\theta \neq \frac{a}{a+b}$. *For* $\theta = \frac{a}{a+b}$, *it converges in two iterations.*

Proof. From the equation (2.2.2) it is clear that the DN algorithm diverges if $\left|1 - \theta\frac{a+b}{a}\right| \geq 1$, i.e. if $\theta \geq \frac{2a}{a+b}$. The convergence is linear for $0 < \theta < \min\left\{1, \frac{2a}{a+b}\right\}$, $\theta \neq \frac{a}{a+b}$. If $\theta = \frac{a}{a+b}$, we have from (2.2.2) $\lambda^1 = 0$, and hence one more iteration produces the desired solution on the entire domain. □

Numerical Test:

The following figures in Figure 2.2.1 represent the convergence behavior of the Dirichlet-Neumann algorithm, applied to the model problem

$$-\triangle u = e^x, \quad u(-3) = 4, u(2) = 5,$$

over the domain $\Omega = (-3, 2)$. For the first experiment, we divide the domain into $(-3, 0)$ and $(0, 2)$, so that $a = 3, b = 2$ in Lemma 5, whereas we split the same domain Ω into $(-3, -1)$ and $(-1, 2)$ for the second experiment with $a = 2, b = 3$ in Lemma 5.

We now extend this algorithm for solving space-time problems in Section 2.3, where we define the DNWR algorithm for parabolic problems, and present a detailed analysis for the one dimensional heat equation. Finally we present the DNWR for hyperbolic problems in Section 2.4, and discuss the convergence analysis in detail for the one dimensional wave equation.

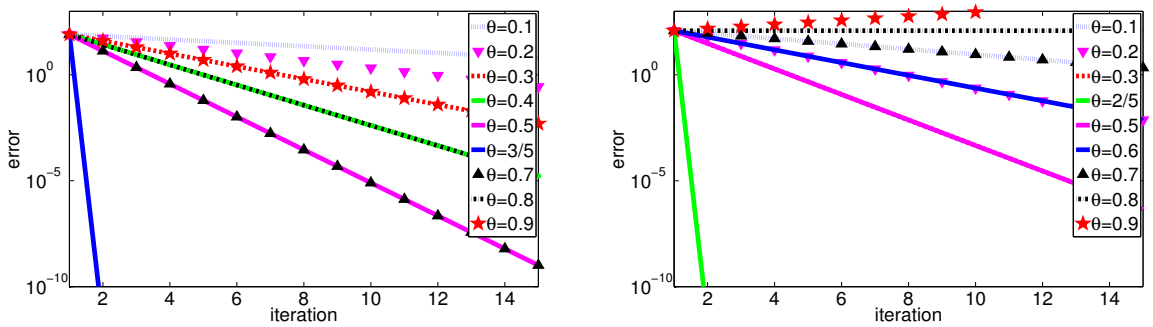


Figure 2.2.1: Convergence of DN using various relaxation parameters θ, on the left for $a > b$ and on the right for $a < b$

2.3 DNWR for Parabolic problems

We now formulate the new DNWR algorithm for the following model parabolic equation on a bounded domain $\Omega \subset \mathbb{R}^d$, $0 < t < T$, $d = 1, 2, 3$:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla \cdot (\kappa(\boldsymbol{x}, t)\nabla u) + f(\boldsymbol{x}, t), && \boldsymbol{x} \in \Omega,\ 0 < t < T, \\ u(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}), && \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}, t) &= g(\boldsymbol{x}, t), && \boldsymbol{x} \in \partial\Omega,\ 0 < t < T, \end{aligned} \tag{2.3.1}$$

where $\kappa_1 \geq \kappa(\boldsymbol{x}, t) \geq \kappa_2 > 0$. We first introduce the non-overlapping DNWR algorithm with two subdomains for the model problem (2.3.1), and we present convergence estimates for DNWR obtained for the special case of the one dimensional heat equation, $\kappa(\boldsymbol{x}, t) = 1$. We then generalize the DNWR algorithm to the multiple subdomains case, and present various possible arrangements in terms of chosen transmission conditions along the interfaces. For the numerically best possible arrangement, we present a convergence estimate for a one dimensional heat equation.

2.3.1 DNWR for two subdomains

To define the DNWR algorithm for the model problem (2.3.1) on the space-time domain $\Omega \times (0, T)$ with Dirichlet data given on $\partial\Omega$, we assume that the spatial domain Ω is partitioned into two non-overlapping subdomains Ω_1 and Ω_2, as illustrated on the left in Figure 2.2.1. We denote by u_i the restriction of the solution u of (2.3.1) to Ω_i, $i = 1, 2$, and by $\boldsymbol{n}_i$ the unit outward normal for Ω_i on the interface $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$.

The Dirichlet-Neumann Waveform Relaxation algorithm consists of the following steps: given an initial guess $h^0(\boldsymbol{x}, t)$ along the interface $\Gamma \times (0, T)$, compute for $k = 1, 2, \ldots$ the approximations

$$\begin{aligned} \partial_t u_1^k - \nabla \cdot \big(\kappa(\boldsymbol{x}, t)\nabla u_1^k\big) &= f, && \text{in } \Omega_1, \\ u_1^k(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_1, \\ u_1^k &= g, && \text{on } \partial\Omega_1 \setminus \Gamma, \\ u_1^k &= h^{k-1}, && \text{on } \Gamma, \end{aligned} \qquad \begin{aligned} \partial_t u_2^k - \nabla \cdot \big(\kappa(\boldsymbol{x}, t)\nabla u_2^k\big) &= f, && \text{in } \Omega_2, \\ u_2^k(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_2, \\ \partial_{\boldsymbol{n}_2} u_2^k &= -\partial_{\boldsymbol{n}_1} u_1^k, && \text{on } \Gamma, \\ u_2^k &= g, && \text{on } \partial\Omega_2 \setminus \Gamma, \end{aligned} \tag{2.3.2}$$

with the update condition along the interface

$$h^k(\boldsymbol{x}, t) = \theta u_2^k \big|_{\Gamma\times(0,T)} + (1 - \theta)h^{k-1}(\boldsymbol{x}, t), \tag{2.3.3}$$

$\theta \in (0, 1]$ being a relaxation parameter. The main goal of the analysis is to study how the error $h^{k-1}(\boldsymbol{x}, t) - u\big|_{\Gamma\times(0,T)}$ converges to zero, and by linearity it suffices to consider the so called error equations, $f(\boldsymbol{x}, t) = 0$, $g(\boldsymbol{x}, t) = 0$, $u_0(\boldsymbol{x}) = 0$ in (2.3.2), and examine how $h^k(\boldsymbol{x}, t)$ converges to zero as $k \to \infty$.

We now analyze the convergence for algorithm (2.3.2)-(2.3.3) for the special case of the heat equation, $\kappa(\boldsymbol{x}, t) = \nu$, on the one dimensional domain $\Omega = (-a, b)$, decomposed into $\Omega_1 = (-a, 0)$ and $\Omega_2 = (0, b)$ as shown on the right in Figure 2.3.1.

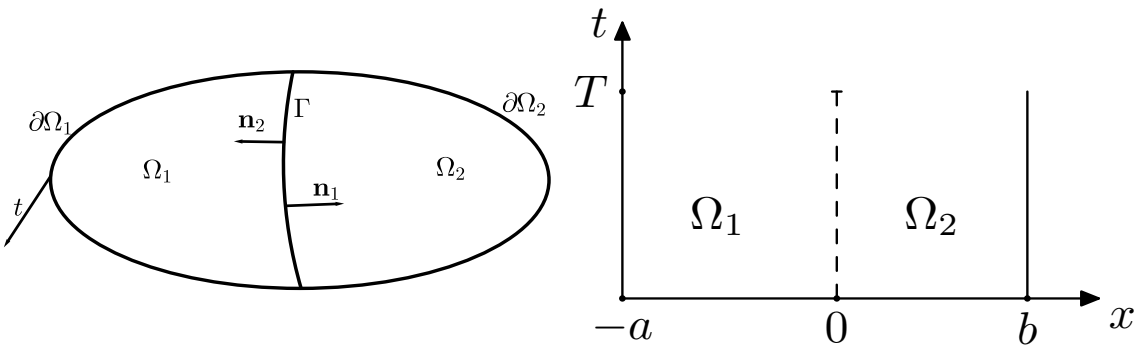


Figure 2.3.1: Splitting into two non-overlapping subdomains

2.3.1.1 Convergence analysis

Our convergence analysis is mainly based on the tools of Laplace transforms*. After a Laplace transform, the DNWR algorithm (2.3.2)-(2.3.3) for the error equations in the one dimensional heat equation setting becomes

$$\begin{aligned} (s-\nu\partial_{xx})\hat{u}_1^k &= 0, && \text{on } (-a,0), \qquad (s-\nu\partial_{xx})\hat{u}_2^k = 0, && \text{on } (0,b),\\ \hat{u}_1^k(-a,s) &= 0, && \partial_x\hat{u}_2^k(0,s) = \partial_x\hat{u}_1^k(0,s), \\ \hat{u}_1^k(0,s) &= \hat{h}^{k-1}(s), && \hat{u}_2^k(b,s) = 0, \end{aligned} \tag{2.3.4}$$

followed by the updating step

$$\hat{h}^k(s) = \theta\hat{u}_2^k(0,s) + (1-\theta)\hat{h}^{k-1}(s). \tag{2.3.5}$$

Solving the two-point boundary value problems in the Dirichlet and Neumann step in (2.3.4), we get

$$\hat{u}_1^k(x,s) = \frac{\hat{h}^{k-1}(s)}{\sinh(a\sqrt{s/\nu})}\sinh\left((x+a)\sqrt{s/\nu}\right), \tag{2.3.6}$$

$$\hat{u}_2^k(x,s) = \hat{h}^{k-1}(s)\frac{\coth(a\sqrt{s/\nu})}{\cosh(b\sqrt{s/\nu})}\sinh((x-b)\sqrt{s/\nu}). \tag{2.3.7}$$

By induction, we therefore find for the updating step the relation

$$\hat{h}^k(s) = \left(1-\theta-\theta\tanh(b\sqrt{s/\nu})\coth(a\sqrt{s/\nu})\right)^k\hat{h}^0(s), \quad k=1,2,3,\ldots \tag{2.3.8}$$

Theorem 6 (Convergence of DNWR for $a=b$)**.** *When the subdomains are of the same size, $a=b$ in (2.3.4)-(2.3.5), the DNWR algorithm converges linearly for $0<\theta<1$, $\theta\neq 1/2$. For $\theta=1/2$, it converges in two iterations. Convergence is independent of the time window size T.*

Proof. For $a=b$, equation (2.3.8) reduces to $\hat{h}^k(s) = (1-2\theta)^k\hat{h}^0(s)$, which has the simple back transform $h^k(t) = (1-2\theta)^k h^0(t)$. Thus the convergence is linear for $0<\theta<1$, $\theta\neq 1/2$. If $\theta=1/2$, we have $h^1(t)=0$, and hence one more iteration produces the desired solution on the entire domain. □

*For a formal discussion on Laplace transforms, see Appendix A

Having treated the simple case where the subdomains are of the same size, $a = b$, we focus now on the more interesting case where $a \neq b$. Defining

$$G(s) := \tanh(b\sqrt{s/\nu})\coth(a\sqrt{s/\nu}) - 1 = \frac{\sinh((b-a)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})\cosh(b\sqrt{s/\nu})}, \tag{2.3.9}$$

the recurrence relation (2.3.8) can be rewritten as

$$\hat{h}^k(s) = \begin{cases} (q(\theta) - \theta G(s))^k \hat{h}^0(s), & \theta \neq 1/2, \\ (-1)^k 2^{-k} G^k(s)\hat{h}^0(s), & \theta = 1/2, \end{cases} \tag{2.3.10}$$

where $q(\theta) = 1 - 2\theta$. Note that for $\mathrm{Re}(s) > 0$, $G(s)$ is $\mathcal{O}(s^{-p})$ for every positive p, which can be seen as follows: setting $s = re^{i\vartheta}$, we obtain for $a \geq b$ the bound

$$|s^p G(s)| \leq \left| \frac{s^p}{\cosh(b\sqrt{s/\nu})} \right| \leq \frac{2r^p}{\left| e^{b\sqrt{r/2\nu}} - e^{-b\sqrt{r/2\nu}} \right|} \to 0 \quad \text{as } r \to \infty,$$

and for $a < b$, we get the bound

$$|s^p G(s)| \leq \left| \frac{s^p}{\sinh(a\sqrt{s/\nu})} \right| \leq \frac{2r^p}{\left| e^{a\sqrt{r/2\nu}} - e^{-a\sqrt{r/2\nu}} \right|} \to 0 \quad \text{as } r \to \infty.$$

Therefore, by [18, p. 178], $G(s)$ is the Laplace transform of an infinitely differentiable function $F_1(t)$, which is the reason why we introduced $G(s)$ in (2.3.9). We now define

$$F_k(t) := \mathcal{L}^{-1}\left\{G^k(s)\right\}, \quad \text{for } k = 1, 2, 3, \ldots.$$

In what follows, we study the special case $\theta = 1/2$, when h^k is given by a convolution of h^0 with the analytic function F_k[†]. We also have to consider two cases: $a > b$, which means that the Dirchlet subdomain is bigger than Neumann subdomain, and $a < b$, when the Neumann subdomain is bigger than the Dirichlet subdomain. But before we proceed further with the main convergence results, we need some technical estimates of kernels arising in the Laplace transform of the DNWR algorithm. We present in the following section the precise estimates needed.

2.3.1.2 Kernel estimates

We start with several elementary properties of positive functions and their Laplace transforms. We define the *convolution* of two functions $g, w : (0, \infty) \to \mathbb{R}$ by

$$(g * w)(t) := \int_0^t g(t-\tau)w(\tau)d\tau.$$

Lemma 7. *Let g and w be two real-valued functions in $(0, \infty)$ with $\hat{w}(s) = \mathcal{L}\{w(t)\}$ the Laplace transform of w. Then for $t \in (0, T)$, we have the following properties:*

1. *If $g(t) \geq 0$ and $w(t) \geq 0$, then $(g * w)(t) \geq 0$.*

[†]For detailed expressions of these kernel, see Appendix B

2. $\|g * w\|_{L^1(0,T)} \leq \|g\|_{L^1(0,T)} \|w\|_{L^1(0,T)}$.

3. $\big|(g * w)(t)\big| \leq \|g\|_{L^\infty(0,T)} \int_0^T \big|w(\tau)\big| d\tau$.

4. $\int_0^t w(\tau)d\tau = (H * w)(t) = \mathcal{L}^{-1}\left(\frac{\hat{w}(s)}{s}\right)$, *$H(t)$ being the Heaviside step function.*

Proof. The proofs follow directly from the definitions[‡]. □

Lemma 8. *Let, $w(t)$ be a continuous and L^1-integrable function on $(0,\infty)$ with $w(t) \geq 0$ for all $t \geq 0$, and $\hat{w}(s) = \mathcal{L}\{w(t)\}$ be its Laplace transform. Then, for $\tau > 0$, we have the bound*

$$\int_0^\tau |w(t)|dt \leq \lim_{s\to 0+} \hat{w}(s). \tag{2.3.11}$$

Proof. With the definition of the Laplace transform[‡] and using positivity, we have

$$\begin{aligned}\int_0^\tau |w(t)|dt &= \int_0^\tau w(t)dt \leq \int_0^\infty w(t)dt = \int_0^\infty \lim_{s\to 0+} e^{-st} w(t)dt \\ &= \lim_{s\to 0+} \int_0^\infty e^{-st} w(t)dt = \lim_{s\to 0+} \hat{w}(s),\end{aligned}$$

where the dominated convergence theorem was used to exchange the order of limit and integration. □

In order to use Lemma 8 in our analysis, we have to show positivity of the inverse transforms of kernels appearing in the DNWR iteration. These results are established in the following lemma.

Lemma 9. *Let $\beta > \alpha \geq 0$ and s be a complex variable. Then, for $t \in (0,\infty)$*

$$\varphi(t) := \mathcal{L}^{-1}\left\{\frac{\sinh(\alpha\sqrt{s/\nu})}{\sinh(\beta\sqrt{s/\nu})}\right\} \geq 0 \quad \textit{and} \quad \psi(t) := \mathcal{L}^{-1}\left\{\frac{\cosh(\alpha\sqrt{s/\nu})}{\cosh(\beta\sqrt{s/\nu})}\right\} \geq 0.$$

Proof. We first prove that φ and ψ are well-defined and continous functions on $(0,\infty)$. Setting $s = re^{i\vartheta}$, a short calculation shows that for $\beta > \alpha \geq 0$ and for every positive p

$$\left|\frac{s^p \sinh(\alpha\sqrt{s/\nu})}{\sinh(\beta\sqrt{s/\nu})}\right| \leq r^p \cdot \left|\frac{e^{\alpha\sqrt{r/2\nu}} + e^{-\alpha\sqrt{r/2\nu}}}{e^{\beta\sqrt{r/2\nu}} - e^{-\beta\sqrt{r/2\nu}}}\right| \to 0 \qquad \text{as } r \to \infty,$$

so by [18, p. 178], its inverse Laplace transform exists and is continuous (in fact, infinitely differentiable). Thus, φ is a continuous function. A similar argument holds for ψ.

Next, we prove the positivity of φ and ψ by noting that these kernels are related to solutions of the heat equation. Let us consider the heat equation $u_t - \nu u_{xx} = 0$ on $(0,\beta)$ with initial condition $u(x,0) = 0$ and boundary conditions $u(0,t) = 0$, $u(\beta,t) = g(t)$. If g is non-negative, then by the maximum principle, this boundary value problem has a non-negative solution $u(\alpha,t)$ for all $\alpha \in [0,\beta]$, $t > 0$. Now performing a Laplace

[‡]For more details, see Appendix A

transform of the heat equation in time, we obtain the transformed solution along $x = \alpha$ to be

$$\hat{u}(\alpha, s) = \hat{g}(s)\frac{\sinh(\alpha\sqrt{s/\nu})}{\sinh(\beta\sqrt{s/\nu})} \quad \Longrightarrow \quad u(\alpha, t) = \int_0^t g(t-\tau)\varphi(\tau)d\tau.$$

We now prove that $\varphi(t) \geq 0$ by contradiction: suppose $\varphi(t_0) < 0$ for some $t_0 > 0$. Then by the continuity of φ, there exists $\delta > 0$ such that $\varphi(\tau) < 0$ for $\tau \in (t_0 - \delta, t_0 + \delta)$. Now for $t > t_0 + \delta$, we choose a non-negative g as follows:

$$g(\zeta) = \begin{cases} 1, & \zeta \in (t - t_0 - \delta, t - t_0 + \delta) \\ 0, & \text{otherwise.} \end{cases}$$

Then $u(\alpha, t) = \int_{t_0-\delta}^{t_0+\delta} g(t-\tau)\varphi(\tau)d\tau = \int_{t_0-\delta}^{t_0+\delta} \varphi(\tau)d\tau < 0$, which is a contradiction, and hence φ must be non-negative. To prove the result for ψ, we use again the heat equation $u_t - \nu u_{xx} = 0$, $u(x, 0) = 0$, but on the domain $(-\beta, \beta)$ and with boundary conditions $u(-\beta, t) = u(\beta, t) = g(t)$. Using a Laplace transform in time gives as solution at $x = \alpha$

$$\hat{u}(\alpha, s) = \hat{g}(s)\frac{\cosh(\alpha\sqrt{s/\nu})}{\cosh(\beta\sqrt{s/\nu})},$$

and hence a similar argument as in the first case proves that ψ is also non-negative. □

The following lemma contains specific estimates for the inverse Laplace transform of two kernels in terms of infinite sums.

Lemma 10. *For $k = 1, 2, 3, \ldots$, we have the identities*

$$\mathcal{L}^{-1}\left(\operatorname{cosech}^k(\alpha\sqrt{s/\nu})\right) = 2^k\sum_{m=0}^{\infty}\binom{m+k-1}{m}\frac{(2m+k)\alpha}{\sqrt{4\pi\nu t^3}}e^{-\frac{(2m+k)^2\alpha^2}{4\nu t}}, \quad (2.3.12)$$

$$\mathcal{L}^{-1}\left(\frac{\operatorname{cosech}^k(\alpha\sqrt{s/\nu})}{s}\right) = 2^k\sum_{m=0}^{\infty}\binom{m+k-1}{m}\operatorname{erfc}\left(\frac{(2m+k)\alpha}{2\sqrt{\nu t}}\right). \quad (2.3.13)$$

In particular, both functions are positive for $t > 0$.

Proof. Using that $\left|e^{-2\alpha\sqrt{s/\nu}}\right| < 1$ for $\operatorname{Re}(s) > 0$, we first expand cosech into an infinite binomial series,

$$\begin{aligned}\operatorname{cosech}^k(\alpha\sqrt{s/\nu}) &= \left(\frac{2}{e^{\alpha\sqrt{s/\nu}} - e^{-\alpha\sqrt{s/\nu}}}\right)^k = 2^k e^{-k\alpha\sqrt{s/\nu}}\left(1 - e^{-2\alpha\sqrt{s/\nu}}\right)^{-k} \\ &= 2^k\sum_{m=0}^{\infty}\binom{m+k-1}{m}e^{-(2m+k)\alpha\sqrt{s/\nu}}. \end{aligned} \quad (2.3.14)$$

Now using the inverse Laplace transform (see Oberhettinger [72])

$$\mathcal{L}^{-1}\left(e^{-\lambda\sqrt{s}}\right) = \frac{\lambda}{\sqrt{4\pi t^3}}e^{-\lambda^2/4t}, \quad \lambda > 0, \quad (2.3.15)$$

we obtain

$$\begin{aligned}\mathcal{L}^{-1}\left(\operatorname{cosech}^k(\alpha\sqrt{s/\nu})\right) &= 2^k\sum_{m=0}^{\infty}\binom{m+k-1}{m}\mathcal{L}^{-1}\left(e^{-(2m+k)\alpha\sqrt{s/\nu}}\right)\\ &= 2^k\sum_{m=0}^{\infty}\binom{m+k-1}{m}\frac{(2m+k)\alpha}{\sqrt{4\pi\nu t^3}}e^{-\frac{(2m+k)^2\alpha^2}{4\nu t}}. \quad (2.3.16)\end{aligned}$$

To justify taking the inverse Laplace transform term by term, we prove that the Laplace transform of the right-hand side of (2.3.16) indeed gives $\operatorname{cosech}^k(\alpha\sqrt{s/\nu})$. Let

$$f_m(t) = 2^k\binom{m+k-1}{m}\frac{(2m+k)\alpha}{\sqrt{4\pi\nu t^3}}\exp\left(-(2m+k)^2\alpha^2/4\nu t\right)$$

be the mth term of the series. Then for any real parameter $s_0 > 0$, we have for $\operatorname{Re}(s) > s_0$

$$\begin{aligned}\int_0^\infty |e^{-st}f_m(t)|\,dt &\le 2^k\int_0^\infty e^{-s_0 t}\binom{m+k-1}{m}\frac{(2m+k)\alpha}{\sqrt{4\pi\nu t^3}}e^{-(2m+k)^2\alpha^2/4\nu t}\,dt\\ &= 2^k\binom{m+k-1}{m}e^{-(2m+k)\alpha\sqrt{s_0/\nu}}.\end{aligned}$$

Thus, we have

$$\sum_{m=0}^{\infty}\int_0^\infty |e^{-st}f_m(t)|\,dt \le \operatorname{cosech}^k(\alpha\sqrt{s_0/\nu}) < \infty.$$

This allows us to use Fubini-Tonelli's theorem[§], where the product measure is between the discrete counting measure and the Lebesgue measure on $[0,\infty)$. We thus obtain for all $\operatorname{Re}(s) \ge s_0$

$$\int_0^\infty e^{-st}\sum_{m=0}^{\infty}f_m(t)\,dt = \sum_{m=0}^{\infty}\int_0^\infty e^{-st}f_m(t)\,dt = \operatorname{cosech}^k(\alpha\sqrt{s/\nu}).$$

The first identity (2.3.12) then follows by taking the inverse Laplace transform on both sides. Since each $f_m(t)$ is positive for $t > 0$, we conclude that the limit function $\sum_{m=0}^{\infty}f_m(t)$ is also positive.

For the second identity, we need the inverse Laplace transform from Lemma 45 of Appendix B

$$\mathcal{L}^{-1}\left(\frac{1}{s}e^{-\lambda\sqrt{s}}\right) = \operatorname{erfc}\left(\frac{\lambda}{2\sqrt{t}}\right), \quad \lambda > 0. \quad (2.3.17)$$

By dividing the expansion (2.3.14) by s, we obtain

$$\begin{aligned}\mathcal{L}^{-1}\left(\frac{\operatorname{cosech}^k(\alpha\sqrt{s/\nu})}{s}\right) &= 2^k\sum_{m=0}^{\infty}\binom{m+k-1}{m}\mathcal{L}^{-1}\left(\frac{1}{s}e^{-(2m+k)\alpha\sqrt{s/\nu}}\right)\\ &= 2^k\sum_{m=0}^{\infty}\binom{m+k-1}{m}\operatorname{erfc}\left(\frac{(2m+k)\alpha}{2\sqrt{\nu t}}\right), \quad (2.3.18)\end{aligned}$$

where we justify the interchanging of sums and inverse transforms in the same way as above. Since erfc is a positive function, so is the kernel (2.3.13). □

[§]see Appendix A

2.3.1.3 Convergence theorems

We have the following two main convergence results for the two subdomains case.

Theorem 11 (Convergence of DNWR for $a > b$)**.** *If $\theta = 1/2$ and the Dirichlet subdomain is larger than the Neumann subdomain, then the error of the DNWR algorithm (2.3.4)-(2.3.5) satisfies for $t \in (0, \infty)$ the linear convergence estimate*

$$\|h^k(\cdot)\|_{L^\infty(0,\infty)} \leq \left(\frac{a-b}{2a}\right)^k \|h^0(\cdot)\|_{L^\infty(0,\infty)}. \tag{2.3.19}$$

On a finite time interval $t \in (0, T)$, the DNWR method converges superlinearly with the estimate

$$\|h^k(\cdot)\|_{L^\infty(0,T)} \leq \left(\frac{a-b}{a}\right)^k \operatorname{erfc}\left(\frac{kb}{2\sqrt{\nu T}}\right) \|h^0(\cdot)\|_{L^\infty(0,T)}. \tag{2.3.20}$$

Proof. For $\theta = 1/2$ we get from (2.3.10) and using part 3 of Lemma 7

$$\left|h^k(t)\right| = \left|2^{-k}(-1)^k \left(h^0 * F_k\right)(t)\right| \leq 2^{-k}\|h^0\|_{L^\infty(0,T)} \int_0^T |F_k(\tau)|\, d\tau. \tag{2.3.21}$$

So in order to get an L^∞ convergence estimate, we need to bound $\int_0^T |F_k(\tau)|\, d\tau$. Now as $a > b$, we write $\mathcal{L}(-F_1(t)) = \frac{\sinh((a-b)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})} \cdot \frac{1}{\cosh(b\sqrt{s/\nu})}$. So by Lemma 8 and the fact that the convolution of two positive functions is positive, see Lemma 7 part 1, $-F_1(t)$ is positive. Thus, $(-1)^k F_k(t) \geq 0$ for all t, and we obtain from Lemma 8

$$\int_0^T \left|(-1)^k F_k(\tau)\right| d\tau \leq \lim_{s\to 0+} (-1)^k G^k(s) = \left(\frac{a-b}{a}\right)^k.$$

This bound is valid for arbitrary values of T, and hence we get from (2.3.21)

$$\|h^k(\cdot)\|_{L^\infty(0,\infty)} \leq \left(\frac{a-b}{2a}\right)^k \|h^0(\cdot)\|_{L^\infty(0,\infty)},$$

which shows that the algorithm is converging at least linearly for $a > b$. To prove the superlinear bound (2.3.20), we define $\hat{v}^k(s) := \cosh^k(b\sqrt{s/\nu})\hat{h}^k(s)$, and rewrite (2.3.10) for $\theta = 1/2$ as

$$\hat{v}^k(s) = 2^{-k}\frac{\sinh^k((a-b)\sqrt{s/\nu})}{\sinh^k(a\sqrt{s/\nu})}\hat{v}^0(s).$$

So if we set $g_k(t) := \mathcal{L}^{-1}\left(\frac{\sinh^k((a-b)\sqrt{s/\nu})}{\sinh^k(a\sqrt{s/\nu})}\right)$, then part 3 of Lemma 7 yields

$$\|v^k(\cdot)\|_{L^\infty(0,T)} \leq 2^{-k}\|v^0(\cdot)\|_{L^\infty(0,T)} \int_0^T |g_k(\tau)| d\tau.$$

By Lemma 8, $\int_0^T g_k(\tau)d\tau \leq \left(\frac{a-b}{a}\right)^k$, and we therefore obtain

$$\|v^k(\cdot)\|_{L^\infty(0,T)} \leq \left(\frac{a-b}{2a}\right)^k \|v^0(\cdot)\|_{L^\infty(0,T)}. \tag{2.3.22}$$

Setting $f_k(t) := \mathcal{L}^{-1}\left(\frac{1}{\cosh^k(b\sqrt{s/\nu})}\right)$ we have

$$h^k(t) = \left(f_k * v^k\right)(t) = \int_0^t f_k(t-\tau)v^k(\tau)d\tau,$$

from which it follows, using again part 3 of Lemma 7 that

$$\|h^k(\cdot)\|_{L^\infty(0,T)} \leq \|v^k(\cdot)\|_{L^\infty(0,T)} \int_0^T |f_k(\tau)|d\tau. \tag{2.3.23}$$

By Lemma 9, $f_k(t) \geq 0$ for all t. To obtain a bound for $\int_0^T f_k(\tau)\,d\tau$, we first show that the function $r_k(t) = \mathcal{L}^{-1}(2^k e^{-kb\sqrt{s/\nu}})$ is greater than or equal to $f_k(t)$ for all $t > 0$, and then bound $\int_0^T r_k(\tau)\,d\tau$ instead. Indeed, we have

$$\begin{aligned}
\mathcal{L}\left\{r_k(t) - f_k(t)\right\} &= 2^k e^{-kb\sqrt{s/\nu}} - \frac{2^k}{(e^{b\sqrt{s/\nu}} + e^{-b\sqrt{s/\nu}})^k} \\
&= \frac{2^k((1+e^{-2b\sqrt{s/\nu}})^k - 1)}{(e^{b\sqrt{s/\nu}} + e^{-b\sqrt{s/\nu}})^k} \\
&= \sum_{j=1}^{k} \binom{k}{j} e^{-2jb\sqrt{s/\nu}} \operatorname{sech}^k(b\sqrt{s/\nu}).
\end{aligned}$$

Note that in addition to $f_k(t) = \mathcal{L}^{-1}(\operatorname{sech}^k(b\sqrt{s/\nu}))$, $\mathcal{L}^{-1}(e^{-2jb\sqrt{s/\nu}})$ is also a positive function for $j = 1, \ldots, k$; see (2.3.15). Thus, $\mathcal{L}^{-1}(e^{-2jb\sqrt{s/\nu}}\operatorname{sech}^k(b\sqrt{s/\nu}))$ is a convolution of positive functions, and hence positive by part 1 of Lemma 7. This implies $r_k(t) - f_k(t) \geq 0$, so we deduce that

$$\int_0^T f_k(\tau)\,d\tau \leq \int_0^T r_k(\tau)\,d\tau = \mathcal{L}^{-1}\left(\frac{2^k e^{-kb\sqrt{s/\nu}}}{s}\right) = 2^k \operatorname{erfc}\left(\frac{kb}{2\sqrt{\nu T}}\right), \tag{2.3.24}$$

where we expressed the second integral as an inverse Laplace transform using Lemma 7, part 4, which we then evaluated using (2.3.17). Finally, we combine the bound above with (2.3.22) and (2.3.23) to obtain the second estimate of Theorem 11, which concludes the proof of this theorem. □

Theorem 12 (Convergence of DNWR for $a < b$)**.** *If $\theta = 1/2$ and the Dirichlet subdomain is smaller than the Neumann subdomain, then the error of the DNWR algorithm (2.3.4)-(2.3.5) satisfies for $t \in (0, \infty)$ the linear convergence estimate*

$$\|h^k(\cdot)\|_{L^\infty(0,\infty)} \leq \left(\frac{b-a}{2a}\right)^k \|h^0(\cdot)\|_{L^\infty(0,\infty)}. \tag{2.3.25}$$

For a finite time interval $t \in (0, T)$, the DNWR converges superlinearly with the estimate

$$\|h^{2k}(\cdot)\|_{L^\infty(0,T)} \leq \left(\frac{\sqrt{2}}{1 - e^{-\frac{(2k+1)a^2}{\nu T}}}\right)^{2k} e^{-k^2a^2/\nu T} \|h^0(\cdot)\|_{L^\infty(0,T)}. \tag{2.3.26}$$

Proof. For the case $0 < a < b$, we claim that $F_1(t)$ is positive. If $b - a \le a$, then we get the positivity by Lemma 9. If this is not the case, then take the integer m so that $ma < b \le (m+1)a$. Then, recursively applying the identity

$$\frac{\sinh((b-ja)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})} = \frac{\sinh((b-(j+1)a)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})}\cosh(a\sqrt{s/\nu}) + \cosh((b-(j+1)a)\sqrt{s/\nu})$$

for $j = 1, \dots, m-1$, we obtain

$$\frac{\sinh((b-a)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})\cosh(b\sqrt{s/\nu})} = \frac{\sinh((b-ma)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})} \cdot \frac{\cosh^{m-1}(a\sqrt{s/\nu})}{\cosh(b\sqrt{s/\nu})} + \sum_{j=0}^{m-2} \frac{\cosh^{j}(a\sqrt{s/\nu})\cosh\left((b-(j+2)a)\sqrt{s/\nu}\right)}{\cosh(b\sqrt{s/\nu})}.$$

Applying the binomial theorem to $\cosh\theta = \left(e^{\theta} + e^{-\theta}\right)/2$ we have the power-reduction formula

$$\cosh^n\theta = \begin{cases} \frac{2}{2^n}\sum_{l=0}^{\frac{n-1}{2}} \binom{n}{l} \cosh\left((n-2l)\theta\right), & n \text{ odd}, \\ \frac{1}{2^n}\binom{n}{n/2} + \frac{2}{2^n}\sum_{l=0}^{\frac{n}{2}-1} \binom{n}{l} \cosh\left((n-2l)\theta\right), & n \text{ even}, \end{cases}$$

so that we can write $\cosh^n\theta = \sum_{l=0}^{n} A_l^n \cosh(l\theta)$ with $\sum_{l=0}^{n} A_l^n = 1$ and $A_l^n \ge 0$. Therefore, we have

$$G(s) = \frac{\sinh((b-a)\sqrt{s/\nu})}{\cosh(b\sqrt{s/\nu})\sinh(a\sqrt{s/\nu})} = \frac{\sinh((b-ma)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})} \sum_{l=0}^{m-1} A_l^{m-1} \frac{\cosh(la\sqrt{s/\nu})}{\cosh(b\sqrt{s/\nu})} + \sum_{j=0}^{m-2}\sum_{l=0}^{j} \frac{A_l^j}{2} \left\{ \frac{\cosh\left((b-(j+l+2)a)\sqrt{s/\nu}\right)}{\cosh(b\sqrt{s/\nu})} + \frac{\cosh\left((b-(j-l+2)a)\sqrt{s/\nu}\right)}{\cosh(b\sqrt{s/\nu})} \right\},$$

where $\cosh^j\theta = \sum_{l=0}^{j} A_l^j \cosh(l\theta)$. Note that $b - ma \le a$, $(j-l+2)a \le ma < b$ and $|b-(j+l+2)a| < b$ for $0 \le j, l \le m-2$ and cosh is an even function. Thus by Lemma 9, each term in the above expression is the Laplace transform of a positive function. Hence $F_1(t)$ is positive, and therefore the convolution of k F_1 (i.e. $F_k(t)$) is also positive. We have $\lim_{s\to 0+} G(s) = \lim_{s\to 0+} \frac{\sinh((b-a)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})\cosh(b\sqrt{s/\nu})} = \frac{b-a}{a}$, and so by Lemma 8

$$\int_0^T \left|F_k(\tau)\right| d\tau = \int_0^T F_k(\tau)d\tau \le \lim_{s\to 0+} G^k(s) = \left(\lim_{s\to 0+} G(s)\right)^k = \left(\frac{b-a}{a}\right)^k. \tag{2.3.27}$$

The linear estimate follows by inserting the estimate (2.3.27) into (2.3.21).

For the second part, we rewrite (2.3.10) in the form

$$\hat{h}^{2k}(s) = \left(-\frac{1}{2}\right)^2 G(s)\hat{h}^{2k-1}(s) = \frac{1}{4}G^2(s)\hat{h}^{2k-2}(s). \tag{2.3.28}$$

Defining $\hat{\phi}(s) := \frac{\sinh^2((b-a)\sqrt{s/\nu})}{\cosh^2(b\sqrt{s/\nu})}$, $\hat{h}_1(s) = \frac{\cosh^2((b-a)\sqrt{s/\nu})}{\cosh^2(b\sqrt{s/\nu})}$ and $\hat{h}_2(s) = \frac{1}{\cosh^2(b\sqrt{s/\nu})}$, we can write

$$\begin{aligned} G^2(s) &= \frac{\sinh^2((b-a)\sqrt{s/\nu})}{\sinh^2(a\sqrt{s/\nu})\cosh^2(b\sqrt{s/\nu})} \\ &= \frac{1}{\sinh^2(a\sqrt{s/\nu})}\hat{\phi}(s) = \frac{1}{\sinh^2(a\sqrt{s/\nu})}\left(\hat{h}_1(s) - \hat{h}_2(s)\right). \end{aligned}$$

This motivates the definition of the new sequence $\hat{\vartheta}^{2k}(s) := \sinh^{2k}(a\sqrt{s/\nu})\hat{h}^{2k}(s)$, which from (2.3.28) satisfies the recurrence

$$\hat{\vartheta}^{2k}(s) = \frac{1}{4}\hat{\phi}(s)\hat{\vartheta}^{2k-2}(s).$$

Now using part 3 of Lemma 7, we obtain the estimate

$$\|\vartheta^{2k}(\cdot)\|_{L^\infty(0,T)} \leq \frac{1}{4}\|\vartheta^{2k-2}(\cdot)\|_{L^\infty(0,T)} \int_0^T |\phi(\tau)|d\tau, \tag{2.3.29}$$

and using Lemma 8 leads to

$$\begin{aligned} \int_0^T |\phi(\tau)|d\tau &\leq \int_0^T h_1(\tau)d\tau + \int_0^T h_2(\tau)d\tau \\ &\leq \lim_{s\to 0+} \frac{\cosh^2((b-a)\sqrt{s/\nu})}{\cosh^2(b\sqrt{s/\nu})} + \lim_{s\to 0+} \frac{1}{\cosh^2(b\sqrt{s/\nu})} = 2. \end{aligned}$$

By induction, we therefore obtain

$$\|\vartheta^{2k}(\cdot)\|_{L^\infty(0,T)} \leq \frac{1}{2^k}\|\vartheta^0(\cdot)\|_{L^\infty(0,T)} = \frac{1}{2^k}\|h^0(\cdot)\|_{L^\infty(0,T)}. \tag{2.3.30}$$

Now defining $\varphi_{2k}(t) = \mathcal{L}^{-1}\left(\frac{1}{\sinh^{2k}(a\sqrt{s/\nu})}\right)$, we have

$$h^{2k}(t) = \left(\varphi_{2k} * \vartheta^{2k}\right)(t) = \int_0^t \varphi_{2k}(t-\tau)\vartheta^{2k}(\tau)d\tau,$$

from which it follows by part 3 of Lemma 7 and using (2.3.30) that

$$\|h^{2k}(\cdot)\|_{L^\infty(0,T)} \leq \|\vartheta^{2k}(\cdot)\|_{L^\infty(0,T)} \int_0^T |\varphi_{2k}(\tau)|d\tau \leq B(k,T)\|h^0(\cdot)\|_{L^\infty(0,T)}, \tag{2.3.31}$$

where $B(k,T) := \frac{1}{2^k}\int_0^T |\varphi_{2k}(\tau)|d\tau$. By Lemma 10, $\varphi_{2k}(t) \geq 0$ for all $t > 0$. Thus by part 4 of Lemma 7, and then using equation (2.3.13) from Lemma 10, we get

$$\begin{aligned}
B(k,T) &= \frac{1}{2^k}\mathcal{L}^{-1}\left(\frac{\operatorname{cosech}^{2k}(a\sqrt{s/\nu})}{s}\right)_{t=T} \\
&= 2^k \sum_{m=0}^{\infty} \binom{m+2k-1}{m} \operatorname{erfc}\left(\frac{(m+k)a}{\sqrt{\nu T}}\right) \\
&\leq 2^k \sum_{m=0}^{\infty} \binom{m+2k-1}{m} \exp\left(-\frac{(m+k)^2a^2}{\nu T}\right) \\
&= 2^k e^{-k^2a^2/\nu T} \sum_{m=0}^{\infty} \binom{m+2k-1}{m} \exp\left(-\frac{(m^2+2km)a^2}{\nu T}\right) \\
&\leq 2^k e^{-k^2a^2/\nu T} \sum_{m=0}^{\infty} \binom{m+2k-1}{m} \exp\left(-\frac{m(2k+1)a^2}{\nu T}\right) \\
&\leq \left(\frac{\sqrt{2}}{1-e^{-\frac{2k+1}{\nu\sigma}}}\right)^{2k} e^{-k^2/\nu\sigma}, \quad \text{with } \sigma := T/a^2,
\end{aligned}$$

where we used for the first inequality the estimate

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}dt = \frac{2}{\sqrt{\pi}}\int_0^{\infty} e^{-(x+t)^2}dt \leq \frac{2}{\sqrt{\pi}}e^{-x^2}\int_0^{\infty} e^{-t^2}dt \leq e^{-x^2},$$

and for the third inequality

$$\frac{1}{(1-z)^{\beta}} = \sum_{m\geq 0}\binom{m+\beta-1}{m} z^m, \quad \text{for } |z| < 1.$$

Inserting the estimate for $B(k,T)$ into (2.3.31) gives then the result of the theorem. □

Remark 13. (a) The linear estimate (2.3.25) does not always imply convergence, because $b-a$ can be larger than $2a$. In other words, when $b > 3a$, i.e., when the Neumann subdomain is much larger than the Dirichlet one, it is not clear, at least from the expression of the estimate (2.3.25) whether the iteration converges to zero as $k \to \infty$. In this case, one should switch the interface conditions and solve a Dirichlet problem on the larger subdomain.

(b) Note that the factor multiplying $e^{-k^2a^2/\nu T}$ in the estimate (2.3.26) is an increasing function of k in general, since $\frac{\sqrt{2}}{1-e^{-\frac{2k+1}{\nu\sigma}}} > 1$. Thus, the bound (2.3.26) may increase initially for small iteration numbers k, before the factor $e^{-k^2a^2/\nu T}$ starts dominating and causing the bound to decrease to zero superlinearly. To estimate the turning point, let us fix an integer $l > 0$ and consider the behavior of the algorithm for iteration numbers

$k > 2l$. Then by writing $\alpha = e^{-l/\nu\sigma}$, we can bound $B(k,T)$ by

$$B(k,T) = \left(\frac{\sqrt{2}}{1-e^{-\frac{2k+1}{\nu\sigma}}}\right)^{2k} e^{-2kl/\nu\sigma} e^{-k(k-2l)/\nu\sigma}$$
$$\leq \left(\frac{\sqrt{2}e^{-l/\nu\sigma}}{1-e^{-\frac{2k}{\nu\sigma}}}\right)^{2k} e^{-k(k-2l)/\nu\sigma} \leq \underbrace{\left(\frac{\sqrt{2}\alpha}{1-\alpha^4}\right)^{2k}}_{=(*)} e^{-\frac{(k-2l)^2}{\nu\sigma}}.$$

Thus, if $\sqrt{2}\alpha/(1-\alpha^4) < 1$, then the factor $(*)$ is less than 1 and the bound $B(k,T)$ contracts superlinearly for $k > 2l$. This is true whenever $\alpha < \alpha_0$, where $\alpha_0 \approx 0.6095$ is the unique positive root of $\psi(\alpha) = \alpha^4 + \sqrt{2}\alpha - 1$. Hence, we get superlinear convergence for $k > 2l > 0.99\nu T/a^2$.

2.3.1.4 Numerical illustration

We perform experiments to measure the actual convergence rate of the discretized DNWR algorithm for the problem

$$\begin{array}{rcll} \partial_t u - \frac{\partial}{\partial x}\left(\kappa(x)\partial_x u\right) & = & 0, & x \in \Omega, \\ u(x,0) & = & x(x+1)(x+3)(x-2)e^{-x}, & x \in \Omega, \\ u(-3,t) = t,\ u(2,t) & = & te^{-t}, & t > 0. \end{array} \tag{2.3.32}$$

Note that in some of the experiments below, the diffusion coefficient $\kappa(x)$ will be spatially varying. This will allow us to study how spatially varying coefficients affect the performance of our algorithms, which have only been analyzed in the constant coefficient case. We discretize (2.3.32) using standard centered finite differences in space and backward Euler in time on a grid with $\Delta x = 2 \times 10^{-2}$ and $\Delta t = 4 \times 10^{-3}$ as follows: for $n \geq 1$ and $2 \leq i \leq J-1$,

$$u(x_i,t_n) := u_i^n \approx u_i^{n-1} + \frac{\Delta t}{\Delta x^2}\left[\kappa\left(x_{i+\frac{1}{2}}\right)\left(u_{i+1}^n - u_i^n\right) - \kappa\left(x_{i-\frac{1}{2}}\right)\left(u_i^n - u_{i-1}^n\right)\right],$$

where $t_n = n\Delta t$ and $x_i = x_1 + (i-1)\Delta x$ for $i = 2, \ldots, J-1$ with $\Delta x = (x_J - x_1)/(J-1)$ and $\kappa\left(x_{i+\frac{1}{2}}\right)$ is calculated at $x = (x_i + x_{i+1})/2$. For the DNWR method, we consider

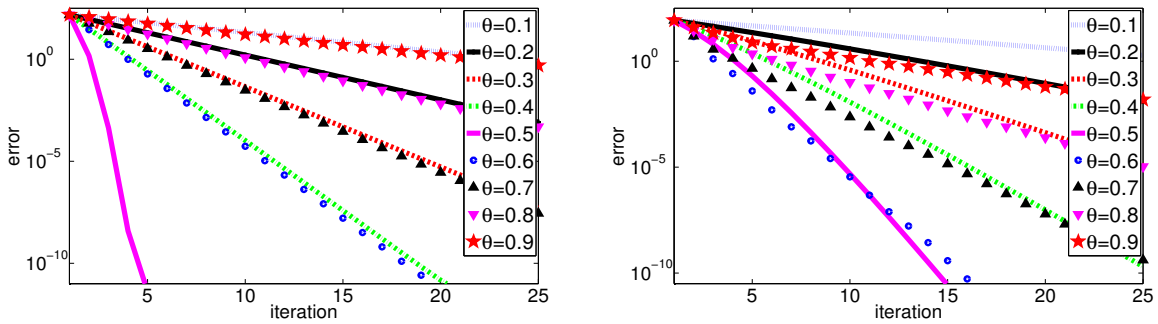


Figure 2.3.2: Convergence of DNWR for $a > b$ using various relaxation parameters θ for $T = 2$, on the left for $\kappa(x) = 1$ and on the right for $\kappa(x) = 1 + e^x$

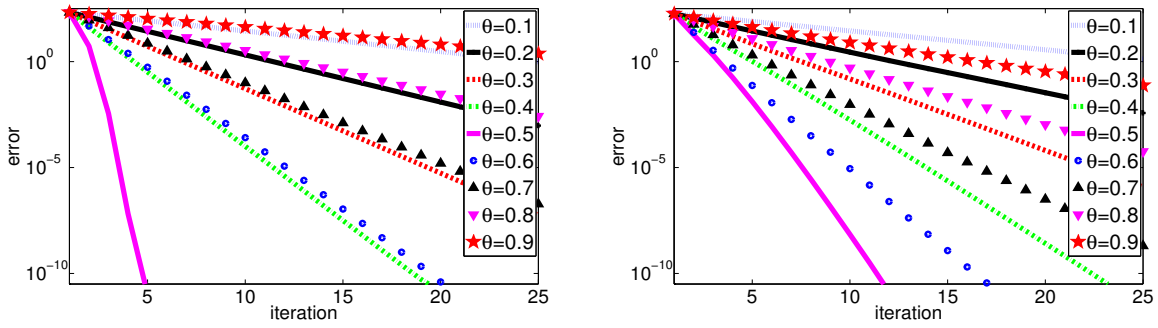


Figure 2.3.3: Convergence of DNWR for $a < b$ using various relaxation parameters θ for $T = 2$, on the left for $\kappa(x) = 1$ and on the right for $\kappa(x) = 1 + e^x$

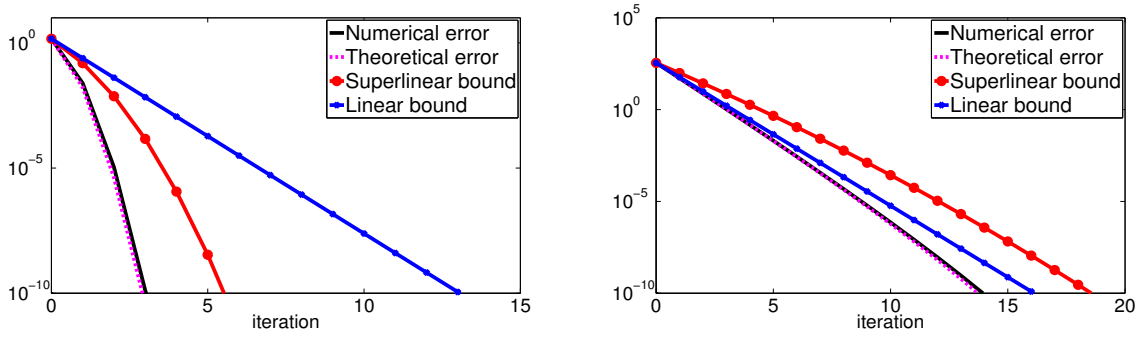


Figure 2.3.4: Comparison of the numerically measured convergence rates and the theoretical error estimates for DNWR for $\kappa(x) = 1$ with $T = 2$ on the left, and $T = 50$ on the right

two cases: first, we choose $a = 3$ and $b = 2$, i.e., we split the spatial domain $\Omega := (-3, 2)$ into two non-overlapping subdomains $\Omega_1 = (-3, 0)$ and $\Omega_2 = (0, 2)$, see Figure 2.3.1. This is the case of DNWR when the Dirichlet subdomain is larger than the Neumann subdomain ($a > b$), corresponding to Theorem 11. For the second case, we take $a = 2$ and $b = 3$, so that the Dirichlet domain is smaller than the Neumann one, as in Theorem 12. We test the algorithm by choosing $h^0(t) = t^2, t \in [0, T]$ as an initial guess. Figures 2.3.2 and 2.3.3 give the convergence curves for $T = 2$ and for different values of the parameter θ for $\kappa(x) = 1$ on the left, and $\kappa(x) = 1+e^x$ on the right. We see that for this reasonably small time window, we get linear convergence for all relaxation parameters θ, except for $\theta = 1/2$, when we observe superlinear convergence. This is the case regardless of whether the diffusion coeffcient varies spatially.

We now compare the numerical behavior of DNWR with our theoretical estimates in Section 2.3.1.3. In Figure 2.3.4, we show for the DNWR algorithm a comparison between the numerically measured convergence for the discretized problem, the theoretical convergence for the continuous model problem (calculated using inverse Laplace transforms from Appendix B), and the linear and superlinear convergence estimates shown in Theorem 11, for $a = 3$, $b = 2$, $\kappa(x) = 1$. We see that for a short time interval, $T = 2$, the algorithm converges superlinearly, and the superlinear bound is quite accurate. For the long time interval $T = 50$, the algorithm converges linearly, and the

linear convergence estimate is now more accurate.

2.3.2 DNWR for multiple subdomains

In this subsection we generalize the DNWR algorithm to multiple subdomains in one spatial dimension. We present different possible arrangements (in terms of placing Dirichlet and Neumann boundary conditions) and with a numerical implementation of these arrangements for a model problem we determine the best possible one. We then analyze the Dirichlet-Neumann method for this arrangement.

2.3.2.1 Motivation

Suppose we want to solve the 1D heat equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= \Delta u, && x \in \Omega,\ 0 < t < T, \\ u(x,0) &= u_0(x), && x \in \Omega, \\ u(x,t) &= g(x,t), && x \in \partial\Omega,\ 0 < t < T, \end{aligned} \tag{2.3.33}$$

using the DNWR method. The spatial domain $\Omega = (0,5)$ is decomposed into five non-overlapping subdomains $\Omega_i = (x_{i-1}, x_i), i = 1, \ldots, 5$, see Figure 2.3.5(a), with three possible combinations of boundary conditions along the interfaces, Figures 2.3.5(b), 2.3.6(a), 2.3.6(b). D in blue denotes the Dirichlet condition along the two physical boundaries, whereas D and N in red denote the Dirichlet and Neumann boundary conditions along the interfaces. We are given Dirichlet traces $\{g_i^0(t)\}_{i=1}^4$ as initial guesses along the interfaces $\{x_i\}_{i=1}^4$.

First arrangement (A1):

We extend the two subdomain-formulation to many subdomains in a natural way, see Figure 2.3.5(b). With the intial guesses, a Dirichlet subproblem is solved in the first subdomain Ω_1, followed by a series of mixed Neumann-Dirichlet subproblem solves in the subsequent subdomains $(\Omega_i, i = 2, \ldots, 5)$, exactly like in the two-subdomain case. Thus the Dirichlet-Neumann Waveform Ralaxation algorithm is given by: for $k = 1, 2 \ldots$ and for $\theta \in (0, 1]$ compute

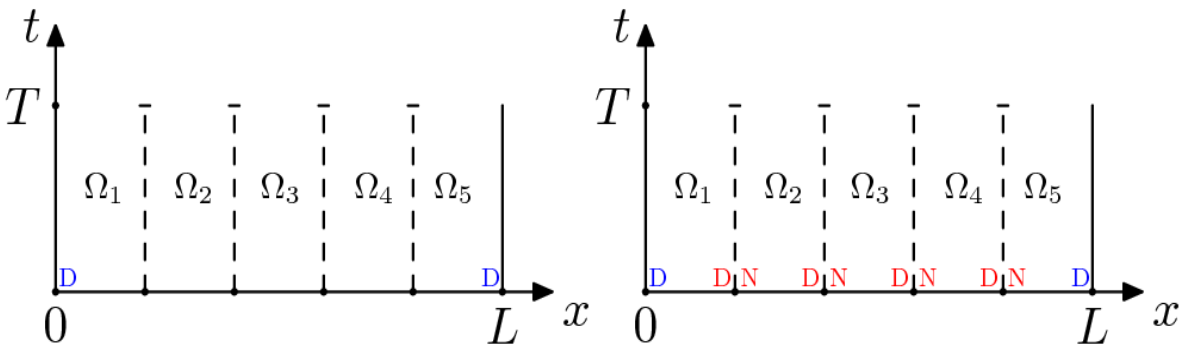
$$\begin{aligned} \partial_t u_1^k - \partial_{xx} u_1^k &= 0, && \text{in } \Omega_1, t > 0, \\ u_1^k(x,0) &= u_0(x), && \text{in } \Omega_1, \\ u_1^k(0,t) &= g(0,t), && t > 0, \\ u_1^k(x_1,t) &= g_1^{k-1}(t), && t > 0, \end{aligned}$$

and for $i = 2, \ldots, 5$

$$\begin{aligned} \partial_t u_i^k - \partial_{xx} u_i^k &= 0, && \text{in } \Omega_i, t > 0, \\ u_i^k(x,0) &= u_0(x), && \text{in } \Omega_i, \\ \partial_x u_i^k(x_{i-1},t) &= \partial_x u_{i-1}^k(x_{i-1},t), && t > 0, \\ u_i^k(x_i,t) &= g_i^{k-1}(t), && t > 0, \end{aligned}$$

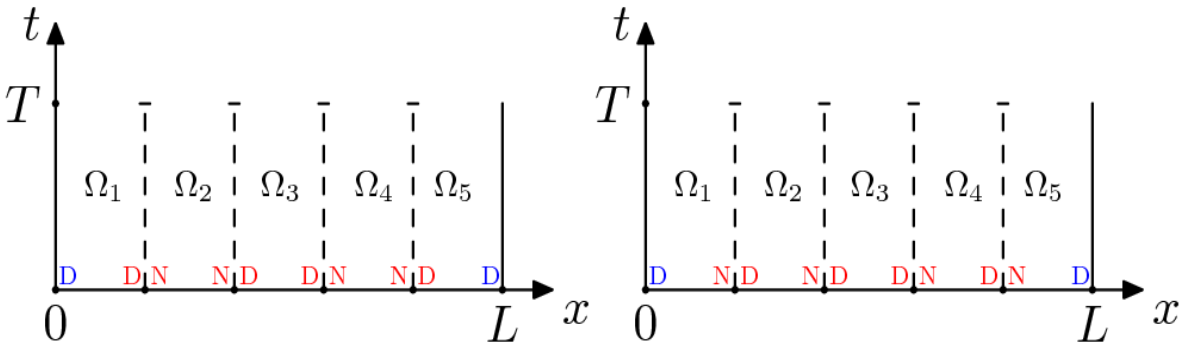
with $g_5^k(t) = g(5,t)$ for the last subdomain along the physical boundary. The updated interface values for the next step are then defined as

$$g_i^k(t) = \theta u_{i+1}^k(x_i,t) + (1-\theta) g_i^{k-1}(t).$$



(a) (b) A1

Figure 2.3.5: Decomposition of the domain, and one of the arrangements of boundary conditions



(a) A2 (b) A3

Figure 2.3.6: Other two arrangements of boundary conditions

We now discretize (2.3.33) using standard centered finite differences in space and backward Euler in time, and solve the equation numerically using the above algorithm for different time windows. For the test we choose $u_0(x) = 0, g(x,t) = (x+1)t, g_i^0(t) = t^2, t \in [0,T]$. Figure 2.3.7 gives the convergence curves for different values of the parameter θ for $T = 2$ on the left, and $T = 20$ on the right.

Second arrangement (A2):

This is the well-known red-black block formulation, described in Figure 2.3.6(a). In this arrangement, we solve a Dirichlet subproblem and a Neumann subproblem in alternating fashion. Given initial Dirichlet traces along the interfaces, a series of Dirichlet subproblems is first solved in parallel in alternating subdomains $(\Omega_1, \Omega_3, \Omega_5)$, and then a series of Neumann subproblems is solved in the remaining subdomains (Ω_2, Ω_4). So this type of Dirichlet-Neumann Waveform Ralaxation algorithm is given by: for $k = 1, 2 \ldots$

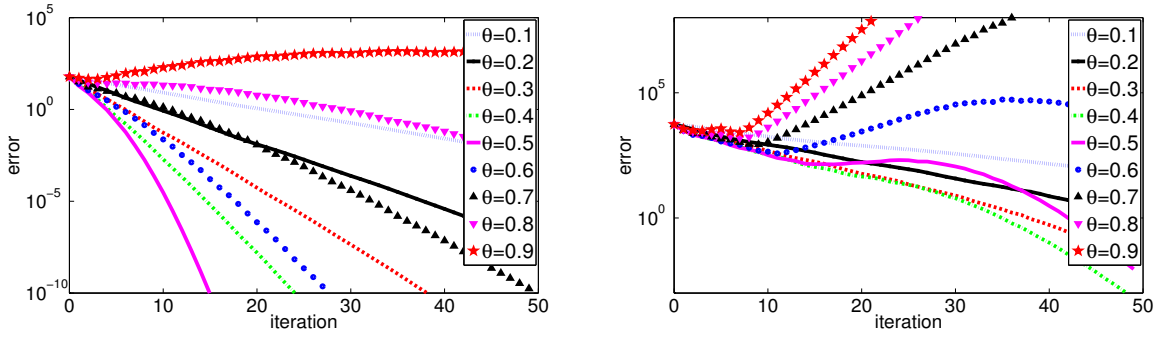


Figure 2.3.7: Convergence of multi-subdomain DNWR for various relaxation parameters θ, on the left for $T = 2$ and on the right for $T = 20$

compute for $i = 1, 3, 5$

$$\begin{aligned} \partial_t u_i^k - \partial_{xx} u_i^k &= 0, && \text{in } \Omega_i, t > 0, \\ u_i^k(x,0) &= u_0(x), && \text{in } \Omega_i, \\ u_i^k(x_{i-1},t) &= g_{i-1}^{k-1}(t), && t > 0, \\ u_i^k(x_i,t) &= g_i^{k-1}(t), && t > 0, \end{aligned}$$

with $g_0^k(t) = g(0,t), g_5^k(t) = g(5,t)$, and for $i = 2, 4$

$$\begin{aligned} \partial_t u_i^k - \partial_{xx} u_i^k &= 0, && \text{in } \Omega_i, t > 0, \\ u_i^k(x,0) &= u_0(x), && \text{in } \Omega_i, \\ \partial_x u_i^k(x_{i-1},t) &= \partial_x u_{i-1}^k(x_{i-1},t), && t > 0, \\ \partial_x u_i^k(x_i,t) &= \partial_x u_{i+1}^k(x_i,t), && t > 0, \end{aligned}$$

together with the updating conditions

$$\begin{aligned} g_i^k(t) &= \theta u_{i+1}^k(x_i,t) + (1-\theta) g_i^{k-1}(t), \ i = 1, 3, \\ g_i^k(t) &= \theta u_i^k(x_i,t) + (1-\theta) g_i^{k-1}(t), \ i = 2, 4, \end{aligned}$$

where $\theta \in (0, 1]$ is a relaxation parameter.

We now implement this version of DNWR algorithm for different time windows, picking the same problem and initial guesses as for A1. Figure 2.3.8 gives the convergence curves for different values of the parameter θ for $T = 2$ on the left, and $T = 20$ on the right.

Third arrangement (A3):

We now consider a completely different type of arrangement, proposed in [24] and shown in Figure 2.3.6(b). Given initial guesses along the interfaces, we begin with a Dirichlet solve in the middle subdomain Ω_3, followed by mixed Neumann-Dirichlet subproblem solves in the adjacent subdomains, in an order Ω_2, Ω_4 first and then in Ω_1, Ω_5. This third version of the Dirichlet-Neumann Waveform Ralaxation algorithms for multiple

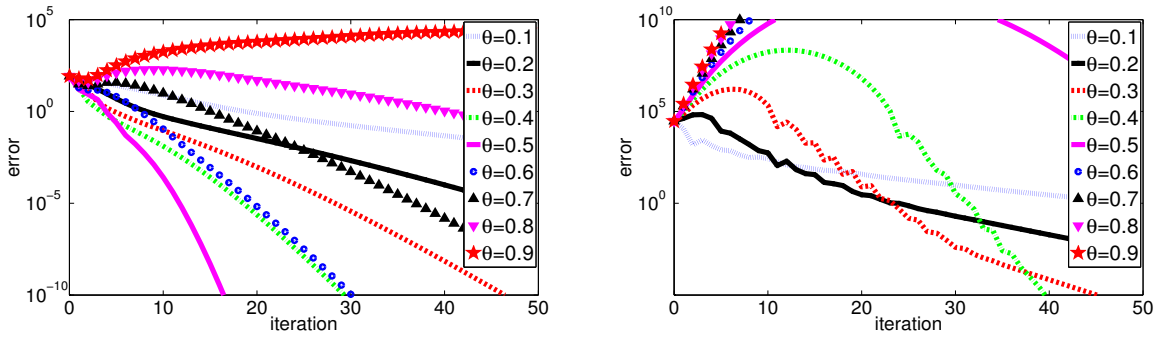


Figure 2.3.8: Convergence of multi-subdomain DNWR for various relaxation parameters θ, on the left for $T = 2$ and on the right for $T = 20$

subdomains is given by: for $k = 1, 2 \ldots$ and for $\theta \in (0, 1]$ compute

$$\begin{aligned} \partial_t u_3^k - \partial_{xx} u_3^k &= 0, && \text{in } \Omega_3, t > 0, \\ u_3^k(x, 0) &= u_0(x), && \text{in } \Omega_3, \\ u_3^k(x_2, t) &= g_2^{k-1}(t), && t > 0, \\ u_3^k(x_3, t) &= g_3^{k-1}(t), && t > 0, \end{aligned} \tag{2.3.34}$$

and then for $i = 2, 1$

$$\begin{aligned} \partial_t u_i^k - \partial_{xx} u_i^k &= 0, && \text{in } \Omega_i, t > 0, \\ u_i^k(x, 0) &= u_0(x), && \text{in } \Omega_i, \\ u_i^k(x_{i-1}, t) &= g_{i-1}^{k-1}(t), && t > 0, \\ \partial_x u_i^k(x_i, t) &= \partial_x u_{i+1}^k(x_i, t), && t > 0, \end{aligned} \tag{2.3.35}$$

and finally for $i = 4, 5$

$$\begin{aligned} \partial_t u_i^k - \partial_{xx} u_i^k &= 0, && \text{in } \Omega_i, t > 0, \\ u_i^k(x, 0) &= u_0(x), && \text{in } \Omega_i, \\ \partial_x u_i^k(x_{i-1}, t) &= \partial_x u_{i-1}^k(x_{i-1}, t), && t > 0, \\ u_i^k(x_i, t) &= g_i^{k-1}(t), && t > 0, \end{aligned} \tag{2.3.36}$$

with $g_0^k(t) = g(0, t), g_5^k(t) = g(5, t)$ for the first and last subdomains at the physical boundaries. The updated interface values for the next step are defined as

$$\begin{aligned} g_i^k(t) &= \theta u_i^k(x_i, t) + (1 - \theta) g_i^{k-1}(t), \ i = 1, 2, \\ g_i^k(t) &= \theta u_{i+1}^k(x_i, t) + (1 - \theta) g_i^{k-1}(t), \ i = 3, 4. \end{aligned}$$

We solve (2.3.33) using the above DNWR algorithm for different time windows for the same setting as in A1. Figure 2.3.9 gives the convergence curves for different values of the parameter θ for $T = 2$ on the left, and $T = 20$ on the right.

From the three numerical tests of the DNWR methods (A1, A2 and A3), it is evident that the behavior of these algorithms are similar for smaller time windows. But we notice clearly faster convergence for the arrangement A3 for large time windows. We therefore focus on the third version (A3) of the DNWR algorithms, and formally define the DNWR method for the general parabolic model problem (2.3.1) for multiple subdomains to be A3, i.e. (2.3.34)-(2.3.35)-(2.3.36).

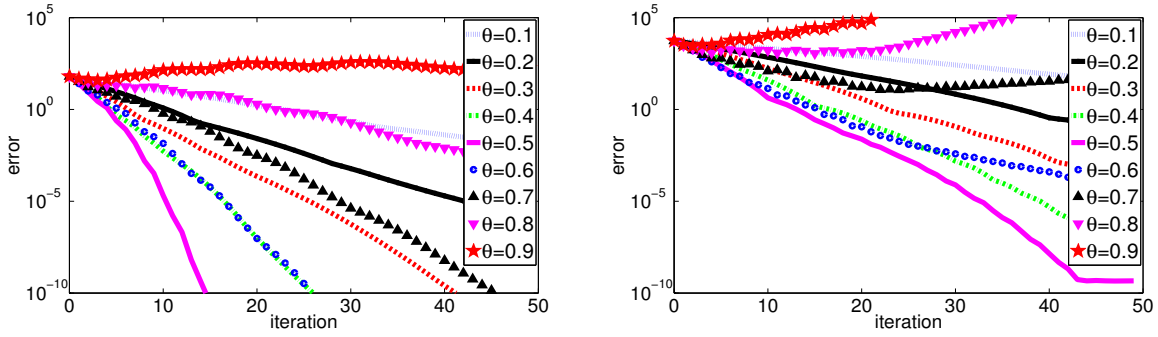


Figure 2.3.9: Convergence of multi-subdomain DNWR for various relaxation parameters θ, on the left for $T = 2$ and on the right for $T = 20$

2.3.2.2 DNWR algorithm

We now formally define the Dirichlet-Neumann Waveform Relaxation method for the model problem (2.3.1) on the space-time domain $\Omega \times (0, T)$ with Dirichlet data given on $\partial\Omega$. Suppose the spatial domain Ω is partitioned into $2m + 1$ non-overlapping subdomains $\Omega_i, i = 1, \ldots, 2m + 1$ without any cross-points, as illustrated in Figure 2.3.10. We denote by u_i the restriction of the solution u of (2.3.1) to Ω_i. For $i = 1, \ldots, 2m$, set $\Gamma_i := \partial\Omega_i \cap \partial\Omega_{i+1}$. We further define $\Gamma_0 = \Gamma_{2m+1} = \emptyset$. We denote by $\boldsymbol{n}_{i,j}$ the unit outward normal for Ω_i on the interface $\Gamma_j, j = i - 1, i$ (for Ω_1, Ω_{2m+1} we have only $\boldsymbol{n}_{1,2}$ and $\boldsymbol{n}_{2m+1,2m}$ respectively). In Figure 2.3.10, D and N in red denote the Dirichlet and Neumann boundary conditions respectively along the interfaces as in the arrangement A3.

The DNWR algorithm starts with initial Dirichlet traces $g_i^0(\boldsymbol{x}, t)$ along the interfaces $\Gamma_i \times (0, T)$, $i = 1, \ldots, 2m$, and then performs the following computation with $\theta \in (0, 1]$ for $k = 1, 2, \ldots$

$$\begin{aligned} \partial_t u_{m+1}^k - \nabla \cdot \left(\kappa(\boldsymbol{x}, t)\nabla u_{m+1}^k\right) &= f, && \text{in } \Omega_{m+1}, \\ u_{m+1}^k(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_{m+1}, \\ u_{m+1}^k &= g, && \text{on } \partial\Omega \cap \partial\Omega_{m+1}, \\ u_{m+1}^k &= g_i^{k-1}, && \text{on } \Gamma_i, i = m, m+1, \end{aligned} \tag{2.3.37}$$

and then for $m \geq i \geq 1$ and $m + 2 \leq j \leq 2m + 1$

$$\begin{aligned} &\partial_t u_i^k = \nabla \cdot \left(\kappa(\boldsymbol{x}, t)\nabla u_i^k\right) + f, \text{in } \Omega_i, && \partial_t u_j^k = \nabla \cdot \left(\kappa(\boldsymbol{x}, t)\nabla u_j^k\right) + f, \text{in } \Omega_j, \\ &u_i^k(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad \text{in } \Omega_i, && u_j^k(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad \text{in } \Omega_j, \\ &u_i^k = \widetilde{g}_i^{k-1}, \quad \text{on } \partial\Omega_i \setminus \Gamma_i, && \partial_{\boldsymbol{n}_{j,j-1}} u_j^k = -\partial_{\boldsymbol{n}_{j-1,j}} u_{j-1}^k, \quad \text{on } \Gamma_{j-1}, \\ &\partial_{\boldsymbol{n}_{i,i+1}} u_i^k = -\partial_{\boldsymbol{n}_{i+1,i}} u_{i+1}^k, \quad \text{on } \Gamma_i, && u_j^k = \check{g}_j^{k-1}, \quad \text{on } \partial\Omega_j \setminus \Gamma_{j-1}, \end{aligned} \tag{2.3.38}$$

with the update conditions along the interfaces

$$\begin{aligned} g_i^k(\boldsymbol{x}, t) &= \theta u_i^k\big|_{\Gamma_i \times (0,T)} + (1 - \theta) g_i^{k-1}(\boldsymbol{x}, t), \ 1 \leq i \leq m, \\ g_j^k(\boldsymbol{x}, t) &= \theta u_{j+1}^k\big|_{\Gamma_j \times (0,T)} + (1 - \theta) g_j^{k-1}(\boldsymbol{x}, t), \ m + 1 \leq j \leq 2m, \end{aligned} \tag{2.3.39}$$

where $\widetilde{g}_i^{k-1} = \begin{cases} g, & \text{on } \partial\Omega \cap \partial\Omega_i, \\ g_{i-1}^{k-1}, & \text{on } \Gamma_{i-1}, \end{cases} i = 1, \ldots, m$ and $\check{g}_j^{k-1} = \begin{cases} g, & \text{on } \partial\Omega \cap \partial\Omega_j, \\ g_j^{k-1}, & \text{on } \Gamma_j, \end{cases} j =$

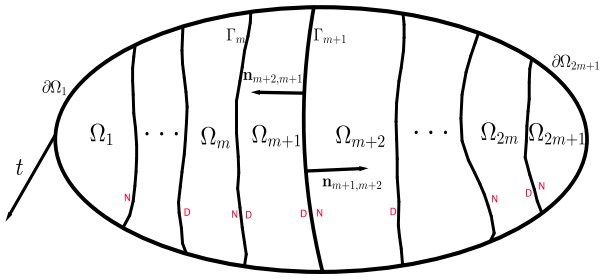


Figure 2.3.10: Splitting into many non-overlapping subdomains

$m+2, \ldots, 2m+1$.

Remark 14. The DNWR algorithm (2.3.37)-(2.3.38)-(2.3.39) is defined for an odd number of subdomains. In case of an even number of subdomains $2m+2$, we treat in a similar way as above the first $2m+1$ subdomains, keeping the last one aside. Then for the last subdomain, we apply a Neumann transmission condition along the interface Γ_{2m+1} and a Dirichlet boundary condition along the physical boundary.

2.3.2.3 Convergence analysis

As in the case of the DNWR algorithm for two subdomains, we prove our results for the 1D heat equation $\kappa(\boldsymbol{x}, t) = \nu$ on the domain $\Omega := (0, L)$ with boundary conditions $u(0,t) = g_0(t)$ and $u(L,t) = g_L(t)$. We decompose Ω into non-overlapping subdomains $\Omega_i := (x_{i-1}, x_i)$, $i = 1, \ldots, 2m+1$, and define the subdomain length $h_i := x_i - x_{i-1}$. Our initial guess is denoted by $\{g_i^0(t)\}_{i=1}^{2m}$ on the interfaces $\{x = x_i\}$. We consider the error equations, $f(x,t) = 0, g_0(t) = g_L(t) = 0 = u_0(x)$, which leads to $g_0^k(t) = g_{2m+1}^k = 0$ for all k. Denoting by $\left\{w_i^k(t)\right\}_{i=1}^{2m}$ for $k = 1, 2 \ldots$ the Neumann traces along the interfaces, we compute

$$\begin{array}{rcll} \partial_t u_{m+1}^k - \nu \partial_{xx} u_{m+1}^k &=& 0, & x \in \Omega_{m+1}, \\ u_{m+1}^k(x,0) &=& 0, & x \in \Omega_{m+1}, \\ u_{m+1}^k(x_m,t) &=& g_m^{k-1}(t), & \\ u_{m+1}^k(x_{m+1},t) &=& g_{m+1}^{k-1}(t). & \end{array} \tag{2.3.40}$$

and then for $m \geq i \geq 1$ and $m+2 \leq j \leq 2m+1$

$$\begin{array}{rlrl} \partial_t u_i^k - \nu \partial_{xx} u_i^k = 0, & x \in \Omega_i, & \partial_t u_j^k - \nu \partial_{xx} u_j^k = 0, & x \in \Omega_j, \\ u_i^k(x,0) = 0, & x \in \Omega_i, & u_j^k(x,0) = 0, & x \in \Omega_j, \\ u_i^k(x_{i-1},t) = g_{i-1}^{k-1}(t), & & -\partial_x u_j^k(x_{j-1},t) = w_{j-1}^k(t), & \\ \partial_x u_i^k(x_i,t) = w_i^k(t), & & u_j^k(x_j,t) = g_j^{k-1}(t), & \end{array} \tag{2.3.41}$$

and finally the update conditions with the parameter $\theta \in (0,1]$

$$\begin{array}{l} g_i^k(t) = \theta u_i^k(x_i,t) + (1-\theta) g_i^{k-1}(t), \quad w_i^k(t) = \partial_x u_{i+1}^k(x_i,t), \ 1 \leq i \leq m, \\ g_j^k(t) = \theta u_{j+1}^k(x_j,t) + (1-\theta) g_j^{k-1}(t), \ w_j^k(t) = -\partial_x u_j^k(x_j,t), \ m+1 \leq j \leq 2m. \end{array} \tag{2.3.42}$$

Theorem 15 (Convergence of DNWR for multiple subdomains)**.** *For $\theta = 1/2$ and $T > 0$ fixed, the DNWR algorithm (2.3.40)-(2.3.41)-(2.3.42) for multiple subdomains, with equal length h, converges superlinearly with the estimate*

$$\max_{1\le i\le 2m} \| g_i^k \|_{L^\infty(0,T)} \le (\min\{2m-1, Q(h,\nu,T)\})^k \operatorname{erfc}\left(\frac{kh}{2\sqrt{\nu T}}\right) \max_{1\le i\le 2m} \| g_i^0 \|_{L^\infty(0,T)},$$

where $Q(h,\nu,T) := 2\operatorname{erfc}\left(\frac{h}{2\sqrt{\nu T}}\right) + \sum_{i=0}^{\infty} 2^{i+1}\operatorname{erfc}\left(\frac{ih}{2\sqrt{\nu T}}\right)$.

Proof. We start by applying the Laplace transform to the homogeneous Dirichlet subproblem in (2.3.40), and obtain

$$s\hat{u}_{m+1}^k - \nu\hat{u}_{m+1,xx}^k = 0, \quad \hat{u}_{m+1}^k(x_m,s) = \hat{g}_m^{k-1}(s), \quad \hat{u}_{m+1}^k(x_{m+1},s) = \hat{g}_{m+1}^{k-1}(s),$$

Defining $\sigma_i := \sinh\left(h_i\sqrt{s/\nu}\right)$ and $\gamma_i := \cosh\left(h_i\sqrt{s/\nu}\right)$, the subdomain (2.3.40) solution becomes

$$\hat{u}_{m+1}^k(x,s) = \frac{1}{\sigma_{m+1}}\left(\hat{g}_{m+1}^{k-1}(s)\sinh\left((x-x_m)\sqrt{s/\nu}\right) + \hat{g}_m^{k-1}(s)\sinh\left((x_{m+1}-x)\sqrt{s/\nu}\right)\right).$$

The solutions of the subproblems (2.3.41) in Laplace space are

$$\begin{aligned}
\hat{u}_i^k(x,s) &= \tfrac{1}{\gamma_i}\tfrac{\hat{w}_i^k}{\sqrt{s/\nu}}\sinh((x-x_{i-1})\sqrt{s/\nu}) + \tfrac{1}{\gamma_i}\hat{g}_{i-1}^{k-1}\cosh((x_i-x)\sqrt{s/\nu}),\\
\hat{u}_j^k(x,s) &= \tfrac{1}{\gamma_j}\hat{g}_j^{k-1}\cosh((x-x_{j-1})\sqrt{s/\nu}) + \tfrac{1}{\gamma_j}\tfrac{\hat{w}_{j-1}^k}{\sqrt{s/\nu}}\sinh((x_j-x)\sqrt{s/\nu}),
\end{aligned}$$

for $1 \le i \le m$ and $m+2 \le j \le 2m+1$. Therefore for $\theta = 1/2$ the update conditions (2.3.42) become

$$\begin{aligned}
\hat{w}_i^k &= -\sqrt{s/\nu}\tfrac{\sigma_{i+1}}{\gamma_{i+1}}\hat{g}_i^{k-1} + \tfrac{1}{\gamma_{i+1}}\hat{w}_{i+1}^k,\ 1\le i\le m-1,\\
\hat{w}_m^k &= -\sqrt{s/\nu}\tfrac{\gamma_{m+1}}{\sigma_{m+1}}\hat{g}_m^{k-1} + \tfrac{\sqrt{s/\nu}}{\sigma_{m+1}}\hat{g}_{m+1}^{k-1},\\
\hat{w}_{m+1}^k &= \tfrac{\sqrt{s/\nu}}{\sigma_{m+1}}\hat{g}_m^{k-1} - \sqrt{s/\nu}\tfrac{\gamma_{m+1}}{\sigma_{m+1}}\hat{g}_{m+1}^{k-1};\\
\hat{w}_j^k &= \tfrac{1}{\gamma_j}\hat{w}_{j-1}^k - \sqrt{s/\nu}\tfrac{\sigma_j}{\gamma_j}\hat{g}_j^{k-1},\ m+2\le j\le 2m,
\end{aligned} \tag{2.3.43}$$

$$\begin{aligned}
\hat{g}_i^k &= \tfrac{1}{2\gamma_i}\hat{g}_{i-1}^{k-1} + \tfrac{1}{2}\hat{g}_i^{k-1} + \tfrac{\sigma_i}{2\gamma_i}\tfrac{\hat{w}_i^k}{\sqrt{s/\nu}},\ 1\le i\le m,\\
\hat{g}_j^k &= \tfrac{\sigma_{j+1}}{2\gamma_{j+1}}\tfrac{\hat{w}_j^k}{\sqrt{s/\nu}} + \tfrac{1}{2}\hat{g}_j^{k-1} + \tfrac{1}{2\gamma_{j+1}}\hat{g}_{j+1}^{k-1},\ m+1\le j\le 2m.
\end{aligned} \tag{2.3.44}$$

For equal-length subdomains we have $h_i = h$ for $i = 1,\dots,2m+1$, so that $\gamma = \gamma_i, \sigma = \sigma_i$ for all i. Now choose $\bar{g}_i^k := \gamma\hat{g}_i^k, \bar{w}_i^k := \frac{\hat{w}_i^k}{\sqrt{s/\nu}}\sigma, 1\le i\le 2m$ to get

$$\bar{g}_i^k = \frac{1}{2\gamma}\bar{g}_{i-1}^{k-1} + \frac{1}{2}\bar{g}_i^{k-1} + \frac{1}{2}\bar{w}_i^k,\ 1\le i\le m;\ \bar{g}_j^k = \frac{1}{2}\bar{w}_j^k + \frac{1}{2}\bar{g}_j^{k-1} + \frac{1}{2\gamma}\bar{g}_{j+1}^{k-1},\ m+1\le j\le 2m,$$

and

$$\begin{aligned}
\bar{w}_i^k &= -\tfrac{\sigma^2}{\gamma^2}\bar{g}_i^{k-1} + \tfrac{1}{\gamma}\bar{w}_{i+1}^k, 1\le i\le m-1;\ \bar{w}_m^k = -\bar{g}_m^{k-1} + \tfrac{1}{\gamma}\bar{g}_{m+1}^{k-1},\\
\bar{w}_{m+1}^k &= \tfrac{1}{\gamma}\bar{g}_m^{k-1} - \bar{g}_{m+1}^{k-1};\quad \bar{w}_j^k = \tfrac{1}{\gamma}\bar{w}_{j-1}^k - \tfrac{\sigma^2}{\gamma^2}\bar{g}_j^{k-1},\ m+2\le j\le 2m.
\end{aligned}$$

Therefore in matrix form, the system becomes

$$A\begin{pmatrix} \bar{g}_1^k \\ \bar{w}_1^k \\ \vdots \\ \vdots \\ \bar{g}_m^k \\ \bar{w}_m^k \\ \bar{w}_{m+1}^k \\ \bar{g}_{m+1}^k \\ \vdots \\ \vdots \\ \bar{w}_{2m}^k \\ \bar{g}_{2m}^k \end{pmatrix} = B\begin{pmatrix} \bar{g}_1^{k-1} \\ \vdots \\ \bar{g}_m^{k-1} \\ \bar{g}_{m+1}^{k-1} \\ \vdots \\ \bar{g}_{2m}^{k-1} \end{pmatrix}, \text{ where } A = \begin{pmatrix} L_1 & \mathbf{0} \\ \mathbf{0} & L_2 \end{pmatrix}, L_1 = \begin{bmatrix} 1 & -\frac{1}{2} & & & & & & \\ & 1 & 0 & -\frac{1}{\gamma} & & & & \\ & & 1 & -\frac{1}{2} & & & & \\ & & & 1 & 0 & -\frac{1}{\gamma} & & \\ & & & & \ddots & \ddots & & \\ & & & & & 1 & 0 & -\frac{1}{\gamma} \\ & & & & & & 1 & -\frac{1}{2} \\ & & & & & & & 1 \end{bmatrix}_{2m\times 2m},$$

$$L_2 = \begin{bmatrix} 1 & & & & & & & \\ -\frac{1}{2} & 1 & & & & & & \\ -\frac{1}{\gamma} & 0 & 1 & & & & & \\ & & -\frac{1}{2} & 1 & & & & \\ & & -\frac{1}{\gamma} & 0 & 1 & & & \\ & & & & \ddots & \ddots & & \\ & & & & -\frac{1}{\gamma} & 0 & 1 & \\ & & & & & & -\frac{1}{2} & 1 \end{bmatrix}_{2m\times 2m} \quad \text{and } B = \left[\begin{array}{cccccccc} \frac{1}{2} & & & & & & & \\ -\frac{\sigma^2}{\gamma^2} & & & & & & & \\ \frac{1}{2\gamma} & \frac{1}{2} & & & & & & \\ & -\frac{\sigma^2}{\gamma^2} & & & & & & \\ & \frac{1}{2\gamma} & \frac{1}{2} & & & & & \\ & & \ddots & & & & & \\ & & \frac{1}{2\gamma} & \frac{1}{2} & & & & \\ & & & -1 & \frac{1}{\gamma} & & & \\ \hline & & & \frac{1}{\gamma} & -1 & & & \\ & & & & \frac{1}{2} & \frac{1}{2\gamma} & & \\ & & & & & -\frac{\sigma^2}{\gamma^2} & & \\ & & & & & \frac{1}{2} & \frac{1}{2\gamma} & \\ & & & & & & \ddots & \\ & & & & & & \frac{1}{2} & \frac{1}{2\gamma} \\ & & & & & & & -\frac{\sigma^2}{\gamma^2} \\ & & & & & & & \frac{1}{2} \end{array}\right].$$

Inverting A, we obtain

$$\begin{pmatrix} \bar{g}_1^k \\ \vdots \\ \bar{g}_m^k \\ \bar{g}_{m+1}^k \\ \vdots \\ \bar{g}_{2m}^k \end{pmatrix} = P\begin{pmatrix} \bar{g}_1^{k-1} \\ \vdots \\ \bar{g}_m^{k-1} \\ \bar{g}_{m+1}^{k-1} \\ \vdots \\ \bar{g}_{2m}^{k-1} \end{pmatrix},$$

where $P = \begin{bmatrix} \frac{1}{2\gamma^2} & -\frac{\sigma^2}{2\gamma^3} & \cdots & -\frac{\sigma^2}{2\gamma^m} & -\frac{1}{2\gamma^{m-1}} & \frac{1}{2\gamma^m} & & & & \\ \frac{1}{2\gamma} & \frac{1}{2\gamma^2} & \ddots & \ddots & -\frac{1}{2\gamma^{m-2}} & \frac{1}{2\gamma^{m-1}} & & & & \\ & \ddots & \ddots & \ddots & \vdots & \vdots & & & & \\ & & \frac{1}{2\gamma} & \frac{1}{2\gamma^2} & -\frac{1}{2\gamma} & \frac{1}{2\gamma^2} & & & & \\ & & & \frac{1}{2\gamma} & 0 & \frac{1}{2\gamma} & & & & \\ & & & & \frac{1}{2\gamma} & 0 & \frac{1}{2\gamma} & & & \\ & & & & \frac{1}{2\gamma^2} & -\frac{1}{2\gamma} & \frac{1}{2\gamma^2} & \frac{1}{2\gamma} & & \\ & & & & \vdots & \vdots & \ddots & \ddots & \ddots & \\ & & & & \frac{1}{2\gamma^{m-1}} & -\frac{1}{2\gamma^{m-2}} & \ddots & \ddots & \frac{1}{2\gamma^2} & \frac{1}{2\gamma} \\ & & & & \frac{1}{2\gamma^m} & -\frac{1}{2\gamma^{m-1}} & -\frac{\sigma^2}{2\gamma^m} & \cdots & -\frac{\sigma^2}{2\gamma^3} & \frac{1}{2\gamma^2} \end{bmatrix}_{2m\times 2m}$.

Therefore, we can write the first equation as

$$\hat{g}_1^k = \frac{1}{2\gamma^2}\hat{g}_1^{k-1} - \frac{\sigma^2}{2\gamma^3}\hat{g}_2^{k-1} - \cdots - \frac{\sigma^2}{2\gamma^m}\hat{g}_{m-1}^{k-1} - \frac{1}{2\gamma^{m-1}}\hat{g}_m^{k-1} + \frac{1}{2\gamma^m}\hat{g}_{m+1}^{k-1},\ m \geq 2. \quad (2.3.45)$$

Now setting $\hat{v}_i^k(s) = 2^k\gamma^k\hat{g}_i^k(s)$, (2.3.45) becomes

$$\begin{aligned} \hat{v}_1^k &= \frac{1}{\gamma}\hat{v}_1^{k-1} - \left(1-\frac{1}{\gamma^2}\right)\hat{v}_2^{k-1} - \cdots - \frac{1}{\gamma^{m-3}}\left(1-\frac{1}{\gamma^2}\right)\hat{v}_{m-1}^{k-1} - \frac{1}{\gamma^{m-2}}\hat{v}_m^{k-1} + \frac{1}{\gamma^{m-1}}\hat{v}_{m+1}^{k-1} \\ &=: \sum_{i=1}^{m+1}\hat{r}_i\hat{v}_i^{k-1}. \end{aligned} \quad (2.3.46)$$

Thus by back transforming into the time domain we obtain $v_1^k(t) = \sum_{i=1}^{m+1}\left(r_i * v_i^{k-1}\right)(t)$. Now part 3 of Lemma 7 yields

$$\|v_1^k\|_{L^\infty(0,T)} \leq \sum_{i=1}^{m+1}\|v_i^{k-1}\|_{L^\infty(0,T)}\int_0^T |r_i(\tau)|d\tau \leq \max_{1\leq i\leq m+1}\| v_i^{k-1} \|_{L^\infty(0,T)}\sum_{i=1}^{m+1}\int_0^T |r_i(\tau)|d\tau. \quad (2.3.47)$$

By Lemma 8, $\int_0^T |r_i(\tau)|d\tau \leq 1$, $\int_0^T |r_j(\tau)|d\tau \leq 2$ for $i = 1, m, m+1$ and $2 \leq j \leq m-1$, and we therefore get

$$\begin{aligned} \| v_1^k(\cdot) \|_{L^\infty(0,T)} &\leq (3+2(m-2))\max_{1\leq j\leq m+1}\| v_j^{k-1}(\cdot) \|_{L^\infty(0,T)} \\ &\leq (2m-1)\max_{1\leq j\leq 2m}\| v_j^{k-1}(\cdot) \|_{L^\infty(0,T)}. \end{aligned}$$

Similarly we get the relations

$$\| v_i^k(\cdot) \|_{L^\infty(0,T)} \leq (2m-1)\max_{1\leq j\leq 2m}\| v_j^{k-1}(\cdot) \|_{L^\infty(0,T)}.$$

By induction, we therefore get

$$\| v_i^k(\cdot) \|_{L^\infty(0,T)} \leq (2m-1)\max_{1\leq j\leq 2m}\| v_j^{k-1} \|_{L^\infty(0,T)} \leq \cdots \leq (2m-1)^k\max_{1\leq j\leq 2m}\| g_j^0(\cdot) \|_{L^\infty(0,T)}. \quad (2.3.48)$$

With a similar argument as in the proof of Theorem 11 we also obtain

$$g_i^k(t) = \frac{1}{2^k}\left(J_k * v_i^k\right)(t) = \frac{1}{2^k}\int_0^t J_k(t-\tau)v_i^k(\tau)d\tau,$$

where $J_k(t) := \mathcal{L}^{-1}\left(\frac{1}{\gamma^k}\right)$. So part 3 of Lemma 7 gives

$$\|g_i^k(\cdot)\|_{L^\infty(0,T)} \leq \operatorname{erfc}\left(\frac{kh}{2\sqrt{\nu T}}\right)\|v_i^k(\cdot)\|_{L^\infty(0,T)}. \tag{2.3.49}$$

Thus combining (2.3.48) and (2.3.49) we get the first estimate

$$\max_{1\leq i\leq 2m}\| g_i^k \|_{L^\infty(0,T)} \leq (2m-1)^k \operatorname{erfc}\left(\frac{kh}{2\sqrt{\nu T}}\right)\max_{1\leq i\leq 2m}\| g_i^0 \|_{L^\infty(0,T)}. \tag{2.3.50}$$

We now present a different estimate starting from (2.3.46). Since $\lim_{s\to 0+}\left|1-\frac{1}{\gamma^2}\right| \leq 2$, we have from (2.3.47)

$$\sum_{i=1}^{m+1}\int_0^T |r_i(\tau)|d\tau \leq \int_0^T \left(J_1(t) + 2 + 2J_1(t) + \cdots + 2J_{m-3}(t) + J_{m-2}(t) + J_{m-1}(t)\right)dt.$$

Therefore using the inequality (2.3.24) $\int_0^T J_k(t)dt \leq 2^k \operatorname{erfc}\left(\frac{kh}{2\sqrt{\nu T}}\right)$, the above expression is bounded by

$$\sum_{i=1}^{m+1}\int_0^T |r_i(\tau)|d\tau \leq 2\operatorname{erfc}\left(\frac{h}{2\sqrt{\nu T}}\right) + \sum_{i=0}^{m-1} 2^{i+1}\operatorname{erfc}\left(\frac{ih}{2\sqrt{\nu T}}\right) \leq Q(h,\nu,T),$$

where $Q(h,\nu,T) := 2\operatorname{erfc}\left(\frac{h}{2\sqrt{\nu T}}\right) + \sum_{i=0}^{\infty} 2^{i+1}\operatorname{erfc}\left(\frac{ih}{2\sqrt{\nu T}}\right)$. So we get the second estimate as

$$\max_{1\leq i\leq 2m}\| g_i^k \|_{L^\infty(0,T)} \leq Q^k \operatorname{erfc}\left(\frac{kh}{2\sqrt{\nu T}}\right)\max_{1\leq i\leq 2m}\| g_i^0 \|_{L^\infty(0,T)}. \tag{2.3.51}$$

The result follows combining the two estimates (2.3.50) and (2.3.51). □

Remark 16. The estimate in Theorem 15 holds for an odd number of subdomains. In case of an even number of equal subdomains $2m+2$, we define the algorithm as given in Remark 14, and the estimate will be of the form

$$\max_{1\leq i\leq 2m+1}\| g_i^k \|_{L^\infty(0,T)} \leq \left(\min\left\{2m+1, Q(h,\nu,T)\right\}\right)^k \operatorname{erfc}\left(\frac{kh}{2\sqrt{\nu T}}\right)\max_{1\leq i\leq 2m+1}\| g_i^0 \|_{L^\infty(0,T)}, \tag{2.3.52}$$

where $Q(h,\nu,T) := 2\operatorname{erfc}\left(\frac{h}{2\sqrt{\nu T}}\right) + \sum_{i=0}^{\infty} 2^{i+1}\operatorname{erfc}\left(\frac{ih}{2\sqrt{\nu T}}\right)$. $2m+1$ appears in (2.3.52) as we calculate the estimate for $2m+3$ subdomains.

We compare the two estimates (2.3.50) and (2.3.51) in Figure 2.3.11 for $\nu = 1$. The region below the red curve is where the estimate (2.3.50) is more accurate than (2.3.51).

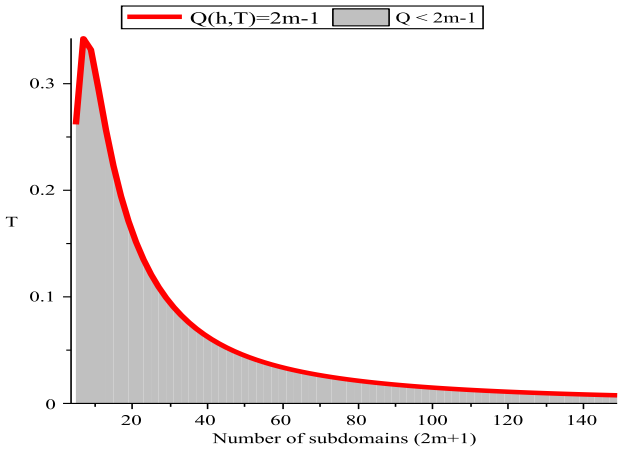


Figure 2.3.11: Comparison of the two estimates (2.3.50) and (2.3.51) for $\nu = 1$

Theorem 17 (Convergence of DNWR for unequal subdomains)**.** *For unequal subdomains, $h_i, 1 \le i \le 2m+1$ with $h_{\max} := \max_{1\le i\le 2m+1} h_i$ and $h_{\min} := \min_{1\le i\le 2m+1} h_i$, the DNWR algorithm converges superlinearly with the estimate*

$$\max_{1\le i\le 2m} \| g_i^k \|_{L^\infty(0,T)} \le \left(2m-3+\frac{2h_{\max}}{h_{m+1}}\right)^k \operatorname{erfc}\left(\frac{kh_{\min}}{2\sqrt{\nu T}}\right) \max_{1\le i\le 2m} \| g_i^0 \|_{L^\infty(0,T)} .$$

Proof. We provide a sketch of the proof. For unequal h_i's and $\theta = 1/2$, choose $\bar{g}_i^k := \gamma_i \hat{g}_i^k$, $\bar{w}_i^k := \frac{\hat{w}_i^k}{\sqrt{s/\nu}}\sigma_i, 1 \le i \le m$ and $\bar{g}_j^k := \gamma_{j+1}\hat{g}_j^k, \bar{w}_j^k := \frac{\hat{w}_j^k}{\sqrt{s/\nu}}\sigma_{j+1}, m+1 \le j \le 2m$ with $\gamma_0 = \gamma_{2m+2} = 1$ in (2.3.43)-(2.3.44) to get

$$\bar{g}_i^k = \tfrac{1}{2\gamma_{i-1}}\bar{g}_{i-1}^{k-1} + \tfrac{1}{2}\bar{g}_i^{k-1} + \tfrac{1}{2}\bar{w}_i^k,\ 1 \le i \le m;\ \bar{g}_j^k = \tfrac{1}{2}\bar{w}_j^k + \tfrac{1}{2}\bar{g}_j^{k-1} + \tfrac{1}{2\gamma_{j+2}}\bar{g}_{j+1}^{k-1},\ m+1 \le j \le 2m,$$

$$\bar{w}_i^k = -\tfrac{\sigma_i\sigma_{i+1}}{\gamma_i\gamma_{i+1}}\bar{g}_i^{k-1} + \tfrac{\sigma_i}{\sigma_{i+1}\gamma_{i+1}}\bar{w}_{i+1}^k, 1 \le i \le m-1;\ \bar{w}_m^k = -\tfrac{\sigma_m\gamma_{m+1}}{\gamma_m\sigma_{m+1}}\bar{g}_m^{k-1} + \tfrac{\sigma_m}{\sigma_{m+1}\gamma_{m+2}}\bar{g}_{m+1}^{k-1},$$
$$\bar{w}_{m+1}^k = \tfrac{\sigma_{m+2}}{\gamma_m\sigma_{m+1}}\bar{g}_m^{k-1} - \tfrac{\gamma_{m+1}\sigma_{m+2}}{\sigma_{m+1}\gamma_{m+2}}\bar{g}_{m+1}^{k-1};\quad \bar{w}_j^k = \tfrac{\sigma_{j+1}}{\gamma_j\sigma_j}\bar{w}_{j-1}^k - \tfrac{\sigma_j\sigma_{j+1}}{\gamma_j\gamma_{j+1}}\bar{g}_j^{k-1},\ m+2 \le j \le 2m.$$

Therefore in the above proof, the matrices L_1, L_2, B and P are in the following form

$$L_1 = \begin{bmatrix} 1 & -\frac{1}{2} & & & & & \\ 1 & 0 & -\frac{\sigma_1}{\sigma_2\gamma_2} & & & & \\ & 1 & -\frac{1}{2} & & & & \\ & & 1 & 0 & -\frac{\sigma_2}{\sigma_3\gamma_3} & & \\ & & & \ddots & \ddots & & \\ & & & & 1 & 0 & -\frac{\sigma_{m-1}}{\sigma_m\gamma_m} \\ & & & & & 1 & -\frac{1}{2} \\ & & & & & & 1 \end{bmatrix}, L_2 = \begin{bmatrix} 1 & & & & & & \\ -\frac{1}{2} & 1 & & & & & \\ -\frac{\sigma_{i+1}}{\sigma_i\gamma_i} & 0 & 1 & & & & \\ & -\frac{1}{2} & 1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & -\frac{1}{2} & 1 & & \\ & & & -\frac{\sigma_{2m+1}}{\sigma_{2m}\gamma_{2m}} & 0 & 1 & \\ & & & & & -\frac{1}{2} & 1 \end{bmatrix},$$

$$B = \left[\begin{array}{ccccc|ccccc}
\frac{1}{2} & & & & & & & & & \\
-\frac{\sigma_1\sigma_2}{\gamma_1\gamma_2} & & & & & & & & & \\
\frac{1}{2\gamma_1} & \frac{1}{2} & & & & & & & & \\
 & -\frac{\sigma_2\sigma_3}{\gamma_2\gamma_3} & & & & & & & & \\
 & \frac{1}{2\gamma_2} & \frac{1}{2} & & & & & & & \\
 & & \ddots & & & & & & & \\
 & & \frac{1}{2\gamma_{m-1}} & \frac{1}{2} & & & & & & \\
 & & & -\frac{\sigma_m\gamma_{m+1}}{\gamma_m\sigma_{m+1}} & \frac{\sigma_m}{\sigma_{m+1}\gamma_{m+2}} & & & & & \\
 & & & \frac{\sigma_{m+2}}{\sigma_{m+1}\gamma_m} & -\frac{\sigma_{m+2}\gamma_{m+1}}{\gamma_{m+2}\sigma_{m+1}} & & & & & \\
 & & & & \frac{1}{2} & \frac{1}{2\gamma_{m+3}} & & & & \\
 & & & & & -\frac{\sigma_{m+2}\sigma_{m+3}}{\gamma_{m+2}\gamma_{m+3}} & & & & \\
 & & & & & \frac{1}{2} & \frac{1}{2\gamma_{m+4}} & & & \\
 & & & & & & \ddots & & & \\
 & & & & & & \frac{1}{2} & \frac{1}{2\gamma_{2m+1}} & & \\
 & & & & & & & -\frac{\sigma_{2m}\sigma_{2m+1}}{\gamma_{2m}\gamma_{2m+1}} & & \\
 & & & & & & & \frac{1}{2} & &
\end{array}\right],$$

$$P = \left(\begin{array}{c|c} K_1 & K_2 \\ \hline K_3 & K_4 \end{array}\right)_{2m\times 2m}, K_1 = \begin{bmatrix}
\frac{\gamma_{1,2}}{2\gamma_1\gamma_2} & -\frac{\sigma_1\sigma_3}{2\gamma_2^2\gamma_3} & \cdots & \frac{-\sigma_1\sigma_m}{2\gamma_2\ldots\gamma_{m-2}\gamma_{m-1}^2\gamma_m} & \frac{-\sigma_1\gamma_{m+1}}{2\gamma_2\ldots\gamma_{m-1}\gamma_m^2\sigma_{m+1}} \\
\frac{1}{2\gamma_1} & \frac{\gamma_{2,3}}{2\gamma_2\gamma_3} & \ddots & \frac{-\sigma_2\sigma_m}{2\gamma_3\ldots\gamma_{m-2}\gamma_{m-1}^2\gamma_m} & \frac{-\sigma_2\gamma_{m+1}}{2\gamma_3\ldots\gamma_{m-1}\gamma_m^2\sigma_{m+1}} \\
0 & \ddots & \ddots & \vdots & \vdots \\
\vdots & 0 & \frac{1}{2\gamma_{m-2}} & \frac{\gamma_{m-1,m}}{2\gamma_{m-1}\gamma_m} & -\frac{\sigma_{m-1}\gamma_{m+1}}{2\gamma_m^2\sigma_{m+1}} \\
0 & \cdots & 0 & \frac{1}{2\gamma_{m-1}} & \frac{\sigma_{m+1,m}}{2\sigma_{m+1}\gamma_m}
\end{bmatrix},$$

$$K_2 = \begin{bmatrix}
\frac{\sigma_1}{2\gamma_2\ldots\gamma_m\sigma_{m+1}\gamma_{m+2}} & 0 & \cdots & \cdots & 0 \\
\frac{\sigma_2}{2\gamma_3\ldots\gamma_m\sigma_{m+1}\gamma_{m+2}} & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\sigma_{m-1}}{2\gamma_m\sigma_{m+1}\gamma_{m+2}} & \vdots & \vdots & \vdots & \vdots \\
\frac{\sigma_m}{2\sigma_{m+1}\gamma_{m+2}} & 0 & \cdots & \cdots & 0
\end{bmatrix}, K_3 = \begin{bmatrix}
0 & \cdots & \cdots & 0 & \frac{\sigma_{m+2}}{2\gamma_m\sigma_{m+1}} \\
\vdots & \vdots & \vdots & \vdots & \frac{\sigma_{m+3}}{2\gamma_m\sigma_{m+1}\gamma_{m+2}} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \frac{\sigma_{2m}}{2\gamma_m\sigma_{m+1}\gamma_{m+2}\ldots\gamma_{2m-1}} \\
0 & \cdots & \cdots & 0 & \frac{\sigma_{2m+1}}{2\gamma_m\sigma_{m+1}\gamma_{m+2}\ldots\gamma_{2m}}
\end{bmatrix},$$

$$K_4 = \begin{bmatrix}
\frac{\sigma_{m+1,m+2}}{2\sigma_{m+1}\gamma_{m+2}} & \frac{1}{2\gamma_{m+3}} & 0 & \cdots & 0 \\
-\frac{\sigma_{m+3}\gamma_{m+1}}{2\sigma_{m+1}\gamma_{m+2}^2} & \frac{\gamma_{m+2,m+3}}{2\gamma_{m+2}\gamma_{m+3}} & \frac{1}{2\gamma_{m+4}} & 0 & \vdots \\
\vdots & \vdots & \ddots & \ddots & 0 \\
\frac{-\sigma_{2m}\gamma_{m+1}}{2\sigma_{m+1}\gamma_{m+2}^2\gamma_{m+3}\ldots\gamma_{2m-1}} & \frac{-\sigma_{2m}\sigma_{m+2}}{2\gamma_{m+2}\gamma_{m+3}^2\gamma_{m+4}\ldots\gamma_{2m-1}} & \ddots & \frac{\gamma_{2m-1,2m}}{2\gamma_{2m-1}\gamma_{2m}} & \frac{1}{2\gamma_{2m+1}} \\
\frac{-\sigma_{2m+1}\gamma_{m+1}}{2\sigma_{m+1}\gamma_{m+2}^2\gamma_{m+3}\ldots\gamma_{2m}} & \frac{-\sigma_{2m+1}\sigma_{m+2}}{2\gamma_{m+2}\gamma_{m+3}^2\gamma_{m+4}\ldots\gamma_{2m}} & \cdots & -\frac{\sigma_{2m-1}\sigma_{2m+1}}{2\gamma_{2m-1}\gamma_{2m}^2} & \frac{\gamma_{2m,2m+1}}{2\gamma_{2m}\gamma_{2m+1}}
\end{bmatrix},$$

with $\gamma_{i,j} = \cosh\left((h_i - h_j)\sqrt{s/\nu}\right), \sigma_{i,j} = \sinh\left((h_i - h_j)\sqrt{s/\nu}\right)$. So with the substitution $\hat{v}_i^k := 2^k\gamma^k\hat{g}_i^k$, $\gamma = \cosh\left(h_{\min}\sqrt{s/\nu}\right)$, we get

$\hat{v}_1^k = \frac{\gamma\gamma_{1,2}}{\gamma_1\gamma_2}\hat{v}_1^{k-1} - \frac{\gamma\sigma_1\sigma_3}{\gamma_1\gamma_2\gamma_3}\hat{v}_2^{k-1} - \cdots - \frac{\gamma\sigma_1\sigma_m}{\gamma_1\gamma_2\ldots\gamma_{m-1}\gamma_m}\hat{v}_{m-1}^{k-1} - \frac{\gamma\sigma_1\gamma_{m+1}}{\gamma_1\ldots\gamma_m\sigma_{m+1}}\hat{v}_m^{k-1} + \frac{\gamma\sigma_1}{\gamma_1\ldots\gamma_m\sigma_{m+1}}\hat{v}_{m+1}^{k-1}$
$=: \sum_{i=1}^{m+1}\hat{d}_{1,i}\hat{v}_i^{k-1}.$

Now using Lemma 8, we obtain

$$\int_0^\infty |d_{1,1}(t)|dt \le \lim_{s\to 0+}\frac{\cosh(h_{\min}\sqrt{s/\nu})\cosh\left((h_1-h_2)\sqrt{s/\nu}\right)}{\cosh(h_2\sqrt{s/\nu})\cosh(h_1\sqrt{s/\nu})} \le 1,$$

so that

$$\|\mathcal{L}^{-1}\left(\hat{d}_{1,1}\,\hat{\nu}_1^{k-1}(s)\right)\|_{L^\infty(0,T)} \le \|\nu_1^{k-1}\|_{L^\infty(0,T)}.$$

Moreover, we write for $j = 2,\ldots,m-1$

$$\begin{aligned}\hat{d}_{1,j} &= \frac{\gamma}{\gamma_2\ldots\gamma_j}\left(1-\frac{\sigma_1\sigma_{j+1}}{\gamma_1\gamma_{j+1}}-1\right) = \frac{\gamma\cosh\left((h_1-h_{j+1})\sqrt{s/\nu}\right)}{\gamma_2\ldots\gamma_j\gamma_1\gamma_{j+1}} - \frac{\gamma}{\gamma_2\ldots\gamma_j}\\ &= \frac{\cosh\left((h_{\min}+h_1-h_{j+1})\sqrt{s/\nu}\right)+\cosh\left((h_{\min}-h_1+h_{j+1})\sqrt{s/\nu}\right)}{2\gamma_2\ldots\gamma_j\gamma_1\gamma_{j+1}} - \frac{\gamma}{\gamma_2\ldots\gamma_j}.\end{aligned}$$

Now $-h_{j+1} \le h_{\min}+h_1-h_{j+1} = h_1+h_{\min}-h_{j+1} \le h_1$, so for the first term, we choose the pairing

$$\frac{1}{\gamma_2\ldots\gamma_j}\cdot\frac{1}{2\cosh(h_{j+1}\sqrt{s/\nu})}\cdot\frac{\cosh\left((h_{\min}+h_1-h_{j+1})\sqrt{s/\nu}\right)}{\cosh(h_1\sqrt{s/\nu})}$$

and

$$\frac{1}{\gamma_2\ldots\gamma_j}\cdot\frac{1}{2\cosh(h_1\sqrt{s/\nu})}\cdot\frac{\cosh\left((h_{\min}-h_1+h_{j+1})\sqrt{s/\nu}\right)}{\cosh(h_{j+1}\sqrt{s/\nu})}.$$

We therefore get

$$\int_0^\infty |d_{1,j}(t)|\,dt \le \lim_{s\to 0+}\left(\frac{\cosh\left((h_{\min}+h_1-h_{j+1})\sqrt{s/\nu}\right)+\cosh\left((h_{\min}-h_1+h_{j+1})\sqrt{s/\nu}\right)}{2\gamma_2\ldots\gamma_j\gamma_1\gamma_{j+1}} + \frac{\gamma}{\gamma_2\ldots\gamma_j}\right) \le 2,$$

so that

$$\left\|\mathcal{L}^{-1}\left(\hat{d}_{1,j}\hat{\nu}_j^{k-1}(s)\right)\right\|_{L^\infty(0,T)} \le 2\,\|\nu_j^{k-1}\|_{L^\infty(0,T)}.$$

Finally we write $\hat{d}_{1,m} = \frac{\gamma}{\gamma_2\ldots\gamma_m}\left(1-\frac{\sigma_1\gamma_{m+1}}{\gamma_1\sigma_{m+1}}-1\right) = -\frac{\gamma}{\gamma_2\ldots\gamma_m}\cdot K_1(s) - \frac{\gamma}{\gamma_2\ldots\gamma_m}$, where $K_1(s) = \frac{\sinh\left((h_1-h_{m+1})\sqrt{s/\nu}\right)}{\sinh(h_{m+1}\sqrt{s/\nu})\cosh(h_1\sqrt{s/\nu})}$, and write

$$\hat{d}_{1,m+1} = \frac{\sigma}{\sigma_{m+1}}\cdot\frac{1}{\gamma_2\ldots\gamma_m}\left(\frac{\sigma_1\gamma}{\gamma_1\sigma}-1+1\right) = \frac{\sigma}{\sigma_{m+1}}\cdot\frac{1}{\gamma_2\ldots\gamma_m}\cdot K_2(s) + \frac{\sigma}{\sigma_{m+1}}\cdot\frac{1}{\gamma_2\ldots\gamma_m},$$

where $\sigma = \sinh\left(h_{\min}\sqrt{s/\nu}\right)$, $K_2(s) = \frac{\sinh\left((h_1-h_{\min})\sqrt{s/\nu}\right)}{\sinh(h_{\min}\sqrt{s/\nu})\cosh(h_1\sqrt{s/\nu})}$. Note that both $K_1(s)$ and $K_2(s)$ are of the form (2.3.9). So by Theorem 11 (for $h_{m+1} > h_1$) or by Theorem 12 (for $h_{m+1} < h_1$), we obtain the bounds

$$\int_0^\infty |d_{1,m}(t)|\,dt \le \lim_{s\to 0+} \frac{\gamma}{\gamma_2\ldots\gamma_m}\,(K_1(s)+1) \le \frac{h_1-h_{m+1}}{h_{m+1}} + 1 = \frac{h_1}{h_{m+1}},$$

and

$$\int_0^\infty |d_{1,m+1}(t)|dt \le \lim_{s\to 0+} \frac{\sigma}{\sigma_{m+1}}\cdot\frac{1}{\gamma_2\ldots\gamma_m}\,(K_2(s)+1) \le \frac{h_{\min}}{h_{m+1}}\left(\frac{h_1-h_{\min}}{h_{\min}}+1\right) = \frac{h_1}{h_{m+1}}.$$

We therefore get

$$\begin{aligned}\left\|v_1^k(\cdot)\right\|_{L^\infty(0,T)} &\le \left(1+2(m-2)+\frac{2h_1}{h_{m+1}}\right)\max_{1\le j\le m+1}\| v_j^{k-1}(\cdot)\|_{L^\infty(0,T)}\\ &\le \left(2m-3+\frac{2h_1}{h_{m+1}}\right)\max_{1\le j\le 2m}\| v_j^{k-1}(\cdot)\|_{L^\infty(0,T)}\,.\end{aligned}$$

We also get similar relations for other equations. By induction, we therefore get

$$\begin{aligned}\left\|v_i^k(\cdot)\right\|_{L^\infty(0,T)} &\le \left(2m-3+\frac{2h_i}{h_{m+1}}\right)\max_{1\le j\le 2m}\| v_j^{k-1}(\cdot)\|_{L^\infty(0,T)}\\ &\le \cdots \le \left(2m-3+\frac{2h_{\max}}{h_{m+1}}\right)^k\max_{1\le j\le 2m}\| g_j^0(\cdot)\|_{L^\infty(0,T)}\,.\end{aligned}$$

Now using the last part of the proof of Theorem 15, we get the expression of the estimate. □

2.3.2.4 Numerical illustration

We show some experiments for the DNWR algorithm in the spatial domain $\Omega = (0,5)$, for the problem $\partial_t u = \Delta u$, with initial condition $u_0(x) = x(5-x)$ and boundary conditions $u(0,t) = t^2, u(5,t) = te^{-t}$. In the first experiment we apply the DNWR for a decomposition into five subdomains and for three different time windows $T = 0.2, T = 2$ and $T = 8$, whereas for a fixed time $T = 2$ we run another experiment for three to six equal subdomains. In Figure 2.3.12, on the left panel, we show the convergence estimate together with the numerically measured convergence in the five-subdomain case as a function of time T, whereas on the right panel, we show the convergence for $T = 2$ as we vary the number of subdomains. We observe superlinear convergence as predicted by the theory, and for small T the estimate is quite sharp. We also see that the convergence slows down as the number of subdomains is increased.

2.4 DNWR for Hyperbolic problems

We now define the DNWR method for the second order hyperbolic equation

$$\begin{aligned}\partial_{tt}u - c^2(\boldsymbol{x})\triangle u &= f(\boldsymbol{x},t), && \boldsymbol{x}\in\Omega, 0<t<T,\\ u(\boldsymbol{x},0) = u_0(\boldsymbol{x}),\ u_t(\boldsymbol{x},0) &= v_0(\boldsymbol{x}), && \boldsymbol{x}\in\Omega,\\ u(\boldsymbol{x},t) &= g(\boldsymbol{x},t), && \boldsymbol{x}\in\partial\Omega, 0<t<T,\end{aligned} \tag{2.4.1}$$

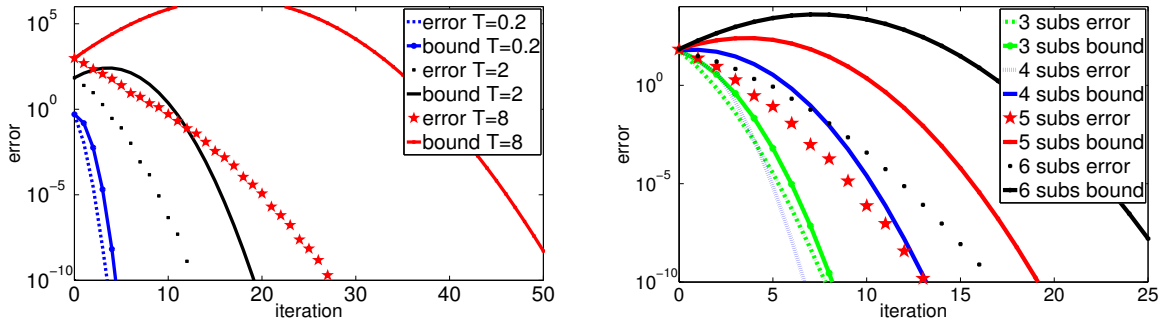


Figure 2.3.12: Convergence estimates of DNWR for $\theta = 1/2$, on the left for various values of T for five subdomains, and on the right for various number of subdomains for $T = 2$

where $c(\boldsymbol{x})$ is a positive function and $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, is a bounded domain with a smooth boundary. The method is based on a non-overlapping spatial domain decomposition, and the iteration involves subdomain solves in space-time with corresponding interface condition, followed by a correction step.

2.4.1 DNWR for two subdomains

To explain the new algorithm for (2.4.1), we take the same spatial domain decomposition of Ω into non-overlapping subdomains Ω_1 and Ω_2, as in Figure 2.3.1 in Subsection 2.3.1.

The DNWR algorithm consists of the following steps: given an initial guess $h^0(\boldsymbol{x}, t)$, $t \in (0, T)$ along the interface $\Gamma \times (0, T)$, compute for $k = 1, 2, ...$

$$
\begin{aligned}
\partial_{tt} u_1^k - c^2(\boldsymbol{x}) \triangle u_1^k &= f, && \text{in } \Omega_1, \\
u_1^k(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_1, \\
\partial_t u_1^k(\boldsymbol{x}, 0) &= v_0(\boldsymbol{x}), && \text{in } \Omega_1, \\
u_1^k &= g, && \text{on } \partial\Omega_1 \setminus \Gamma, \\
u_1^k &= h^{k-1}, && \text{on } \Gamma,
\end{aligned}
\qquad
\begin{aligned}
\partial_{tt} u_2^k - c^2(\boldsymbol{x}) \triangle u_2^k &= f, && \text{in } \Omega_2, \\
u_2^k(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_2, \\
\partial_t u_2^k(\boldsymbol{x}, 0) &= v_0(\boldsymbol{x}), && \text{in } \Omega_2, \\
\partial_{n_2} u_2^k &= -\partial_{n_1} u_1^k, && \text{on } \Gamma, \\
u_2^k &= g, && \text{on } \partial\Omega_2 \setminus \Gamma,
\end{aligned}
$$

$$
h^k(\boldsymbol{x}, t) = \theta u_2^k \big|_{\Gamma \times (0,T)} + (1 - \theta) h^{k-1}(\boldsymbol{x}, t), \tag{2.4.2}
$$

where $\theta \in (0, 1]$ is a relaxation parameter.

2.4.1.1 Convergence analysis

We analyze the convergence of the DNWR algorithm (2.4.2) for the 1D wave equation with constant wave speed c, i.e., $c(\boldsymbol{x}) = c$, with $\Omega = (-a, b)$, $\Omega_1 = (-a, 0)$ and $\Omega_2 = (0, b)$. By linearity, it suffices to study the error equations, $f(\boldsymbol{x}, t) = 0$, $g(\boldsymbol{x}, t) = 0$, $u_0(\boldsymbol{x}) = v_0(\boldsymbol{x}) = 0$ in (2.4.2), and to examine convergence to zero.

Our convergence analysis is again based on Laplace transforms. Applying a Laplace transform to the DNWR algorithm (2.4.2) in 1D, we obtain for the transformed error

equations

$$\begin{aligned}(s^2 - c^2\partial_{xx})\hat{u}_1^k = 0, \quad &\text{in } (-a,0), \quad (s^2 - c^2\partial_{xx})\hat{u}_2^k = 0, \quad \text{in } (0,b),\\ \hat{u}_1^k(-a,s) = 0, \quad &\partial_x\hat{u}_2^k(0,s) = \partial_x\hat{u}_1^k(0,s),\\ \hat{u}_1^k(0,s) = \hat{h}^{k-1}(s), \quad &\hat{u}_2^k(b,s) = 0,\\ \hat{h}^k(s) = \theta\hat{u}_2^k(0,s) &+ (1-\theta)\hat{h}^{k-1}(s).\end{aligned} \tag{2.4.3}$$

Solving the two-point boundary value problems in (2.4.3), we get

$$\hat{u}_1^k(x,s) = \frac{\hat{h}^{k-1}(s)}{\sinh(as/c)}\sinh\left((x+a)\frac{s}{c}\right), \hat{u}_2^k(x,s) = \hat{h}^{k-1}(s)\frac{\coth(as/c)}{\cosh(bs/c)}\sinh\left((x-b)\frac{s}{c}\right),$$

and inserting them into the updating condition (last line in (2.4.3)), we get by induction

$$\hat{h}^k(s) = \left[1 - \theta - \theta\coth(as/c)\tanh(bs/c)\right]^k \hat{h}^0(s), \quad k = 1,2,\ldots \tag{2.4.4}$$

2.4.1.2 Kernel estimate and convergence theorems

We present in this section the convergence results both for symmetric and asymmetric subdomain lengths. The main convergence result given in Theorem 20 is based on a simplification of a kernel arising in the Laplace transform of the DNWR algorithm.

Theorem 18 (Convergence of DNWR, symmetric decomposition)**.** *For a symmetric decomposition, $a = b$, convergence is linear for the DNWR (2.4.2) with $\theta \in (0,1)$, $\theta \neq 1/2$. If $\theta = 1/2$, convergence is achieved in two iterations.*

Proof. For $a = b$, equation (2.4.4) reduces to $\hat{h}^k(s) = (1-2\theta)^k\hat{h}^0(s)$, which has the simple back transform $h^k(t) = (1-2\theta)^k h^0(t)$. Thus for the DNWR method, the convergence is linear for $0 < \theta < 1, \theta \neq \frac{1}{2}$. For $\theta = 1/2$, we have $h^1(t) = 0$. Hence, one more iteration produces the desired solution on the whole domain. □

We next analyze the case of an asymmetric decomposition, $a \neq b$. Defining

$$G_b^a(s) := \coth(as/c)\tanh(bs/c) - 1, \tag{2.4.5}$$

we obtain for (2.4.4)

$$\hat{h}^k(s) = \left\{(1-2\theta) - \theta G_b^a(s)\right\}^k \hat{h}^0(s). \tag{2.4.6}$$

Now if $\theta = 1/2$, we see that the linear factor in (2.4.6) vanishes, and convergence will be governed by convolutions of $G_b^a(s)$. We show next that this choice also gives finite step convergence, but the number of steps depends on the length of the time window T.

Lemma 19. *Let $a, b > 0$ and $s \in \mathbb{C}$, with $\mathrm{Re}(s) > 0$. Then, we have the identity*

$$G_b^a(s) = 2\sum_{m=1}^{\infty} e^{-2ams/c} - 2\sum_{n=1}^{\infty}(-1)^{n-1}e^{-2bns/c} - 4\sum_{n=1}^{\infty}\sum_{m=1}^{\infty}(-1)^{n-1}e^{-2(bn+am)s/c}.$$

Proof. Using that $\left|e^{-2bs/c}\right| < 1$ for $\mathrm{Re}(s) > 0$, we expand $\left(1+e^{-2bs/c}\right)^{-1}$ into an infinite binomial series to obtain

$$\tanh\left(\frac{bs}{c}\right) = \frac{e^{\frac{bs}{c}} - e^{-\frac{bs}{c}}}{e^{\frac{bs}{c}} + e^{-\frac{bs}{c}}} = \left(1-e^{-\frac{2bs}{c}}\right)\left(1+e^{-\frac{2bs}{c}}\right)^{-1} = 1 - 2\sum_{n=1}^{\infty}(-1)^{n-1}e^{-\frac{2bns}{c}}.$$

Similarly, we get $\coth(as/c) = 1+2\sum_{m=1}^{\infty} e^{-\frac{2ams}{c}}$, and multiplying the two and subtracting 1, we obtain the expression for $G_b^a(s)$ in the Lemma. □

Theorem 20 (Convergence of DNWR, asymmetric decomposition)**.** *Let $\theta = 1/2$. Then the DNWR algorithm (2.4.2) converges in at most $k+1$ iterations for two subdomains of lengths $a \neq b$, if the time window length T satisfies $T/k \leq 2\min\{a/c, b/c\}$, where c is the wave speed.*

Proof. With $\theta = 1/2$ we obtain from (2.4.6) for $k = 1, 2, \ldots$

$$\begin{aligned}\hat{h}^k(s) = \left(-\frac{1}{2}\right)^k \left\{G_b^a(s)\right\}^k \hat{h}^0(s) = \Bigg[-e^{-\frac{2as}{c}} + e^{-\frac{2bs}{c}} + \Bigg(\sum_{n>1}^{\infty}(-1)^{n-1}e^{-\frac{2bns}{c}} \\ - \sum_{m>1}^{\infty} e^{-\frac{2ams}{c}} + 2\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}(-1)^{n-1}e^{-\frac{2(am+bn)s}{c}}\Bigg)\Bigg]^k \hat{h}^0(s) = (-1)^k e^{-\frac{2aks}{c}}\hat{h}^0(s) \\ + e^{-\frac{2bks}{c}}\hat{h}^0(s) + \left(\sum_{l>k}^{\infty} p_l^{(k)} e^{-\frac{2bls}{c}} + \sum_{l>k}^{\infty} q_l^{(k)} e^{-\frac{2als}{c}} + \sum_{m+n\geq k} r_{m,n}^{(k)} e^{-\frac{2(am+bn)s}{c}}\right)\hat{h}^0(s), \quad (2.4.7)\end{aligned}$$

$p_l^{(k)}, q_l^{(k)}, r_{m,n}^{(k)}$ being the corresponding coefficients. Using the property L2 of Laplace transform[¶]

$$\mathcal{L}^{-1}\left\{e^{-\alpha s}\hat{g}(s)\right\} = H(t-\alpha)g(t-\alpha), \quad (2.4.8)$$

we obtain

$$\begin{aligned}h^k(t) = (-1)^k h^0(t-2ak/c)H(t-2ak/c) + h^0(t-2bk/c)H(t-2bk/c) \\ + \sum_{l>k}^{\infty} p_l^{(k)} h^0(t-2bl/c)H(t-2bl/c) + \sum_{l>k}^{\infty} q_l^{(k)} h^0(t-2al/c)H(t-2al/c) \\ + \sum_{m+n\geq k} r_{m,n}^{(k)} h^0(t-2(am+bn)/c)H(t-2(am+bn)/c).\end{aligned}$$

Now if we choose our time window such that $T \leq 2k\min\left\{\frac{a}{c}, \frac{b}{c}\right\}$, then $h^k(t) = 0$, and therefore one more iteration produces the desired solution on the entire domain. □

2.4.1.3 Numerical illustration

Having obtained convergence bounds at the continuous level in the previous section, we perform now numerical experiments to measure the actual convergence rate of the

[¶]For more details, see Appendix A

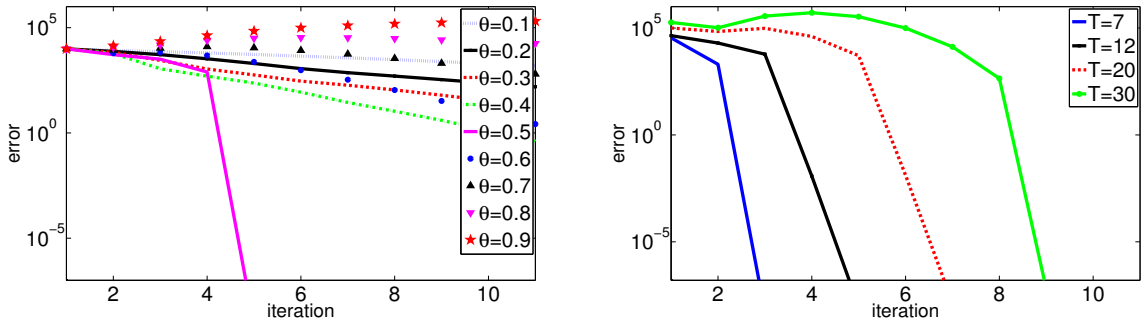


Figure 2.4.1: Convergence of DNWR for various values of θ and $T = 16$ on the left, and for various lengths T of the time window and $\theta = 1/2$ on the right

discretized DNWR and NNWR algorithms for the model problem

$$\begin{aligned} \partial_{tt}u - c^2\partial_{xx}u &= 0, & & x \in (-3,2), t > 0, \\ u(x,0) = 0,\ u_t(x,0) &= xe^{-x}, & & -3 < x < 2, \\ u(-3,t) = -3e^3t,\ u(2,t) &= 2te^{-2}, & & t > 0, \end{aligned} \tag{2.4.9}$$

with $c = 1$, $\Omega_1 = (-3,0)$ and $\Omega_2 = (0,2)$, so that $a = 3$ and $b = 2$ in (2.4.3)-(2.4.4). We discretize the wave equation using centered finite differences in both space and time (Leapfrog scheme) on a grid with $\Delta x = \Delta t = 2\times 10^{-2}$ as follows: for $n \geq 1$ and $2 \leq i \leq J-1$

$$u(x_i,t_{n+1}) := u_i^{n+1} \approx 2\left(1 - \left(c\frac{\Delta t}{\Delta x}\right)^2\right)u_i^n - u_i^{n-1} + \left(c\frac{\Delta t}{\Delta x}\right)^2\left(u_{i+1}^n + u_{i-1}^n\right),$$

where $t_n = n\Delta t$, for $n = 0,1,\ldots$ and $x_i = x_1 + (i-1)\Delta x$, $\Delta x = (x_J - x_1)/(J-1)$ for $i = 2,\ldots,J-1$. The scheme at $t = 0$ is given by

$$u(x_i,t_1) := u_i^1 \approx u_i^0 + \frac{1}{2}\left(c\frac{\Delta t}{\Delta x}\right)^2\left(u_{i+1}^0 - 2u_i^0 + u_{i-1}^0\right) + \Delta t \cdot g_i,\ i = 2,\ldots,J-1,$$

where $u_t(x,0) = g(x)$ with $g_i = g(x_i)$. The error is calculated by $\|u - u^k\|_{L^\infty(0,T;L^2(\Omega))}$, where u is the discrete monodomain solution and u^k is the discrete solution in the k-th iteration. We test the DNWR algorithm by choosing $h^0(t) = t^2, t \in (0,T]$ as an initial guess. In Figure 2.4.1, we show the convergence behavior for different values of the parameter θ for $T = 16$ on the left, and on the right for the best parameter $\theta = 1/2$ for different time window length T. Note that, the iteration number starts from one in both the experiments. It is clear from the fact that, the iteration number k of the numerical errors in (2.4.2) begins from one, whereas the initial guess h^k along the subdomain interface is given for $k = 0$.

2.4.2 DNWR for multiple subdomains

We define the Dirichlet-Neumann Waveform Relaxation method with many subdomains for the model problem (2.4.1) on the space-time domain $\Omega \times (0,T)$ with Dirichlet data

given on $\partial\Omega$. Suppose the spatial domain Ω is partitioned into $2m+1$ non-overlapping subdomains $\Omega_i, i = 1, \ldots, 2m+1$, as illustrated in Figure 2.3.10 in Subsection 2.3.2.2. For $i = 1, \ldots, 2m$, set $\Gamma_i := \partial\Omega_i \cap \partial\Omega_{i+1}$. Similarly as the parabolic equation we define $\Gamma_0 = \Gamma_{2m+1} = \emptyset$. We denote by $\boldsymbol{n}_{i,j}$ the unit outward normal for Ω_i on the interface $\Gamma_j, j = i-1, i$ (for Ω_1, Ω_{2m+1} we have only $\boldsymbol{n}_{1,2}$ and $\boldsymbol{n}_{2m+1,2m}$ respectively). We define the DNWR method as in the arrangement A3, but one can also consider as in A1 and A2 from Subsection 2.3.2.

Given initial Dirichlet traces $g_i^0(\boldsymbol{x}, t)$ along the interfaces $\Gamma_i \times (0, T)$, $i = 1, \ldots, 2m$, the DNWR algorithm consists of the following computation for $k = 1, 2, \ldots$

$$\begin{array}{rcll} \partial_{tt} u_{m+1}^k - c^2(\boldsymbol{x})\triangle u_{m+1}^k &=& f, & \text{in } \Omega_{m+1}, \\ u_{m+1}^k(\boldsymbol{x}, 0) &=& u_0(\boldsymbol{x}), & \text{in } \Omega_{m+1}, \\ \partial_t u_{m+1}^k(\boldsymbol{x}, 0) &=& v_0(\boldsymbol{x}), & \text{in } \Omega_{m+1}, \\ u_{m+1}^k &=& g, & \text{on } \partial\Omega \cap \partial\Omega_{m+1}, \\ u_{m+1}^k &=& g_i^{k-1}, & \text{on } \Gamma_i, i = m, m+1, \end{array} \tag{2.4.10}$$

and then for $m \geq i \geq 1$ and $m+2 \leq j \leq 2m+1$

$$\begin{array}{llll} \partial_{tt} u_i^k = c^2(\boldsymbol{x})\triangle u_i^k + f, & \text{in } \Omega_i, & \partial_{tt} u_j^k = c^2(\boldsymbol{x})\triangle u_j^k + f, & \text{in } \Omega_j, \\ u_i^k(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), & \text{in } \Omega_i, & u_j^k(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), & \text{in } \Omega_j, \\ \partial_t u_i^k(\boldsymbol{x}, 0) = v_0(\boldsymbol{x}), & \text{in } \Omega_i, & \partial_t u_j^k(\boldsymbol{x}, 0) = v_0(\boldsymbol{x}), & \text{in } \Omega_j, \\ u_i^k = \widetilde{g}_i^{k-1}, & \text{on } \partial\Omega_i \setminus \Gamma_i, & \partial_{\boldsymbol{n}_{j,j-1}} u_j^k = -\partial_{\boldsymbol{n}_{j-1,j}} u_{j-1}^k, & \text{on } \Gamma_{j-1}, \\ \partial_{\boldsymbol{n}_{i,i+1}} u_i^k = -\partial_{\boldsymbol{n}_{i+1,i}} u_{i+1}^k, & \text{on } \Gamma_i, & u_j^k = \check{g}_j^{k-1}, & \text{on } \partial\Omega_j \setminus \Gamma_{j-1}, \end{array} \tag{2.4.11}$$

with the update conditions along the interfaces

$$\begin{array}{l} g_i^k(\boldsymbol{x}, t) = \theta u_i^k\big|_{\Gamma_i \times (0,T)} + (1-\theta) g_i^{k-1}(\boldsymbol{x}, t), \ 1 \leq i \leq m, \\ g_j^k(\boldsymbol{x}, t) = \theta u_{j+1}^k\big|_{\Gamma_j \times (0,T)} + (1-\theta) g_j^{k-1}(\boldsymbol{x}, t), \ m+1 \leq j \leq 2m, \end{array} \tag{2.4.12}$$

where $\theta \in (0, 1]$, and the functions $\widetilde{g}_i^{k-1}, i = 1, \ldots, m$ and $\check{g}_j^{k-1}, j = m+2, \ldots, 2m+1$ are given by:

$$\widetilde{g}_i^{k-1} = \begin{cases} g, & \text{on } \partial\Omega \cap \partial\Omega_i, \\ g_{i-1}^{k-1}, & \text{on } \Gamma_{i-1}, \end{cases} \quad \check{g}_j^{k-1} = \begin{cases} g, & \text{on } \partial\Omega \cap \partial\Omega_j, \\ g_j^{k-1}, & \text{on } \Gamma_j. \end{cases}$$

We present the convergence result of this DNWR algorithm (2.4.10)-(2.4.11)-(2.4.12) in the next subsection for the 1D wave equation with constant wave speed c. The illustration of the result with some numerical experiments is given in Subsection 2.4.2.2. We also show with numerical test that, the convergence result for the wave equation is exactly the same for other two rearrangements A1 and A2.

2.4.2.1 Convergence analysis

As like in Subsection 2.3.2.3, we split the domain $\Omega := (0, L)$ into non-overlapping subdomains $\Omega_i := (x_{i-1}, x_i)$, $i = 1, \ldots, 2m+1$, and define the subdomain length $h_i := x_i - x_{i-1}$. Also, define the physical boundary conditions as $u(0, t) = g_0(t)$ and $u(L, t) = g_L(t)$, which in turn become zeros as we consider the error equations, $f(x, t) = 0, g_0(t) = g_L(t) = 0 = u_0(x)$. We take $\{g_i^0(t)\}_{i=1}^{2m}$ as initial guesses along the interfaces $\{x = x_i\} \times$

$(0,T)$, and for sake of consistency we denote $g_0^k(t) = g_{2m+1}^k = 0$ for all k corresponding to the physical boundaries. Denoting by $\left\{w_i^k(t)\right\}_{i=1}^{2m}$ for $k = 1, 2\ldots$ the Neumann traces along the interfaces, we compute

$$\begin{aligned} \partial_{tt}u_{m+1}^k - c^2\partial_{xx}u_{m+1}^k &= 0, && x \in \Omega_{m+1}, \\ u_{m+1}^k(x,0) &= 0, && x \in \Omega_{m+1}, \\ \partial_t u_{m+1}^k(x,0) &= 0, && x \in \Omega_{m+1}, \\ u_{m+1}^k(x_m,t) &= g_m^{k-1}(t), \\ u_{m+1}^k(x_{m+1},t) &= g_{m+1}^{k-1}(t). \end{aligned} \tag{2.4.13}$$

and then for $m \geq i \geq 1$ and $m+2 \leq j \leq 2m+1$

$$\begin{aligned} \partial_{tt}u_i^k - c^2\partial_{xx}u_i^k = 0, && x \in \Omega_i, & \partial_{tt}u_j^k - c^2\partial_{xx}u_j^k = 0, && x \in \Omega_j, \\ u_i^k(x,0) = 0, && x \in \Omega_i, & u_j^k(x,0) = 0, && x \in \Omega_j, \\ \partial_t u_i^k(x,0) = 0, && x \in \Omega_i, & \partial_t u_j^k(x,0) = 0, && x \in \Omega_j, \\ u_i^k(x_{i-1},t) = g_{i-1}^{k-1}(t), & & -\partial_x u_j^k(x_{j-1},t) = w_{j-1}^k(t), \\ \partial_x u_i^k(x_i,t) = w_i^k(t), & & u_j^k(x_j,t) = g_j^{k-1}(t), \end{aligned} \tag{2.4.14}$$

and finally the update conditions with the parameter $\theta \in (0,1]$

$$\begin{aligned} g_i^k(t) &= \theta u_i^k(x_i,t) + (1-\theta)g_i^{k-1}(t), \quad w_i^k(t) = \partial_x u_{i+1}^k(x_i,t),\ 1 \leq i \leq m, \\ g_j^k(t) &= \theta u_{j+1}^k(x_j,t) + (1-\theta)g_j^{k-1}(t), \ w_j^k(t) = -\partial_x u_j^k(x_j,t),\ m+1 \leq j \leq 2m. \end{aligned} \tag{2.4.15}$$

Theorem 21 (Convergence of DNWR for multiple subdomains)**.** *Let $\theta = 1/2$. Then the DNWR algorithm (2.4.13)-(2.4.14)-(2.4.15) converges in at most $k+1$ iterations for multiple subdomains, if the time window length T satisfies $T/k \leq h_{\min}/c$, where c is the wave speed.*

Proof. We apply the Laplace transform to the homogeneous Dirichlet subproblem in (2.4.13) to get

$$s^2\hat{u}_{m+1}^k - c^2\hat{u}_{m+1,xx}^k = 0, \quad \hat{u}_{m+1}^k(x_m,s) = \hat{g}_m^{k-1}(s), \quad \hat{u}_{m+1}^k(x_{m+1},s) = \hat{g}_{m+1}^{k-1}(s),$$

Define $\sigma_i := \sinh(h_i s/c)$ and $\gamma_i := \cosh(h_i s/c)$. Then the subdomain (2.4.13) solution becomes

$$\hat{u}_{m+1}^k(x,s) = \frac{1}{\sigma_{m+1}}\left(\hat{g}_{m+1}^{k-1}(s)\sinh\left((x-x_m)s/c\right) + \hat{g}_m^{k-1}(s)\sinh\left((x_{m+1}-x)s/c\right)\right).$$

One thus can write the update conditions (2.4.15) in a similar way as in the proof of Theorem 17 by changing $\sqrt{s/\nu}$ to s/c. We choose

$$\bar{g}_i^k := \gamma_i\hat{g}_i^k, \ \bar{w}_i^k := \frac{\hat{w}_i^k}{s/c}\sigma_i, \quad \text{for } 1 \leq i \leq m$$

and

$$\bar{g}_j^k := \gamma_{j+1}\hat{g}_j^k, \ \bar{w}_j^k := \frac{\hat{w}_j^k}{s/c}\sigma_{j+1}, \quad \text{for } m+1 \leq j \leq 2m$$

with $\gamma_0 = \gamma_{2m+2} = 1$ in the corresponding equations of (2.3.43)-(2.3.44) with s/c to get

$$\bar{g}_i^k = \tfrac{1}{2\gamma_{i-1}}\bar{g}_{i-1}^{k-1} + \tfrac{1}{2}\bar{g}_i^{k-1} + \tfrac{1}{2}\bar{w}_i^k,\ 1 \leq i \leq m;\ \bar{g}_j^k = \tfrac{1}{2}\bar{w}_j^k + \tfrac{1}{2}\bar{g}_j^{k-1} + \tfrac{1}{2\gamma_{j+2}}\bar{g}_{j+1}^{k-1},\ m+1 \leq j \leq 2m,$$

$$\bar{w}_i^k = -\frac{\sigma_i\sigma_{i+1}}{\gamma_i\gamma_{i+1}}\bar{g}_i^{k-1} + \frac{\sigma_i}{\sigma_{i+1}\gamma_{i+1}}\bar{w}_{i+1}^k, 1 \le i \le m-1;\ \bar{w}_m^k = -\frac{\sigma_m\gamma_{m+1}}{\gamma_m\sigma_{m+1}}\bar{g}_m^{k-1} + \frac{\sigma_m}{\sigma_{m+1}\gamma_{m+2}}\bar{g}_{m+1}^{k-1},$$
$$\bar{w}_{m+1}^k = \frac{\sigma_{m+2}}{\gamma_m\sigma_{m+1}}\bar{g}_m^{k-1} - \frac{\gamma_{m+1}\sigma_{m+2}}{\sigma_{m+1}\gamma_{m+2}}\bar{g}_{m+1}^{k-1};\quad \bar{w}_j^k = \frac{\sigma_{j+1}}{\gamma_j\sigma_j}\bar{w}_{j-1}^k - \frac{\sigma_j\sigma_{j+1}}{\gamma_j\gamma_{j+1}}\bar{g}_j^{k-1},\ m+2 \le j \le 2m.$$

Therefore as in the proof of Theorem 17 we can write in matrix form

$$\begin{pmatrix} \bar{g}_1^k \\ \vdots \\ \bar{g}_m^k \\ \bar{g}_{m+1}^k \\ \vdots \\ \bar{g}_{2m}^k \end{pmatrix} = P \begin{pmatrix} \bar{g}_1^{k-1} \\ \vdots \\ \bar{g}_m^{k-1} \\ \bar{g}_{m+1}^{k-1} \\ \vdots \\ \bar{g}_{2m}^{k-1} \end{pmatrix},$$

where

$$P = \left(\begin{array}{c|c} K_1 & K_2 \\ \hline K_3 & K_4 \end{array}\right)_{2m\times 2m}, K_1 = \begin{bmatrix} \frac{\gamma_{1,2}}{2\gamma_1\gamma_2} & -\frac{\sigma_1\sigma_3}{2\gamma_2^2\gamma_3} & \cdots & \frac{-\sigma_1\sigma_m}{2\gamma_2\ldots\gamma_{m-2}\gamma_{m-1}^2\gamma_m} & \frac{-\sigma_1\gamma_{m+1}}{2\gamma_2\ldots\gamma_{m-1}\gamma_m^2\sigma_{m+1}} \\ \frac{1}{2\gamma_1} & \frac{\gamma_{2,3}}{2\gamma_2\gamma_3} & \ddots & \frac{-\sigma_2\sigma_m}{2\gamma_3\ldots\gamma_{m-2}\gamma_{m-1}^2\gamma_m} & \frac{-\sigma_2\gamma_{m+1}}{2\gamma_3\ldots\gamma_{m-1}\gamma_m^2\sigma_{m+1}} \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & 0 & \frac{1}{2\gamma_{m-2}} & \frac{\gamma_{m-1,m}}{2\gamma_{m-1}\gamma_m} & -\frac{\sigma_{m-1}\gamma_{m+1}}{2\gamma_m^2\sigma_{m+1}} \\ 0 & \cdots & 0 & \frac{1}{2\gamma_{m-1}} & \frac{\sigma_{m+1,m}}{2\sigma_{m+1}\gamma_m} \end{bmatrix},$$

$$K_2 = \begin{bmatrix} \frac{\sigma_1}{2\gamma_2\ldots\gamma_m\sigma_{m+1}\gamma_{m+2}} & 0 & \cdots & \cdots & 0 \\ \frac{\sigma_2}{2\gamma_3\ldots\gamma_m\sigma_{m+1}\gamma_{m+2}} & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\sigma_{m-1}}{2\gamma_m\sigma_{m+1}\gamma_{m+2}} & \vdots & \vdots & \vdots & \vdots \\ \frac{\sigma_m}{2\sigma_{m+1}\gamma_{m+2}} & 0 & \cdots & \cdots & 0 \end{bmatrix}, K_3 = \begin{bmatrix} 0 & \cdots & \cdots & 0 & \frac{\sigma_{m+2}}{2\gamma_m\sigma_{m+1}} \\ \vdots & \vdots & \vdots & \vdots & \frac{\sigma_{m+3}}{2\gamma_m\sigma_{m+1}\gamma_{m+2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \frac{\sigma_{2m}}{2\gamma_m\sigma_{m+1}\gamma_{m+2}\ldots\gamma_{2m-1}} \\ 0 & \cdots & \cdots & 0 & \frac{\sigma_{2m+1}}{2\gamma_m\sigma_{m+1}\gamma_{m+2}\ldots\gamma_{2m}} \end{bmatrix},$$

$$K_4 = \begin{bmatrix} \frac{\sigma_{m+1,m+2}}{2\sigma_{m+1}\gamma_{m+2}} & \frac{1}{2\gamma_{m+3}} & 0 & \cdots & 0 \\ -\frac{\sigma_{m+3}\gamma_{m+1}}{2\sigma_{m+1}\gamma_{m+2}^2} & \frac{\gamma_{m+2,m+3}}{2\gamma_{m+2}\gamma_{m+3}} & \frac{1}{2\gamma_{m+4}} & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \frac{-\sigma_{2m}\gamma_{m+1}}{2\sigma_{m+1}\gamma_{m+2}^2\gamma_{m+3}\ldots\gamma_{2m-1}} & \frac{-\sigma_{2m}\sigma_{m+2}}{2\gamma_{m+2}\gamma_{m+3}^2\gamma_{m+4}\ldots\gamma_{2m-1}} & \ddots & \frac{\gamma_{2m-1,2m}}{2\gamma_{2m-1}\gamma_{2m}} & \frac{1}{2\gamma_{2m+1}} \\ \frac{-\sigma_{2m+1}\gamma_{m+1}}{2\sigma_{m+1}\gamma_{m+2}^2\gamma_{m+3}\ldots\gamma_{2m}} & \frac{-\sigma_{2m+1}\sigma_{m+2}}{2\gamma_{m+2}\gamma_{m+3}^2\gamma_{m+4}\ldots\gamma_{2m}} & \cdots & -\frac{\sigma_{2m-1}\sigma_{2m+1}}{2\gamma_{2m-1}\gamma_{2m}^2} & \frac{\gamma_{2m,2m+1}}{2\gamma_{2m}\gamma_{2m+1}} \end{bmatrix},$$

with $\gamma_{i,j} = \cosh\left((h_i - h_j)s/c\right), \sigma_{i,j} = \sinh\left((h_i - h_j)s/c\right)$. Therefore the updating conditions become

$$\hat{g}_i^k(s) = \sum_{l=i-1}^{m+1} \hat{k}_{i,l}\hat{g}_l^{k-1}(s), 1 \le i \le m;\quad \hat{g}_j^k(s) = \sum_{l=m}^{j+1} \hat{k}_{j,l}\hat{g}_l^{k-1}(s), m+1 \le j \le 2m, \quad (2.4.16)$$

where $\hat{k}_{1,0} = 0$; for $i+1 \le l < m$, $\hat{k}_{i,i-1} = \frac{1}{2\gamma_i}, \hat{k}_{i,i} = \frac{\gamma_{i,i+1}}{2\gamma_i\gamma_{i+1}}, \hat{k}_{i,l} = -\frac{\sigma_i\sigma_{l+1}}{2\gamma_i\gamma_{i+1}\ldots\gamma_{l+1}}$, $\hat{k}_{i,m} = -\frac{\sigma_i\gamma_{m+1}}{2\gamma_i\ldots\gamma_m\sigma_{m+1}}, \hat{k}_{i,m+1} = \frac{\sigma_i}{2\gamma_i\ldots\gamma_m\sigma_{m+1}}$ for $1 \le i < m$, and $\hat{k}_{2m,2m+1} = 0$; for $m+1 < l \le j-1$, $\hat{k}_{j,j} = \frac{\gamma_{j,j+1}}{2\gamma_j\gamma_{j+1}}, \hat{k}_{j,l} = -\frac{\sigma_{j+1}\sigma_l}{2\gamma_l\gamma_{l+1}\ldots\gamma_{j+1}}, \hat{k}_{j,j+1} = \frac{1}{2\gamma_{j+1}}$, $\hat{k}_{j,m+1} = -\frac{\sigma_{j+1}\gamma_{m+1}}{2\sigma_{m+1}\gamma_{m+2}\ldots\gamma_{j+1}}, \hat{k}_{j,m} = \frac{\sigma_{j+1}}{2\sigma_{m+1}\gamma_{m+2}\ldots\gamma_{j+1}}$ for $m+1 < j \le 2m$. Also, $\hat{k}_{m,m-1} = \frac{1}{2\gamma_m}, \hat{k}_{m,m} = \frac{\sigma_{m+1,m}}{2\sigma_{m+1}\gamma_m}, \hat{k}_{m,m+1} =$

$\frac{\sigma_m}{2\sigma_{m+1}\gamma_m}$ and $\hat{k}_{m+1,m} = \frac{\sigma_{m+2}}{2\sigma_{m+1}\gamma_{m+2}}, \hat{k}_{m+1,m+1} = \frac{\sigma_{m+1,m+2}}{2\sigma_{m+1}\gamma_{m+2}}, \hat{k}_{m+1,m+2} = \frac{1}{2\gamma_{m+2}}$. So by induction on (2.4.16) we can write for $1 \le i \le 2m$

$$\hat{g}_i^k(s) = \sum_{j=1}^{2m} p_{i,j}^n \left(\hat{k}_{1,1}, \hat{k}_{1,2}, \ldots, \hat{k}_{2m,2m-1}, \hat{k}_{2m,2m}\right) \hat{g}_j^{k-n}(s), \tag{2.4.17}$$

where the coefficients $p_{i,j}^n$ are either zero or homogeneous polynomials of degree n. A similar calculation as in (2.4.7) yields for $i+1 \le l < m$ and $1 \le i < m$

$$\hat{k}_{i,i} = \frac{\cosh\left((h_i - h_{i+1})s/c\right)}{2\cosh(h_i s/c)\cosh(h_{i+1}s/c)} = \left(e^{-2h_i s/c} + e^{-2h_{i+1}s/c}\right)\left[1 + \sum_{l=1}^{\infty}(-1)^l e^{-2h_i l s/c} + \sum_{n=1}^{\infty}(-1)^n e^{-2h_{i+1}ns/c} + \sum_{l=1}^{\infty}\sum_{n=1}^{\infty}(-1)^{l+n} e^{-2(lh_i+nh_{i+1})s/c}\right],$$

$$\hat{k}_{i,l} = -\frac{\sinh\left(h_i s/c\right)\sinh\left(h_{l+1}s/c\right)}{2\cosh(h_i s/c)\cosh(h_{i+1}s/c)\ldots\cosh(h_{l+1}s/c)}$$

$$= -2^{l-i-1} e^{-(h_{i+1}+\cdots+h_l)s/c}\left(1 - e^{-2h_i s/c} - e^{-2h_{l+1}s/c} + e^{-2(h_i+h_{l+1})s/c}\right)\prod_{n=i}^{l+1}\left(1 + e^{-2h_n s/c}\right)^{-1},$$

$$\hat{k}_{i,i-1} = \frac{1}{2\cosh(h_i s/c)} = e^{-h_i s/c}\left[1 + \sum_{l=1}^{\infty}(-1)^l e^{-2h_i l s/c}\right],$$

$$\hat{k}_{i,m} = -\frac{\sinh\left(h_i s/c\right)\cosh\left(h_{m+1}s/c\right)}{2\cosh(h_i s/c)\ldots\cosh(h_m s/c)\sinh(h_{m+1}s/c)} = -2^{m-i-1} e^{-(h_{i+1}+\cdots+h_m)s/c}.$$

$$\left(1 - e^{-2h_i s/c} + e^{-2h_{m+1}s/c} - e^{-2(h_i+h_{m+1})s/c}\right)\left(1 - e^{-2h_{m+1}s/c}\right)^{-1}\prod_{l=i}^{m}\left(1 + e^{-2h_l s/c}\right)^{-1},$$

$$\hat{k}_{i,m+1} = \frac{\sinh\left(h_i s/c\right)}{2\cosh(h_i s/c)\ldots\cosh(h_m s/c)\sinh(h_{m+1}s/c)}$$

$$= 2^{m-i} e^{-(h_{i+1}+\cdots+h_{m+1})s/c}\left(1 - e^{-2h_i s/c}\right)\left(1 - e^{-2h_{m+1}s/c}\right)^{-1}\prod_{l=i}^{m}\left(1 + e^{-2h_l s/c}\right)^{-1},$$

$$\hat{k}_{m,m} = \frac{\sinh\left((h_{m+1} - h_m)s/c\right)}{2\cosh(h_m s/c)\sinh(h_{m+1}s/c)} = \left(e^{-2h_m s/c} - e^{-2h_{m+1}s/c}\right)\cdot$$

$$\left[1 + \sum_{l=1}^{\infty}(-1)^l e^{-2h_m l s/c} + \sum_{n=1}^{\infty} e^{-2h_{m+1}ns/c} + \sum_{l=1}^{\infty}\sum_{n=1}^{\infty}(-1)^l e^{-2(lh_m+nh_{m+1})s/c}\right],$$

$$\hat{k}_{m,m+1} = \frac{\sinh\left(h_m s/c\right)}{2\cosh(h_m s/c)\sinh(h_{m+1}s/c)} = \left(e^{-h_{m+1}s/c} - e^{-(2h_m+h_{m+1})s/c}\right)\cdot$$

$$\left[1 + \sum_{l=1}^{\infty}(-1)^l e^{-2h_m l s/c} + \sum_{n=1}^{\infty} e^{-2h_{m+1}ns/c} + \sum_{l=1}^{\infty}\sum_{n=1}^{\infty}(-1)^l e^{-2(lh_m+nh_{m+1})s/c}\right].$$

The argument also holds similarly for other terms. Now using these expressions we can write (2.4.17) as

$$\hat{g}_i^k(s) = \sum_{j=1}^{2m} r_{i,j}^k(s)\, \hat{g}_j^0(s), \tag{2.4.18}$$

where $r_{i,j}^k(s)$ are linear combinations of terms of the form e^{-sz} with $z \geq kh_l/c$ for some $l \in \{1, 2, \ldots 2m+1\}$. Now we use (2.4.8) to back transform (2.4.18) and obtain

$$g_i^k(t) = g_j^0\left(t - \frac{kh_l}{c}\right) H\left(t - \frac{kh_l}{c}\right) + \text{ other terms},$$

for some $j \in \{1, 2, \ldots 2m\}$ and $l \in \{1, 2, \ldots 2m+1\}$. Thus for $T \leq kh_{\min}/c$, we get $g_i^k(t) = 0$ for all i, and the conclusion follows. □

2.4.2.2 Numerical illustration

We perform numerical experiments to see the convergence behavior of the DNWR algorithm with multiple subdomains for the model problem

$$\begin{aligned} \partial_{tt}u - \partial_{xx}u &= 0, && x \in (0,5), t > 0, \\ u(x,0) = 0,\ u_t(x,0) &= 0, && 0 < x < 5, \\ u(0,t) = t^2,\ u(5,t) &= t^2 e^{-t}, && t > 0, \end{aligned}$$

which is discretized again using centered finite differences in both space and time on a grid with $\Delta x = \Delta t = 2{\times}10^{-2}$. We take the initial guesses $g_j^0(t) = t^2, t \in (0,T]$ for $1 \leq j \leq 4$, and consider a decomposition of $(0,5)$ into five unequal subdomains, whose widths h_i are $1, 0.5, 1.5, 1, 1$ respectively, so that $h_{\min} = 0.5$. On the left panel of Figure 2.5.1, we show the convergence for different values of the parameter θ for $T = 5$, and on the right the results for the best parameter $\theta = 1/2$ for different time window length T. These convergence curves justify our convergence result for the arrangement A3 in Theorem 21. We see two-step convergence for $\theta = 1/2$ for a sufficiently small time window T. Coincidentally we observe exactly the same convergence behavior for other two arrangements A1 and A2 from Subsection 2.3.2, see Figure 2.5.2 and Figure 2.5.3 respectively.

2.5 Conclusion

In this chapter, we have introduced a new class of space-time parallel algorithms, namely, the Dirichlet-Neumann waveform relaxation (DNWR) algorithm for parabolic as well as hyperbolic problems. For the one-dimensional heat equation, we proved superlinear convergence for this algorithm for a particular choice of the relaxation parameter, $\theta = 1/2$. We have also presented a relatively superior arrangement of transmission conditions (Dirichlet or Neumann boundary conditions along the interfaces) for a decomposition into many subdomains, and we have given a convergence estimate for that particular arrangement for the heat equation. We also defined the DNWR algorithm for the second order wave equation, and analyzed its convergence properties for the 1D case and two subdomains. We showed using numerical experiments that for a particular choice of the relaxation parameter $\theta = 1/2$, convergence can be achieved in a finite

number of steps for two as well as multiple subdomains. In fact, the DNWR algorithm can be used to solve such problems in only two iterations, choosing the time window lengh T carefully.

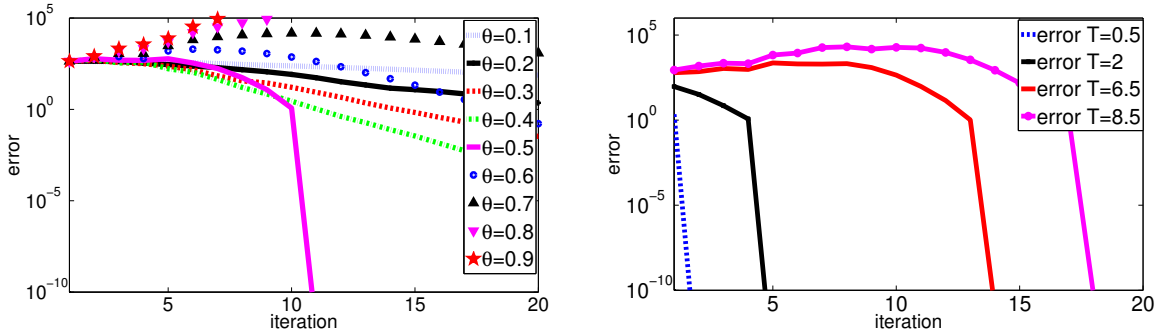


Figure 2.5.1: Convergence of DNWR for the arrangement A3 with various values of θ for $T = 5$ on the left, and for various lengths T of the time window and $\theta = 1/2$ on the right

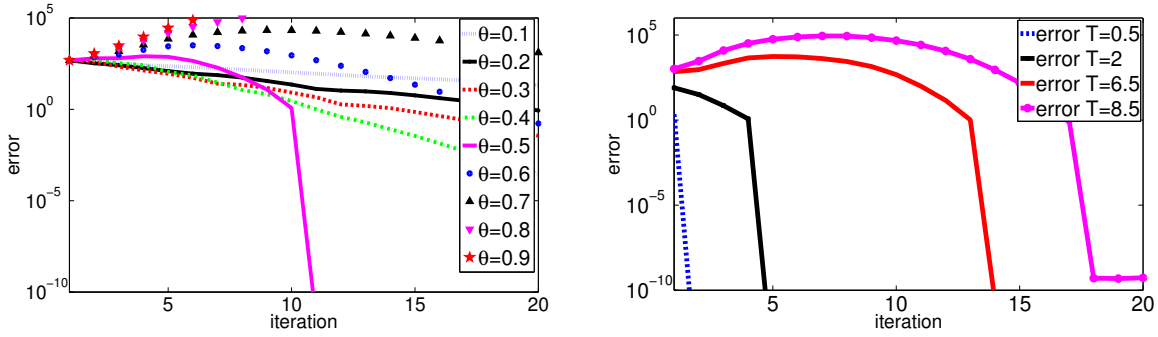


Figure 2.5.2: Convergence of DNWR for the arrangement A1 with various values of θ for $T = 5$ on the left, and for various lengths T of the time window and $\theta = 1/2$ on the right

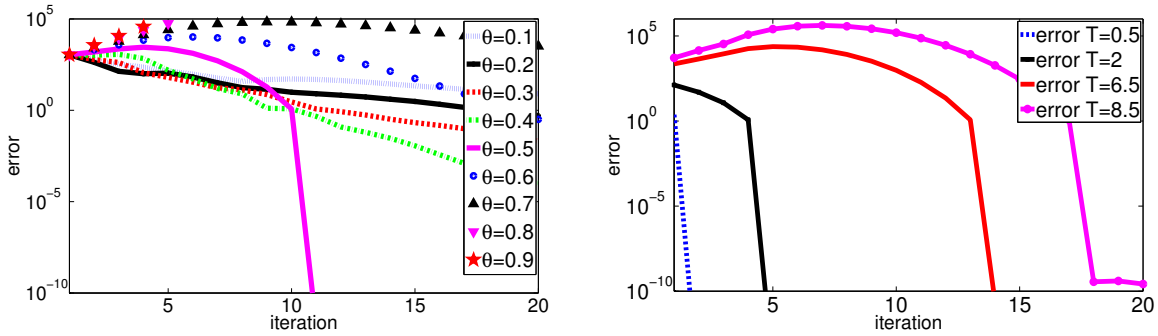


Figure 2.5.3: Convergence of DNWR for the arrangement A2 with various values of θ for $T = 5$ on the left, and for various lengths T of the time window and $\theta = 1/2$ on the right

3

Neumann-Neumann Waveform Relaxation Methods

3.1 Introduction

In this chapter, we introduce and analyze another variant of waveform relaxation (WR) methods based on the Neumann-Neumann algorithm for steady problems. For solving elliptic problems by the Neumann-Neumann domain decomposition method, the iteration involves solving the subdomain problems using Dirichlet interface conditions in the first step, followed by a correction step involving Neumann interface conditions. We extend this method to solve space-time problems, both parabolic and hyperbolic. We call our algorithm Neumann-Neumann Waveform Relaxation (NNWR). In Section 3.3 we present the NNWR algorithm both for two and multiple subdomains for the general problem (2.3.1), and present convergence estimates for the one dimensional heat equation. In Subsection 3.3.3 we show how the analysis of the NNWR can be generalized to higher spatial dimensions, and prove that the convergence estimates do not change. Finally we introduce the NNWR for hyperbolic problems in Section 3.4, and present convergence analysis for 1D and 2D wave equation. We begin our discussion with the Neumann-Neumann algorithm for steady problems.

3.2 Steady-state analysis

We formulate the Neumann-Neumann (NN) algorithm for the steady equation:

$$-\triangle u = f \text{ over a domain } \Omega = (-a, b), \text{ with boundary conditions } u = g.$$

We split Ω into two non-overlapping subdomains $\Omega_1 = (-a, 0)$ and $\Omega_2 = (0, b)$. The Neumann-Neumann algorithm consists of the following steps: given an initial guess λ^0 along the interface $\{x = 0\}$, compute for $\theta \in (0, 1]$ with $k = 1, 2, \ldots$

$$\begin{cases} -\triangle u_1^k = f, & \text{in } \Omega_1, \\ u_1^k(-a) = g(-a), \\ u_1^k(0) = \lambda^{k-1}, \end{cases} \qquad \begin{cases} -\triangle u_2^k = f, & \text{in } \Omega_2, \\ u_2^k(0) = \lambda^{k-1}, \\ u_2^k(b) = g(b), \end{cases} \tag{3.2.1}$$

$$\begin{cases} -\triangle\psi_1^k = 0, & \text{in } \Omega_1, \\ \psi_1^k(-a) = 0, \\ \partial_x\psi_1^k(0) = \partial_x u_1^k(0) - \partial_x u_2^k(0), \end{cases} \qquad \begin{cases} -\triangle\psi_2^k = 0, & \text{in } \Omega_2, \\ \partial_x\psi_2^k(0) = \partial_x u_1^k(0) - \partial_x u_2^k(0), \\ u_2^k(b) = 0, \end{cases} \tag{3.2.2}$$

with the updating condition

$$\lambda^k = \lambda^{k-1} - \theta\left\{\psi_1^k(0) - \psi_2^k(0)\right\}.$$

Considering only the error equations, $f = g = 0$ in (3.2.1)-(3.2.2), we solve both error equations simultaneously to obtain the updating condition:

$$\lambda^k = \left\{1 - \theta\frac{(a+b)^2}{ab}\right\}^k \lambda^0, \ k = 1, 2, \ldots. \tag{3.2.3}$$

Clearly equation (3.2.3) is symmetric with respect to the subdomain lengths a, b.

Lemma 22 (Convergence of NN)**.** *The Neumann-Neumann algorithm* (3.2.1)-(3.2.2) *converges linearly for* $0 < \theta < \frac{2ab}{(a+b)^2}$, $\theta \neq \frac{ab}{(a+b)^2}$. *For* $\theta = \frac{ab}{(a+b)^2}$, *it converges in two iterations.*

Proof. From the equation (3.2.3) it is clear that the NN algorithm diverges if

$$\left|1 - \theta\frac{(a+b)^2}{ab}\right| \geq 1,$$

i.e. if $\theta \geq \frac{2ab}{(a+b)^2}$. Also note that, $\frac{2ab}{(a+b)^2} < 1$ for $a, b > 0$. Therefore the convergence is linear for $0 < \theta < \frac{2ab}{(a+b)^2}$, $\theta \neq \frac{ab}{(a+b)^2}$. Now if $\theta = \frac{ab}{(a+b)^2}$, from (3.2.3) we have $\lambda^1 = 0$, and hence one more iteration produces the desired solution on the whole region. □

Numerical Test:

Figure 3.2.1 represents the convergence behavior of the Neumann-Neumann algorithm, applied to the model problem

$$-\triangle u = e^x, u(-3) = 4, u(2) = 5,$$

over the domain $\Omega = (-3, 2)$. For the numerical experiment, we decompose the domain into $(-3, 0)$ and $(0, 2)$, so that $a = 3, b = 2$ in Lemma 22.

3.3 NNWR for Parabolic problems

We now introduce the NNWR algorithm for the model problem (2.3.1) in a spatial domain Ω, partitioned into two and multiple non-overlapping subdomains.

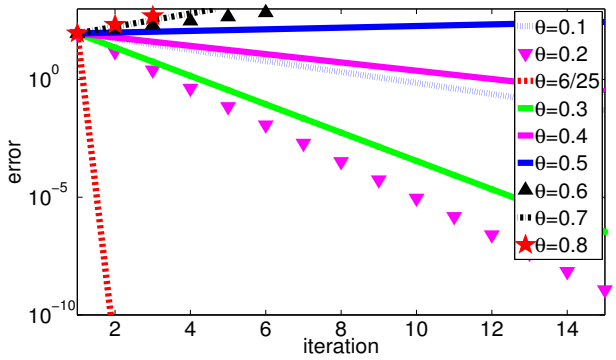


Figure 3.2.1: Convergence of NN for various values relaxation parameters θ for $a > b$

3.3.1 NNWR for two subdomains

The Neumann-Neumann Waveform Relaxation algorithm for two subdomains, as shown in Figure 2.3.1, is given by the following steps: given an initial guess $w^0(\boldsymbol{x},t)$ along the interface $\Gamma \times (0,T)$, compute for $k = 1, 2, \ldots$

$$\begin{aligned}
\partial_t u_1^k - \nabla\cdot\left(\kappa(\boldsymbol{x},t)\nabla u_1^k\right) &= f, && \text{in } \Omega_1,\\
u_1^k(\boldsymbol{x},0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_1,\\
u_1^k &= g, && \text{on } \partial\Omega_1 \setminus \Gamma,\\
u_1^k &= w^{k-1}, && \text{on } \Gamma,
\end{aligned}
\qquad
\begin{aligned}
\partial_t u_2^k - \nabla\cdot\left(\kappa(\boldsymbol{x},t)\nabla u_2^k\right) &= f, && \text{in } \Omega_2,\\
u_2^k(\boldsymbol{x},0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_2,\\
u_2^k &= w^{k-1}, && \text{on } \Gamma,\\
u_2^k &= g, && \text{on } \partial\Omega_2 \setminus \Gamma,
\end{aligned} \tag{3.3.1}$$

$$\begin{aligned}
\partial_t \varphi_1^k &= \nabla\cdot\left(\kappa(\boldsymbol{x},t)\nabla\varphi_1^k\right), && \text{in } \Omega_1,\\
\varphi_1^k(\boldsymbol{x},0) &= 0, && \text{in } \Omega_1,\\
\varphi_1^k &= 0, && \text{on } \partial\Omega_1 \setminus \Gamma,\\
\partial_{\boldsymbol{n}_1}\varphi_1^k &= \partial_{\boldsymbol{n}_1} u_1^k + \partial_{\boldsymbol{n}_2} u_2^k, && \text{on } \Gamma,
\end{aligned}
\qquad
\begin{aligned}
\partial_t \varphi_2^k &= \nabla\cdot\left(\kappa(\boldsymbol{x},t)\nabla\varphi_2^k\right), && \text{in } \Omega_2,\\
\varphi_2^k(\boldsymbol{x},0) &= 0, && \text{in } \Omega_2,\\
\partial_{\boldsymbol{n}_2}\varphi_2^k &= \partial_{\boldsymbol{n}_1} u_1^k + \partial_{\boldsymbol{n}_2} u_2^k, && \text{on } \Gamma,\\
\varphi_2^k &= 0, && \text{on } \partial\Omega_2 \setminus \Gamma,
\end{aligned} \tag{3.3.2}$$

with the updating condition along the interface

$$w^k(\boldsymbol{x},t) = w^{k-1}(\boldsymbol{x},t) - \theta\left(\varphi_1^k\big|_{\Gamma\times(0,T)} + \varphi_2^k\big|_{\Gamma\times(0,T)}\right), \tag{3.3.3}$$

$\theta \in (0,1]$ being a relaxation parameter. We are interested in how the error $w^k(\boldsymbol{x},t) - u\big|_{\Gamma\times(0,T)}$ converges to zero, and by linearity it suffices to consider the error equations, $f(\boldsymbol{x},t) = 0$, $g(\boldsymbol{x},t) = 0$, $u_0(\boldsymbol{x}) = 0$ in (3.3.1), and examine how $w^k(\boldsymbol{x},t)$ converges to zero as $k \to \infty$.

3.3.1.1 Convergence analysis

We present a convergence estimate for algorithm (3.3.1)-(3.3.2)-(3.3.3) for the special case of the heat equation, $\kappa(\boldsymbol{x},t) = \nu$, on the one dimensional domain $\Omega = (-a,b)$ with subdomains $\Omega_1 = (-a,0)$ and $\Omega_2 = (0,b)$ and interface $\Gamma = \{0\}$.

After a Laplace transform, the NNWR algorithm (3.3.1)-(3.3.2)-(3.3.3) for the error equation in the one dimensional heat equation setting becomes

$$\begin{aligned}
(s - \nu\partial_{xx})\hat{u}_1^k &= 0 && \text{on } (-a,0),\\
\hat{u}_1^k(-a,s) &= 0,\\
\hat{u}_1^k(0,s) &= \hat{w}^{k-1}(s),
\end{aligned}
\qquad
\begin{aligned}
(s - \nu\partial_{xx})\hat{u}_2^k &= 0 && \text{on } (0,b),\\
\hat{u}_2^k(0,s) &= \hat{w}^{k-1}(s),\\
\hat{u}_2^k(b,s) &= 0,
\end{aligned} \tag{3.3.4}$$

$$\begin{aligned}
(s-\nu\partial_{xx})\hat{\varphi}_1^k &= 0 && \text{on } (-a,0), \qquad (s-\nu\partial_{xx})\hat{\varphi}_2^k = 0 && \text{on } (0,b),\\
\hat{\varphi}_1^k(-a,s) &= 0, && -\partial_x\hat{\varphi}_2^k = \partial_x\hat{u}_1^k - \partial_x\hat{u}_2^k, && \text{on } \Gamma,\\
\partial_x\hat{\varphi}_1^k &= \partial_x\hat{u}_1^k - \partial_x\hat{u}_2^k, \text{ on } \Gamma, && \hat{\varphi}_2^k(b,s) = 0,
\end{aligned} \tag{3.3.5}$$

and

$$\hat{w}^k(s) = \hat{w}^{k-1}(s) - \theta\left(\hat{\varphi}_1^k(0,s) + \hat{\varphi}_2^k(0,s)\right). \tag{3.3.6}$$

Solving the two-point boundary value problems in the Dirichlet step (3.3.4) yields

$$\hat{u}_1^k(x,s) = \frac{\hat{w}^{k-1}(s)}{\sinh(a\sqrt{s/\nu})}\sinh\left\{(x+a)\sqrt{s/\nu}\right\}$$

and

$$\hat{u}_2^k(x,s) = -\frac{\hat{w}^{k-1}(s)}{\sinh(b\sqrt{s/\nu})}\sinh\left\{(x-b)\sqrt{s/\nu}\right\}.$$

Similarly, we get the solutions of the Neumann step (3.3.5) as:

$$\hat{\varphi}_1^k(x,s) = \frac{\hat{w}^{k-1}(s)\Psi(s)}{\cosh(a\sqrt{s/\nu})}\sinh\left\{(x+a)\sqrt{s/\nu}\right\}$$

and

$$\hat{\varphi}_2^k(x,s) = \frac{\hat{w}^{k-1}(s)\Psi(s)}{\cosh(b\sqrt{s/\nu})}\sinh\left\{(x-b)\sqrt{s/\nu}\right\},$$

where $\Psi(s) = \left\{\coth(a\sqrt{s/\nu}) + \coth(b\sqrt{s/\nu})\right\}$. Therefore, the update condition (3.3.6) becomes

$$\hat{w}^k(s) = \hat{w}^{k-1}(s)\left[1-\theta\left(2+\frac{\tanh(a\sqrt{s/\nu})}{\tanh(b\sqrt{s/\nu})}+\frac{\tanh(b\sqrt{s/\nu})}{\tanh(a\sqrt{s/\nu})}\right)\right]. \tag{3.3.7}$$

3.3.1.2 Convergence theorems

We have the following convergence results for the NNWR algorithm (3.3.1)-(3.3.2)-(3.3.3) with two subdomains from [51].

Theorem 23 (Convergence of NNWR for $a = b$)**.** *When the subdomains are of the same size, $a = b$ in (3.3.4)-(3.3.5)-(3.3.6), the NNWR algorithm converges linearly for $0 < \theta < 1/2$, $\theta \neq 1/4$. For $\theta = 1/4$, it converges in two iterations. Convergence is independent of the time window size T.*

Proof. If $a = b$, then (3.3.7) gives $\hat{w}^k(s) = \hat{w}^{k-1}(s)(1-4\theta)$, which means the method converges to the exact solution in two iterations for $\theta = 1/4$. Also, the algorithm diverges if $|1-4\theta| \geq 1$, i.e. if $\theta \geq 1/2$. Therefore the convergence is linear for $0 < \theta < 1/2$, $\theta \neq 1/4$. □

Thus, we observe that the classical result for elliptic problems also holds for the time-dependent case, if the subdomains are of equal length. Next we focus on the case when the subdomains are unequal, i.e., when $a \neq b$. We study the special case $\theta = 1/4$. For θ not equal to $1/4$, one would require different techniques to analyze the behavior of the NNWR algorithm, as we mentioned already for the DNWR algorithm in Subsection 2.3.1.1.

Theorem 24 (Convergence of NNWR for $a \neq b$)**.** *Let $\theta = 1/4$. Then the error of the NNWR method (3.3.4)-(3.3.5)-(3.3.6) for two subdomains of lengths $a \neq b$ satisfies*

$$\|w^k(\cdot)\|_{L^\infty(0,\infty)} \leq \left(\frac{(a-b)^2}{4ab}\right)^k \|w^0(\cdot)\|_{L^\infty(0,\infty)}. \tag{3.3.8}$$

Moreover, for every finite time interval $(0,T)$, NNWR converges superlinearly with the estimate

$$\|w^k(\cdot)\|_{L^\infty(0,T)} \leq \left(\frac{(a-b)^2}{ab}\right)^k \operatorname{erfc}\left(\frac{\tilde{m}k}{\sqrt{\nu T}}\right) \|w^0(\cdot)\|_{L^\infty(0,T)}, \tag{3.3.9}$$

where $\tilde{m} = \min\{a,b\}$.

Proof. Since (3.3.7) is symmetric with respect to a and b, we will assume without loss of generality that $a > b$. For $\theta = 1/4$, the recurrence (3.3.7) simplifies to

$$\hat{w}^k(s) = -\hat{w}^{k-1}(s)\frac{\sinh^2((a-b)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})\sinh(2b\sqrt{s/\nu})} = -\hat{w}^{k-1}(s)Y(s), \tag{3.3.10}$$

which implies $\hat{w}^k(s) = (-1)^k Y^k(s)\hat{w}^0(s)$. Note that for $\mathrm{Re}(s) > 0$, we have $Y(s) = \mathcal{O}(e^{-4b|s/\nu|^{1/2}})$ as $|s| \to \infty$, i.e., $Y(s)$ decays exponentially as $|s| \to \infty$. Thus, by [18, p. 183], $Y(s)$ is the Laplace transform of an infinitely differentiable function $\varpi_1(t)$. If we now define $\varpi_k(t) := \mathcal{L}^{-1}\{Y^k(s)\}$, then for $t \in (0,T)$, we have

$$|w^k(t)| = \left|\int_0^t w^0(t-\tau)\varpi_k(\tau)\,d\tau\right| \leq \|w^0\|_{L^\infty(0,T)} \int_0^T |\varpi_k(\tau)|\,d\tau. \tag{3.3.11}$$

Thus, to obtain L^∞ convergence estimates, we need bounds on $\int_0^T |\varpi_k(\tau)|\,d\tau$. We first show that $\varpi_k(t) \geq 0$, for $t > 0$. Take an integer m such that $mb < a \leq (m+1)b$. Now for $k < m$, we have the identity

$$\begin{aligned}\sinh^2((a-kb)\sqrt{s/\nu}) - \sinh^2((a-(k+1)b)\sqrt{s/\nu}) \\ = \frac{1}{2}\left[\cosh(2(a-kb)\sqrt{s/\nu}) - 1 - \cosh(2(a-(k+1)b)\sqrt{s/\nu}) + 1\right] \\ = \sinh((2a-(2k+1)b)\sqrt{s/\nu})\sinh(b\sqrt{s/\nu}).\end{aligned}$$

Since $k < m$, we have $0 < 2a - (2k+1)b < 2a$, which gives

$$\begin{aligned}\frac{\sinh^2((a-kb)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})\sinh(2b\sqrt{s/\nu})} = \frac{\sinh((2a-(2k+1)b)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})} \cdot \frac{\sinh(b\sqrt{s/\nu})}{\sinh(2b\sqrt{s/\nu})} \\ + \frac{\sinh^2((a-(k+1)b)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})\sinh(2b\sqrt{s/\nu})}.\end{aligned}$$

Applying this identity repeatedly for $k = 1, \ldots, m-1$ gives

$$\begin{aligned}
Y(s) &= \frac{\sinh^2((a-b)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})\sinh(2b\sqrt{s/\nu})} \\
&= \frac{\sinh^2((a-mb)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})\sinh(2b\sqrt{s/\nu})} + \sum_{k=1}^{m-1} \frac{\sinh((2a-(2k+1)b)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})} \cdot \frac{\sinh(b\sqrt{s/\nu})}{\sinh(2b\sqrt{s/\nu})} \\
&= \frac{1}{2\cosh^2(b\sqrt{s/\nu})} \left[\frac{\sinh^2((a-mb)\sqrt{s/\nu})\cosh(b\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})\sinh(b\sqrt{s/\nu})} \right. \\
&\qquad \left. + \sum_{k=1}^{m-1} \frac{\sinh((2a-(2k+1)b)\sqrt{s/\nu})\cosh(b\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})} \right] \\
&= \frac{1}{4\cosh^2(b\sqrt{s/\nu})} \left[\frac{\sinh((a-mb)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})} \cdot \frac{\sinh((a-mb)\sqrt{s/\nu})}{\sinh(b\sqrt{s/\nu})} \cdot \frac{\cosh(b\sqrt{s/\nu})}{\cosh(a\sqrt{s/\nu})} \right. \\
&\qquad \left. + \sum_{k=1}^{m-1} \left(\frac{\sinh((2a-2kb)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})} + \frac{\sinh((2a-2(k+1)b)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})} \right) \right].
\end{aligned}$$

Set $V(s) = 1/\cosh^2(b\sqrt{s/\nu})$ and let $H(s)$ be the rest. Then since $0 < a - mb \le b < a$, we see that $H(s)$ consists of a sum of products of functions of the form $Q_1(s)$ and $Q_2(s)$ in Lemma 9. Thus, $h(t) = \mathcal{L}^{-1}\{H(s)\}$ is positive. Moreover, since $v(t) = \mathcal{L}^{-1}\{V(s)\}$ is also positive by Lemma 9, we see that $\varpi_1(t) = (v * h)(t)$ is positive by part 1 of Lemma 7, and so $\varpi_k(t)$ is positive. As $\lim_{s\to 0+} V(s) = 1$, we have

$$\lim_{s\to 0+} Y(s) = \lim_{s\to 0+} \frac{\sinh^2((a-b)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})\sinh(2b\sqrt{s/\nu})} = \frac{(a-b)^2}{4ab}.$$

Therefore by Lemma 8, we have

$$\int_0^T |\varpi_k(\tau)|\, d\tau = \int_0^T \varpi_k(\tau)\, d\tau \le \lim_{s\to 0+} Y^k(s) = \left(\lim_{s\to 0+} Y(s) \right)^k = \left(\frac{(a-b)^2}{4ab} \right)^k,$$

which combined with (3.3.11) gives the linear bound (3.3.8).

For $T < \infty$, let $v_k(t) := \mathcal{L}^{-1}\{V^k(s)\}$ and $h_k(t) := \mathcal{L}^{-1}\{H^k(s)\}$. Then since

$$\int_0^\infty h_k(t)\, dt = \lim_{s\to 0+} H^k(s) = (\lim_{s\to 0+} H(s))^k,$$

we have by part 2 of Lemma 7

$$\|\varpi_k\|_{L^1(0,T)} \le \|v_k\|_{L^1(0,T)} \cdot \|h_k\|_{L^1(0,T)} \le \left(\frac{(a-b)^2}{4ab} \right)^k \int_0^T v_k(\tau)\, d\tau. \tag{3.3.12}$$

To bound the remaining integral, let $D(s) = 4^k e^{-2kb\sqrt{s/\nu}} - V^k(s)$. We will show that $d(t) = \mathcal{L}^{-1}\{D(s)\} \ge 0$. We have

$$\begin{aligned}
D(s) &= 4^k e^{-2kb\sqrt{s/\nu}} - \frac{2^{2k}}{(e^{b\sqrt{s/\nu}} + e^{-b\sqrt{s/\nu}})^{2k}} \\
&= 4^k \frac{(1 + e^{-2b\sqrt{s/\nu}})^{2k} - 1}{(e^{b\sqrt{s/\nu}} + e^{-b\sqrt{s/\nu}})^{2k}} = \sum_{m=1}^{2k} \binom{2k}{m} e^{-2bm\sqrt{s/\nu}} V^k(s).
\end{aligned}$$

From [72], we know that $\mathcal{L}^{-1}\{e^{-2bm\sqrt{s/\nu}}\} = \frac{bm}{\sqrt{\pi\nu t^3}}e^{-b^2m^2/\nu t}$ is a positive function for $m \geq 1$. Since $v_k(t) = \mathcal{L}^{-1}\{V^k(s)\}$ is also positive, we see that $d(t)$ is in fact a sum of convolutions of positive functions. Hence $d(t) \geq 0$, as claimed. Thus, we have by part 4 of Lemma 7

$$\begin{aligned}\int_0^T v_k(\tau)\,d\tau \leq \int_0^T (v_k(\tau)+d(\tau))\,d\tau &= \int_0^T 4^k \frac{kb}{\sqrt{\pi\nu\tau^3}} e^{-k^2b^2/\nu\tau}\,d\tau \\ &= \mathcal{L}^{-1}\left\{\frac{4^k e^{-2kb\sqrt{s/\nu}}}{s}\right\} = 4^k \operatorname{erfc}\left(\frac{kb}{\sqrt{\nu T}}\right).\end{aligned}$$

Introducing this into (3.3.12) gives the estimate

$$\|\varpi_k\|_{L^1(0,T)} \leq \left(\frac{(a-b)^2}{ab}\right)^k \operatorname{erfc}\left(\frac{kb}{\sqrt{\nu T}}\right),$$

which tends to zero as $k \to \infty$. Inserting this estimate into (3.3.11) completes the theorem. □

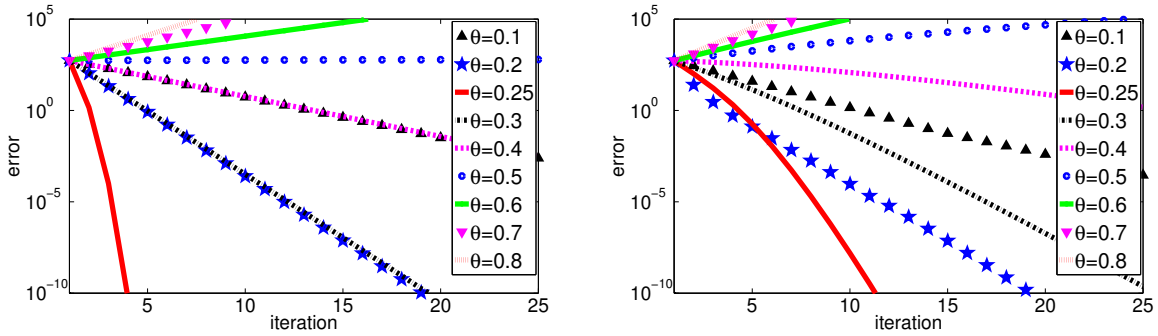


Figure 3.3.1: Convergence of NNWR using various relaxation parameters θ for $T = 6$, on the left for $\kappa(x) = 1$ and on the right for $\kappa(x) = 1 + e^x$

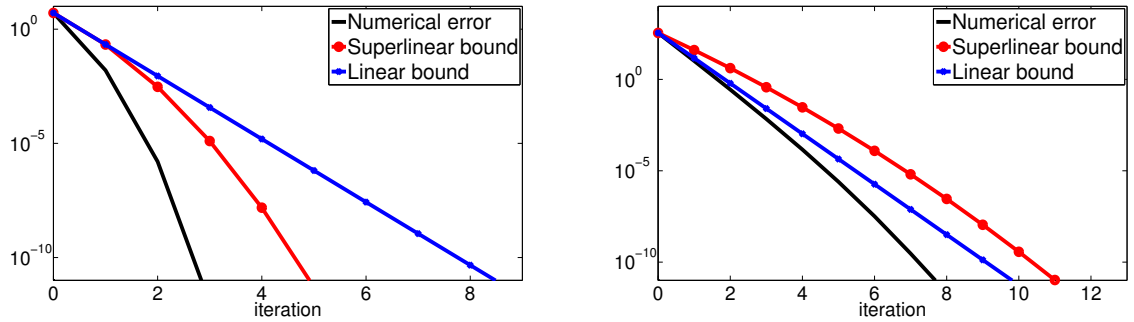


Figure 3.3.2: Comparison of the numerically measured convergence rates and the theoretical error estimates for NNWR for $\kappa(x) = 1$ with $T = 6$ on the left, and $T = 50$ on the right

3.3.1.3 Numerical illustration

We test our results for the NNWR algorithm for the model problem

$$\begin{array}{rcll} \partial_t u - \frac{\partial}{\partial x}\left(\kappa(x)\partial_x u\right) & = & 0, & x \in \Omega, \\ u(x,0) & = & x(x+1)(x+3)(x-2)e^{-x}, & x \in \Omega, \\ u(-3,t) = t,\ u(2,t) & = & te^{-t}, & t > 0. \end{array} \tag{3.3.13}$$

We discretize (3.3.13) like in Subsection 2.3.1.4, using standard centered finite differences in space and backward Euler in time with $\Delta x = 2 \times 10^{-2}$ and $\Delta t = 4 \times 10^{-3}$. We split the spatial domain $\Omega := (-3, 2)$ into two non-overlapping subdomains $\Omega_1 = (-3, 0)$ and $\Omega_2 = (0, 2)$, see Figure 2.3.1. We test the algorithm by choosing $w^0(t) = t^2, t \in [0, T]$ as an initial guess. Figure 3.3.1 gives the convergence curves for $T = 6$ and for different values of the parameter θ for $\kappa(x) = 1$ on the left, and $\kappa(x) = 1 + e^x$ on the right. We see that for a small time window, we get linear convergence for all relaxation parameters θ, except for $\theta = 1/4$, when we observe superlinear convergence.

We now compare the numerical behavior of NNWR with our theoretical estimates of Theorem 24 from Section 3.3.1.2. In Figure 3.3.2, we show for the NNWR algorithm a comparison between the numerically measured convergence for the discretized problem, and the linear and superlinear convergence estimates shown in Theorem 24, for $a = 3$, $b = 2$, $\kappa(x) = 1$. We see that for a short time interval, $T = 6$, the algorithm converges superlinearly, and the superlinear estimate is more accurate than the linear estimate. On the contrary, for the long time interval $T = 50$, the algorithm converges almost linearly, and the linear convergence estimate is now more accurate.

3.3.2 NNWR for multiple subdomains

We now introduce the NNWR algorithm for the model problem (2.3.1) for multiple subdomains. Suppose Ω is partitioned into non-overlapping subdomains $\{\Omega_i,\ 1 \le i \le N\}$, as illustrated in Figure 3.3.3. For $i = 1, \ldots, N$ set $\Gamma_i := \partial\Omega_i \setminus \partial\Omega$, $\Lambda_i := \{j \in \{1, \ldots, N\} : \Gamma_i \cap \Gamma_j \text{ has nonzero measure}\}$ and $\Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j$, so that the interface of Ω_i can be rewritten as $\Gamma_i = \bigcup_{j\in\Lambda_i} \Gamma_{ij}$. We denote by $\boldsymbol{n}_{ij}$ the unit outward normal for Ω_i on the interface Γ_{ij}. The NNWR algorithm starts with an initial guess $w^0_{ij}(\boldsymbol{x}, t)$ along the interfaces $\Gamma_{ij} \times (0, T)$, $j \in \Lambda_i$, $i = 1, \ldots, N$, and then performs the following two-step iteration: at each iteration k, one first solves Dirichlet problems on each Ω_i in parallel,

$$\begin{array}{rcll} \partial_t u_i^k - \nabla \cdot \left(\kappa(\boldsymbol{x}, t)\nabla u_i^k\right) & = & f, & \text{in } \Omega_i, \\ u_i^k(\boldsymbol{x}, 0) & = & u_0(\boldsymbol{x}), & \text{in } \Omega_i, \\ u_i^k & = & g, & \text{on } \partial\Omega_i \setminus \Gamma_i, \\ u_i^k & = & w_{ij}^{k-1}, & \text{on } \Gamma_{ij}, j \in \Lambda_i. \end{array} \tag{3.3.14}$$

One then solves Neumann problems on all subdomains,

$$\begin{array}{rcll} \partial_t \psi_i^k - \nabla \cdot \left(\kappa(\boldsymbol{x}, t)\nabla \psi_i^k\right) & = & 0, & \text{in } \Omega_i, \\ \psi_i^k(\boldsymbol{x}, 0) & = & 0, & \text{in } \Omega_i, \\ \psi_i^k & = & 0, & \text{on } \partial\Omega_i \setminus \Gamma_i, \\ \partial_{\boldsymbol{n}_{ij}} \psi_i^k & = & \partial_{\boldsymbol{n}_{ij}} u_i^k + \partial_{\boldsymbol{n}_{ji}} u_j^k, & \text{on } \Gamma_{ij}, j \in \Lambda_i. \end{array} \tag{3.3.15}$$

The interface values are then updated with the formula

$$w_{ij}^k(\boldsymbol{x}, t) = w_{ij}^{k-1}(\boldsymbol{x}, t) - \theta\left(\psi_i^k\big|_{\Gamma_{ij}\times(0,T)} + \psi_j^k\big|_{\Gamma_{ij}\times(0,T)}\right), \tag{3.3.16}$$

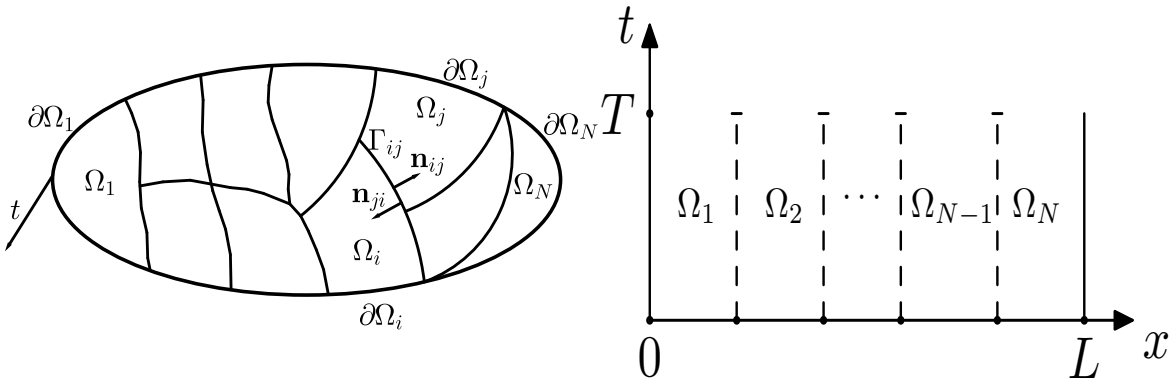


Figure 3.3.3: Splitting into many non-overlapping subdomains

where $\theta \in (0, 1]$ is a relaxation parameter.

3.3.2.1 Convergence analysis

As in the case of the DNWR algorithm in the previous chapter, we prove our results for the one dimensional heat equation on the domain $\Omega := (0, L)$ with boundary conditions $u(0, t) = g_0(t)$ and $u(L, t) = g_L(t)$. We decompose Ω into non-overlapping subdomains $\Omega_i := (x_{i-1}, x_i)$, $i = 1, \ldots, N$, and define the subdomain length $h_i := x_i - x_{i-1}$, and $h_{\min} := \min_{1\le i\le N} h_i$. Our initial guess is denoted by $\{w_i^0(t)\}_{i=1}^{N-1}$ on the interfaces x_i. By linearity, we again study the error equations, $f = 0$, $g_0 = g_L = 0$ and $u_0 = 0$, which leads with $w_0^k(t) = w_N^k(t) = 0$ for all k to

$$\begin{aligned}
\partial_t u_i^k - \nu\partial_{xx} u_i^k &= 0, \quad \text{in } \Omega_i, & \partial_t \psi_i^k - \nu\partial_{xx}\psi_i^k &= 0, \quad \text{in } \Omega_i,\\
u_i^k(x,0) &= 0, \quad \text{in } \Omega_i, & \psi_i^k(x,0) &= 0, \quad \text{in } \Omega_i,\\
u_i^k(x_{i-1},t) &= w_{i-1}^{k-1}(t), & -\partial_x\psi_i^k(x_{i-1},t) &= (\partial_x u_{i-1}^k - \partial_x u_i^k)(x_{i-1},t),\\
u_i^k(x_i,t) &= w_i^{k-1}(t), & \partial_x\psi_i^k(x_i,t) &= (\partial_x u_i^k - \partial_x u_{i+1}^k)(x_i,t),
\end{aligned} \tag{3.3.17}$$

except for the first and last subdomains, where in the Neumann step the Neumann conditions are replaced by homogeneous Dirichlet conditions at the physical boundaries. The new interface values for the next step are then defined as

$$w_i^k(t) = w_i^{k-1}(t) - \theta\left(\psi_i^k(x_i,t) + \psi_{i+1}^k(x_i,t)\right). \tag{3.3.18}$$

We have the following convergence theorem, that we prove using the auxiliary results on kernel estimates from Subsection 2.3.1.2 of the previous chapter.

Theorem 25 (Convergence of NNWR for multiple subdomains)**.** *For $\theta = 1/4$ and $T > 0$ fixed, the NNWR algorithm (3.3.17)-(3.3.18) converges superlinearly with the estimate*

$$\max_{1\le i\le N-1} \|w_i^k(\cdot)\|_{L^\infty(0,T)} \le \left(\frac{\sqrt{6}}{1 - e^{-\frac{(2k+1)h_{\min}^2}{\nu T}}}\right)^{2k} e^{-k^2 h_{\min}^2/\nu T} \max_{1\le i\le N-1} \|w_i^0(\cdot)\|_{L^\infty(0,T)}. \tag{3.3.19}$$

Proof. We start by applying the Laplace transform to the homogeneous Dirichlet subproblems in (3.3.17), and obtain

$$s\hat{u}_i - \nu\hat{u}_{i,xx} = 0, \quad \hat{u}_i(x_{i-1}, s) = \hat{w}_{i-1}(s), \quad \hat{u}_i(x_i, s) = \hat{w}_i(s),$$

for $i = 2, ..., N-1$. These subdomain problems have the solutions

$$\hat{u}_i(x,s) = \frac{1}{\sinh(h_i\sqrt{s/\nu})}\left(\hat{w}_i(s)\sinh\left((x-x_{i-1})\sqrt{s/\nu}\right) + \hat{w}_{i-1}(s)\sinh\left((x_i-x)\sqrt{s/\nu}\right)\right).$$

Next we apply the Laplace transform to the Neumann subproblems (3.3.17) for subdomains not touching the physical boundary, and obtain

$$\hat{\psi}_i(x,s) = C_i(s)\cosh\left((x-x_{i-1})\sqrt{s/\nu}\right) + D_i(s)\cosh\left((x_i-x)\sqrt{s/\nu}\right),$$

where the notation $\sigma_i := \sinh\left(h_i\sqrt{s/\nu}\right)$ and $\gamma_i := \cosh\left(h_i\sqrt{s/\nu}\right)$ gives

$$\begin{aligned} C_i &= \frac{1}{\sigma_i}\left(\hat{w}_i\left(\frac{\gamma_i}{\sigma_i} + \frac{\gamma_{i+1}}{\sigma_{i+1}}\right) - \frac{\hat{w}_{i-1}}{\sigma_i} - \frac{\hat{w}_{i+1}}{\sigma_{i+1}}\right), \\ D_i &= \frac{1}{\sigma_i}\left(\hat{w}_{i-1}\left(\frac{\gamma_i}{\sigma_i} + \frac{\gamma_{i-1}}{\sigma_{i-1}}\right) - \frac{\hat{w}_{i-2}}{\sigma_{i-1}} - \frac{\hat{w}_i}{\sigma_i}\right). \end{aligned}$$

We therefore obtain for $i = 2, ..., N-2$, at iteration k

$$\begin{aligned} \hat{w}_i^k(s) &= \hat{w}_i^{k-1}(s) - \theta\left(\hat{\psi}_i^k(x_i,s) + \hat{\psi}_{i+1}^k(x_i,s)\right) \\ &= \hat{w}_i^{k-1}(s) - \theta\left(C_i\gamma_i + D_i + C_{i+1} + D_{i+1}\gamma_{i+1}\right). \end{aligned}$$

Using the identity $\gamma_i^2 - 1 = \sigma_i^2$ and simplifying, we get

$$\begin{aligned} \hat{w}_i^k = \hat{w}_i^{k-1} - \theta\Bigg(&\hat{w}_i^{k-1}\left(2 + \frac{2\gamma_i\gamma_{i+1}}{\sigma_i\sigma_{i+1}}\right) + \frac{\hat{w}_{i+1}^{k-1}}{\sigma_{i+1}}\left(\frac{\gamma_{i+2}}{\sigma_{i+2}} - \frac{\gamma_i}{\sigma_i}\right) \\ &+ \frac{\hat{w}_{i-1}^{k-1}}{\sigma_i}\left(\frac{\gamma_{i-1}}{\sigma_{i-1}} - \frac{\gamma_{i+1}}{\sigma_{i+1}}\right) - \frac{\hat{w}_{i+2}^{k-1}}{\sigma_{i+1}\sigma_{i+2}} - \frac{\hat{w}_{i-2}^{k-1}}{\sigma_i\sigma_{i-1}}\Bigg). \end{aligned} \tag{3.3.20}$$

For $i = 1$ and $i = N$, the Neumann conditions on the physical boundary are replaced by homogeneous Dirichlet conditions $\psi_1(0,t) = 0$ and $\psi_N(L,t) = 0$, $t > 0$. For these two subdomains, we obtain as solution after a Laplace transform

$$\begin{aligned} \hat{\psi}_1(x,s) &= \frac{1}{\gamma_1}\left(\hat{w}_1\left(\frac{\gamma_1}{\sigma_1} + \frac{\gamma_2}{\sigma_2}\right) - \frac{\hat{w}_2}{\sigma_2}\right)\sinh\left((x-x_0)\sqrt{s/\nu}\right), \\ \hat{\psi}_N(x,s) &= \frac{1}{\gamma_N}\left(\hat{w}_{N-1}\left(\frac{\gamma_{N-1}}{\sigma_{N-1}} + \frac{\gamma_N}{\sigma_N}\right) - \frac{\hat{w}_{N-2}}{\sigma_{N-1}}\right)\sinh\left((x_N-x)\sqrt{s/\nu}\right), \end{aligned}$$

and thus the recurrence relations on the first interface is

$$\hat{w}_1^k = \hat{w}_1^{k-1} - \theta\left(\hat{w}_1^{k-1}\left(2 + \frac{\gamma_1\gamma_2}{\sigma_1\sigma_2} + \frac{\sigma_1\gamma_2}{\gamma_1\sigma_2}\right) + \frac{\hat{w}_2^{k-1}}{\sigma_2}\left(\frac{\gamma_3}{\sigma_3} - \frac{\sigma_1}{\gamma_1}\right) - \frac{\hat{w}_3^{k-1}}{\sigma_2\sigma_3}\right), \tag{3.3.21}$$

and on the last interface, we obtain

$$\hat{w}_{N-1}^{k} = \hat{w}_{N-1}^{k-1} - \theta\left(\hat{w}_{N-1}^{k-1}\left(2 + \frac{\gamma_{N-1}\gamma_N}{\sigma_{N-1}\sigma_N} + \frac{\sigma_N\gamma_{N-1}}{\gamma_N\sigma_{N-1}}\right) + \frac{\hat{w}_{N-2}^{k-1}}{\sigma_{N-1}}\left(\frac{\gamma_{N-2}}{\sigma_{N-2}} - \frac{\sigma_N}{\gamma_N}\right) - \frac{\hat{w}_{N-3}^{k-1}}{\sigma_{N-1}\sigma_{N-2}}\right). \quad (3.3.22)$$

Defining $\sigma := \sinh(h_{\min}\sqrt{s/\nu})$ where $h_{\min} = \min_{1\le i\le N} h_i$, and setting

$$\hat{\nu}_i^k(s) := \sigma^{2k}\hat{w}_i^k(s), \quad (3.3.23)$$

relation (3.3.20) reduces for the special choice $\theta = 1/4$ to

$$\hat{\nu}_i^k(s) = -\frac{1}{4}\left(\hat{t}_{i,i}\hat{\nu}_i^{k-1}(s) + \hat{t}_{i,i+1}\hat{\nu}_{i+1}^{k-1}(s) + \hat{t}_{i,i-1}\hat{\nu}_{i-1}^{k-1}(s) - \hat{t}_{i,i+2}\hat{\nu}_{i+2}^{k-1}(s) - \hat{t}_{i,i-2}\hat{\nu}_{i-2}^{k-1}(s)\right), \quad (3.3.24)$$

where we defined $\hat{t}_{i,i} := \frac{2\sigma^2}{\sigma_i\sigma_{i+1}}(\gamma_i\gamma_{i+1} - \sigma_i\sigma_{i+1})$, $\hat{t}_{i,i+1} := \frac{\sigma^2}{\sigma_i\sigma_{i+1}\sigma_{i+2}}(\sigma_i\gamma_{i+2} - \gamma_i\sigma_{i+2})$, $\hat{t}_{i,i-1} := \frac{\sigma^2}{\sigma_i\sigma_{i-1}\sigma_{i+1}}(\sigma_{i+1}\gamma_{i-1} - \gamma_{i+1}\sigma_{i-1})$, $\hat{t}_{i,i+2} := \frac{\sigma^2}{\sigma_{i+1}\sigma_{i+2}}$, and $\hat{t}_{i,i-2} := \frac{\sigma^2}{\sigma_i\sigma_{i-1}}$. Similarly, we obtain for (3.3.21)

$$\hat{\nu}_1^k(s) = -\frac{1}{4}\left(\hat{t}_{1,1}\hat{\nu}_1^{k-1}(s) + \hat{t}_{1,2}\hat{\nu}_2^{k-1}(s) - \hat{t}_{1,3}\hat{\nu}_3^{k-1}(s)\right), \quad (3.3.25)$$

where we defined $\hat{t}_{1,1} := \sigma^2\left(\frac{\sigma_1\gamma_2}{\gamma_1\sigma_2} + \frac{\gamma_1\gamma_2}{\sigma_1\sigma_2} - 2\right)$, $\hat{t}_{1,2} = \frac{\sigma^2}{\sigma_2}\left(\frac{\gamma_3}{\sigma_3} - \frac{\sigma_1}{\gamma_1}\right)$ and $\hat{t}_{1,3} = \frac{\sigma^2}{\sigma_2\sigma_3}$. From (3.3.22), we obtain

$$\hat{\nu}_{N-1}^k(s) = -\frac{1}{4}\left(\hat{t}_{N-1,N-1}\hat{\nu}_{N-1}^{k-1}(s) + \hat{t}_{N-1,N-2}\hat{\nu}_{N-2}^{k-1}(s) - \hat{t}_{N-1,N-3}\hat{\nu}_{N-3}^{k-1}(s)\right), \quad (3.3.26)$$

where we defined

$$\hat{t}_{N-1,N-1} := \sigma^2\left(\frac{\sigma_{N-1}\gamma_{N-2}}{\gamma_{N-1}\sigma_{N-2}} + \frac{\gamma_{N-1}\gamma_{N-2}}{\sigma_{N-1}\sigma_{N-2}} - 2\right), \hat{t}_{N-1,N-2} = \frac{\sigma^2}{\sigma_{N-2}}\left(\frac{\gamma_{N-3}}{\sigma_{N-3}} - \frac{\sigma_{N-1}}{\gamma_{N-1}}\right)$$

and $\hat{t}_{N-1,N-3} = \frac{\sigma^2}{\sigma_{N-2}\sigma_{N-3}}$. Now we will show that for all i

$$\|\nu_i^k(\cdot)\|_{L^\infty(0,T)} \le \frac{3}{2}\max_{1\le j\le N-1}\|\nu_j^{k-1}(\cdot)\|_{L^\infty(0,T)}, \quad k = 1, 2, 3, \ldots. \quad (3.3.27)$$

At first glance it seems that inequality (3.3.27) is not enough to prove convergence, because the factor 3/2 is bigger than one, but the ν_i^k are related to the w_i^k by σ, see (3.3.23), and the superlinear convergence of w_i^k will come from σ. To obtain (3.3.27), we have to bound $\int_0^\infty |t_{i,j}(t)|dt$, where $t_{i,j} = \mathcal{L}^{-1}\left\{\hat{t}_{i,j}\right\}$. We first consider $\hat{t}_{i,i+2}$. If $h_{i+1} = h_{i+2} = h_{\min}$, then the terms in $\hat{t}_{i,i+2}$ cancel and we simply get $\hat{t}_{i,i+2} = 1$. If $h_{\min} \le h_{i+1}$ and $h_{\min} \le h_{i+2}$, then the kernel $t_{i,i+2}$, being a convolution of two positive functions, is positive by part 1 of Lemma 7, and using Lemma 8 its integral is bounded by

$$\int_0^\infty |t_{i,i+2}(t)|dt \le \lim_{s\to 0+}\frac{\sinh^2(h_{\min}\sqrt{s/\nu})}{\sinh(h_{i+1}\sqrt{s/\nu})\sinh(h_{i+2}\sqrt{s/\nu})} \le \frac{h_{\min}^2}{h_{i+1}h_{i+2}} \le 1,$$

so that

$$\|\mathcal{L}^{-1}\left(\hat{t}_{i,i+2}\hat{\nu}_{i+2}^{k}(s)\right)\|_{L^\infty(0,T)} \leq \|\nu_{i+2}^{k}\|_{L^\infty(0,T)}.$$

The same argument also holds for the term involving $\hat{\nu}_{i-2}^{k-1}$. Now for $\hat{\nu}_{i+1}^{k-1}$, we rewrite $\hat{t}_{i,i+1}$ as

$$\hat{t}_{i,i+1} = \frac{\sinh\left((h_i - h_{i+2})\sqrt{s/\nu}\right)\sinh^2(h_{\min}\sqrt{s/\nu})}{\sinh(h_i\sqrt{s/\nu})\sinh(h_{i+1}\sqrt{s/\nu})\sinh(h_{i+2}\sqrt{s/\nu})}.$$

Assuming that $h_i \geq h_{i+2}$, we use Lemma 9 and Lemma 8 to get a bound of the form

$$\int_0^\infty |t_{i,i+1}(t)|dt \leq \frac{|h_i - h_{i+2}|h_{\min}^2}{h_i h_{i+1} h_{i+2}} < 1,$$

and similarly for the term involving $\hat{\nu}_{i-1}^{k-1}$. Finally, for the term $\hat{t}_{i,i}$, we use the trigonometric identity $\sinh(A)\cosh(B) = \frac{1}{2}\left(\sinh(A+B) + \sinh(A-B)\right)$ to obtain

$$\hat{t}_{i,i} = \frac{\sinh(h_{\min}\sqrt{s/\nu})\sinh\left((h_{\min} + h_i - h_{i+1})\sqrt{s/\nu}\right)}{\sinh(h_i\sqrt{s/\nu})\sinh(h_{i+1}\sqrt{s/\nu})} + \frac{\sinh(h_{\min}\sqrt{s/\nu})\sinh\left((h_{\min} + h_{i+1} - h_i)\sqrt{s/\nu}\right)}{\sinh(h_i\sqrt{s/\nu})\sinh(h_{i+1}\sqrt{s/\nu})}.$$

Each term is again a ratio of hyperbolic sines, so we only need to pair the factors so that the coefficient in the numerator is always smaller than the one in the denominator. Now $-h_{i+1} \leq h_{\min} + h_i - h_{i+1} = h_i + h_{\min} - h_{i+1} \leq h_i$, so for the first term, we choose the pairing

$$\frac{\sinh(h_{\min}\sqrt{s/\nu})}{\sinh(h_{i+1}\sqrt{s/\nu})} \cdot \frac{\sinh\left((h_{\min} + h_i - h_{i+1})\sqrt{s/\nu}\right)}{\sinh(h_i\sqrt{s/\nu})} \quad \text{if } h_{i+1} \leq h_i,$$

and

$$\frac{\sinh(h_{\min}\sqrt{s/\nu})}{\sinh(h_i\sqrt{s/\nu})} \cdot \frac{\sinh\left((h_{\min} + h_i - h_{i+1})\sqrt{s/\nu}\right)}{\sinh(h_{i+1}\sqrt{s/\nu})} \quad \text{if } h_{i+1} \geq h_i.$$

A similar argument holds also for the second term. Now using Lemma 8, we have again integrals of kernels bounded by 1. In summary, we get for $2 \leq i \leq N-2$

$$\|\nu_i^k(\cdot)\|_{L^\infty(0,T)} \leq \frac{1}{2}\|\nu_i^{k-1}(\cdot)\|_{L^\infty(0,T)} + \frac{1}{4}\left(\|\nu_{i-2}^{k-1}(\cdot)\|_{L^\infty(0,T)} + \|\nu_{i-1}^{k-1}(\cdot)\|_{L^\infty(0,T)} + \|\nu_{i+1}^{k-1}(\cdot)\|_{L^\infty(0,T)} + \|\nu_{i+2}^{k-1}(\cdot)\|_{L^\infty(0,T)}\right),$$

and the estimate (3.3.27) is established for interior subdomains. For the left subdomain touching the boundary, the kernel $t_{1,3}$ in (3.3.25) can be estimated like $t_{i,i+2}$. For $\hat{t}_{1,2}$, we have

$$\hat{t}_{1,2} = \frac{\cosh\left((h_1 - h_3)\sqrt{s/\nu}\right)\sinh^2(h_{\min}\sqrt{s/\nu})}{\cosh(h_1\sqrt{s/\nu})\sinh(h_2\sqrt{s/\nu})\sinh(h_3\sqrt{s/\nu})}.$$

If $h_1 \geq h_3$, the decomposition $\hat{t}_{1,2} = \frac{\cosh\left((h_1-h_3)\sqrt{s/\nu}\right)}{\cosh(h_1\sqrt{s/\nu})} \cdot \frac{\sinh(h_{\min}\sqrt{s/\nu})}{\sinh(h_2\sqrt{s/\nu})} \cdot \frac{\sinh(h_{\min}\sqrt{s/\nu})}{\sinh(h_3\sqrt{s/\nu})}$ shows that one can bound

$$\int_0^\infty |t_{1,2}(t)|dt \leq h_{\min}^2/h_2h_3 < 1.$$

If $h_3 > h_1$, then we rewrite

$$\hat{t}_{1,2} = \frac{1}{\cosh(h_1\sqrt{s/\nu})} \cdot \frac{\sinh(h_{\min}\sqrt{s/\nu})}{\sinh(h_2\sqrt{s/\nu})} \left(\frac{\sinh\left((h_{\min}+h_1-h_3)\sqrt{s/\nu}\right)}{2\sinh(h_3\sqrt{s/\nu})} + \frac{\sinh\left((h_{\min}+h_3-h_1)\sqrt{s/\nu}\right)}{2\sinh(h_3\sqrt{s/\nu})} \right),$$

which again shows using Lemma 8 that the integral is bounded by 1. Finally we consider

$$\begin{aligned} \hat{t}_{1,1} &= \frac{2\cosh\left((2h_1-h_2)\sqrt{s/\nu}\right)\sinh^2(h_{\min}\sqrt{s/\nu})}{\sinh(2h_1\sqrt{s/\nu})\sinh(h_2\sqrt{s/\nu})} \\ &= \frac{\sinh(h_{\min}\sqrt{s/\nu})\sinh\left((h_{\min}+2h_1-h_2)\sqrt{s/\nu}\right)}{\sinh(2h_1\sqrt{s/\nu})\sinh(h_2\sqrt{s/\nu})} \\ &\quad + \frac{\sinh(h_{\min}\sqrt{s/\nu})\sinh\left((h_{\min}-2h_1+h_2)\sqrt{s/\nu}\right)}{\sinh(2h_1\sqrt{s/\nu})\sinh(h_2\sqrt{s/\nu})}. \end{aligned}$$

Using the inequalities $-h_2 \leq h_{\min}+2h_1-h_2 \leq 2h_1$ and $-2h_1 \leq h_{\min}-2h_1+h_2 \leq h_2$ we can again, with an appropriate pairing of factors, bound the integral of each term by 1. Thus we have for the subdomain touching the left physical boundary

$$\|\nu_1^k(\cdot)\|_{L^\infty(0,T)} \leq \frac{1}{2}\|\nu_1^{k-1}(\cdot)\|_{L^\infty(0,T)} + \frac{1}{4}\left(\|\nu_2^{k-1}(\cdot)\|_{L^\infty(0,T)} + \|\nu_3^{k-1}(\cdot)\|_{L^\infty(0,T)}\right).$$

A similar result holds for $\hat{\nu}_{N-1}^k(s)$, and hence the inequality (3.3.27) holds for all $1 \leq i \leq N-1$. Therefore, by induction, we obtain

$$\max_{1\leq j\leq N-1}\|\nu_j^k(\cdot)\|_{L^\infty(0,T)} \leq \left(\frac{3}{2}\right)^k \max_{1\leq j\leq N-1}\|\nu_j^0(\cdot)\|_{L^\infty(0,T)} = \left(\frac{3}{2}\right)^k \max_{1\leq j\leq N-1}\|w_j^0(\cdot)\|_{L^\infty(0,T)}.$$

Now since

$$w_i^k(t) = \left(\phi_{2k} * \nu_i^k\right)(t) = \int_0^t \phi_{2k}(t-\tau)\nu_i^k(\tau)d\tau,$$

with $\phi_{2k} = \mathcal{L}^{-1}\left(\frac{1}{\sinh^{2k}(h_{\min}\sqrt{s/\nu})}\right)$, a similar estimate using part 3 of Lemma 7 as in the proof of Theorem 12 leads to the superlinear convergence estimate (3.3.19). □

Table 3.1: Subdomain widths used for the NNWR experiments in Fig. 3.3.4 & Fig. 3.3.5.

No. of subdomains	h_1	h_2	h_3	h_4	h_5	h_6
2	3.50	2.50				
3	2.30	2.30	1.40			
4	1.20	2.40	1.80	0.60		
5	1.80	1.40	1.08	1.00	0.72	
6	1.20	0.80	1.00	1.20	1.00	0.80

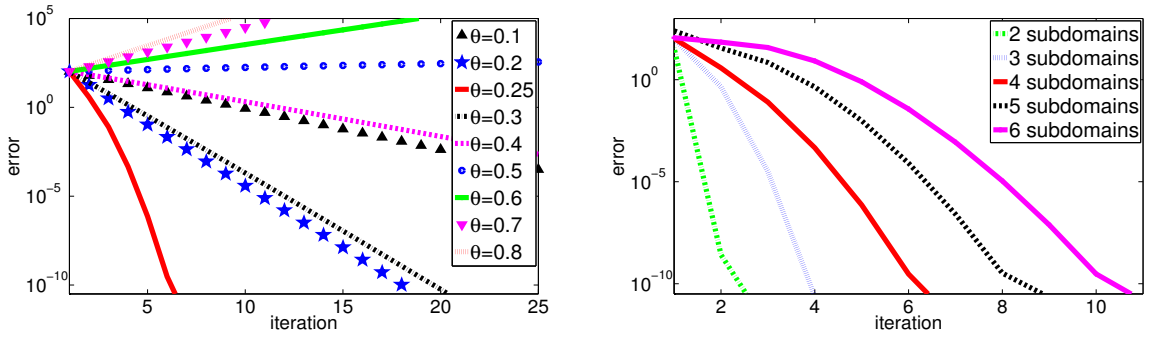


Figure 3.3.4: Convergence of NNWR with four subdomains and various relaxation parameters on the left, and dependence of NNWR on the number of subdomains for $\theta = 1/4$ on the right

3.3.2.2 Numerical illustration

We show experiments for the NNWR algorithm in the spatial domain $\Omega = (0, 6)$, with the same model problem (3.3.13) and the discretization parameters Δx and Δt as in Subsection 3.2.1.3, and for the time window $T = 2$. From now onward we always use $\kappa(x) = 1$, unless otherwise specified. In Figure 3.3.4, we consider a decomposition into two to six unequal subdomains, whose widths are shown in Table 3.1. On the left panel, we show the convergence in the four-subdomain case as a function of the relaxation parameter θ, whereas on the right panel, we show the convergence for $\theta = 1/4$ as we vary the number of subdomains. We observe superlinear convergence for $\theta = 1/4$, and only linear convergence for the other choices. We also see that convergence slows down as the number of subdomains is increased, as expected.

We show in Figure 3.3.5 a comparison of the numerically measured convergence for the NNWR algorithm for $\theta = 1/4$ and $\kappa(x) = 1$, and the theoretical estimates from Theorem 25. On the left, we show the results for many subdomains of equal length, and on the right, we show the results for the case of many subdomains of unequal length (subdomain lengths are as in Table 3.1).

3.3.3 NNWR in 2D

In this section we formulate and analyze the NNWR algorithm, applied to the two-dimensional heat equation

$$\partial_t u - \nu \Delta u = f(x, y, t), \quad (x, y) \in \Omega = (l, L) \times (0, \pi),\ t \in (0, T]$$

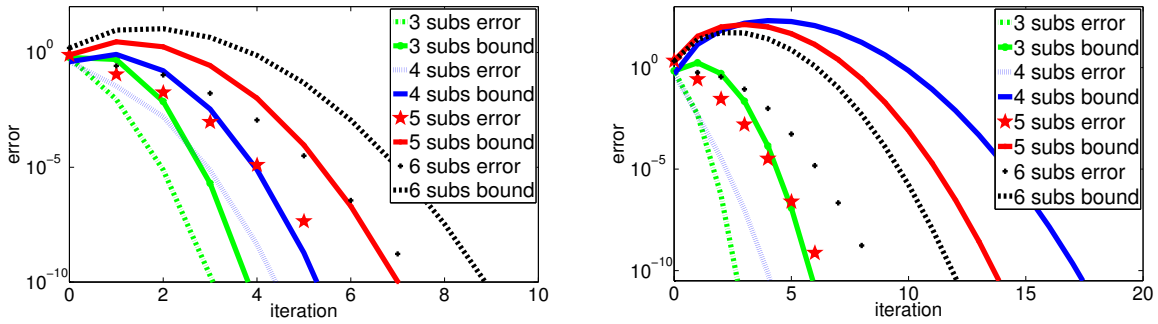


Figure 3.3.5: Comparison of the numerically measured convergence rates and the theoretical error estimates for NNWR for $\kappa(x) = 1$ with $\theta = 1/4$ and $T = 2$, on the left for many equal subdomains, and on the right for many unequal subdomains

with initial condition $u(x, y, 0) = u_0(x, y)$ and Dirichlet boundary conditions. To define the Neumann-Neumann algorithm, we decompose Ω into strips of the form $\Omega_i = (x_{i-1}, x_i) \times (0, \pi)$, $l = x_0 < x_1 < \cdots < x_N = L$. The Neumann-Neumann algorithm, considering directly the error equations with $f(x, y, t) = 0, u_0(x, y) = 0$ and homogeneous Dirichlet boundary conditions, is then given by performing iteratively for $k = 1, 2, \ldots$ and for $i = 1, \ldots, N$ the Dirichlet and Neumann steps

$$\begin{aligned}
\partial_t u_i^k - \nu\Delta u_i^k &= 0, \qquad \text{in } \Omega_i, & \partial_t \psi_i^k - \nu\Delta \psi_i^k &= 0, \qquad \text{in } \Omega_i,\\
u_i^k(x, y, 0) &= 0, & \psi_i^k(x, y, 0) &= 0,\\
u_i^k(x_{i-1}, y, t) &= g_{i-1}^{k-1}(y, t), & \partial_{n_i}\psi_i^k(x_{i-1}, y, t) &= (\partial_{n_{i-1}} u_{i-1}^k + \partial_{n_i} u_i^k)(x_{i-1}, y, t),\\
u_i^k(x_i, y, t) &= g_i^{k-1}(y, t), & \partial_{n_i}\psi_i^k(x_i, y, t) &= (\partial_{n_{i-1}} u_{i-1}^k + \partial_{n_i} u_i^k)(x_i, y, t),\\
u_i^k(x, 0, t) &= u_i^k(x, \pi, t) = 0, & \psi_i^k(x, 0, t) &= \psi_i^k(x, \pi, t) = 0,
\end{aligned} \tag{3.3.28}$$

except for the first and last subdomain, where in the Neumann step the Neumann conditions are replaced by homogeneous Dirichlet conditions along the physical boundaries, as in the one dimensional case. The new interface values for the next step are then defined as

$$g_i^k(y, t) = g_i^{k-1}(y, t) - \theta\left(\psi_i^k(x_i, y, t) + \psi_{i+1}^k(x_i, y, t)\right).$$

3.3.3.1 Convergence analysis

To analyze the NNWR algorithm (3.3.28) in two dimensions, we first reduce the problem to a collection of one-dimensional problems by performing a Fourier transform along the y direction. More precisely, we use a Fourier sine series along the y-direction,

$$u_i^k(x, y, t) = \sum_{n\geq 1} U_i^k(x, n, t)\sin(ny),$$

where

$$U_i^k(x, n, t) = \frac{2}{\pi}\int_0^\pi u_i^k(x, \eta, t)\sin(n\eta)d\eta.$$

Thus, the 2D problems in the NNWR algorithm (3.3.28) become a sequence of 1D problems indexed by n,

$$\frac{\partial U_i^k}{\partial t}(x,n,t) - \nu\frac{\partial^2 U_i^k}{\partial x^2}(x,n,t) + \nu n^2 U_i^k(x,n,t) = 0, \tag{3.3.29}$$

and the boundary conditions for $U_i^k(x,n,t)$ are identical to the one-dimensional case for each n.

Theorem 26 (Convergence of NNWR in 2D)**.** *Let $\theta = 1/4$. For $T > 0$ fixed, the NNWR algorithm (3.3.28) converges superlinearly with the estimate*

$$\max_{1\le i\le N-1}\|g_i^k\|_{L^\infty(0,T;L^2(0,\pi))} \le \left(\frac{\sqrt{6}}{1-e^{-\frac{(2k+1)h_{\min}^2}{\nu T}}}\right)^{2k} e^{-k^2h_{\min}^2/\nu T}\max_{1\le i\le N-1}\|g_i^0\|_{L^\infty(0,T;L^2(0,\pi))},$$

where $h_{\min}$ is the minimum subdomain width.

Proof. We take Laplace transforms in t of (3.3.29) to get

$$(s+\nu n^2)\hat{U}_i^k - \nu\frac{d^2\hat{U}_i^k}{dx^2} = 0,$$

and now treat each n as in the one-dimensional analysis in the proof of Theorem 25, where the recurrence relations (3.3.20), (3.3.21) and (3.3.22) of the form

$$\hat{w}_i^k(s) = \sum_j A_{ij}^{(k)}(s)\hat{w}_j^0(s)$$

now become for each $n = 1, 2, \ldots$

$$\hat{G}_i^k(n,s) = \sum_j A_{ij}^{(k)}(s+\nu n^2)\hat{G}_j^0(n,s). \tag{3.3.30}$$

If $a_{ij}^{(k)}(t)$ is the inverse Laplace transform of $A_{ij}^{(k)}(s)$, i.e.,

$$A_{ij}^{(k)}(s) = \int_0^\infty a_{ij}^{(k)}(t)e^{-st}dt, \tag{3.3.31}$$

then if we replace s by $s+\nu n^2$ in (3.3.31), we get $A_{ij}^{(k)}(s+\nu n^2) = \int_0^\infty a_{ij}^{(k)}(t)e^{-\nu n^2 t}e^{-st}dt$, so the inverse Laplace transform of $A_{ij}^{(k)}(s+\nu n^2)$ is just $a_{ij}^{(k)}(t)e^{-\nu n^2 t}$. Hence taking the inverse Laplace transform of (3.3.30), we get

$$G_i^k(n,t) = \sum_j\int_0^t a_{ij}^{(k)}(\tau)e^{-\nu n^2\tau}G_j^0(n,t-\tau)d\tau.$$

So the interface functions $g_i^k(y,t)$ can be written as

$$\begin{aligned} g_i^k(y,t) &= \sum_{n\ge1} G_i^k(n,t)\sin(ny) \\ &= \sum_{n\ge1}\sum_j\int_0^t a_{ij}^{(k)}(\tau)e^{-\nu n^2\tau}\left(\frac{2}{\pi}\int_0^\pi g_j^0(\eta,t-\tau)\sin(n\eta)d\eta\right)\sin(ny)d\tau. \end{aligned}$$

Next, we justify the exchange of the infinite sum and the integrals using Fubini-Tonelli's theorem*. Here, we need to check that $\left|\sum_{n=1}^{\infty} a_{ij}^{(k)}(\tau)e^{-\nu n^2\tau}\right|$ remains bounded for all $\tau \geq 0$. For τ bounded away from zero, this follows from the boundedness of $|a_{ij}^{(k)}|$ and from the geometric series, so it suffices to show boundedness for τ close to zero. To do so, note that $A_{ij}^{(k)}(s)$ contains $1/\sinh^{2k}(h_{\min}\sqrt{s})$ as a factor, which implies $\lim_{s\to\infty} s^p A_{ij}^{k}(s) = 0$ for all $p > 1$. This means $a_{ij}^{(k)}(\tau)$ is infinitely differentiable at $\tau = 0$ and its derivatives of all orders vanish there. Thus, by Taylor's theorem, there exists a constant C such that $|a_{ij}^{(k)}(\tau)| \leq C\tau^2$ for $\tau > 0$ small enough, so we have

$$\left|\sum_{n=1}^{M} a_{ij}^{(k)}(\tau)e^{-\nu n^2\tau}\right| \leq \frac{C\tau^2}{1-e^{-\nu\tau}}. \tag{3.3.32}$$

In particular, for $0 < \nu\tau < 1$, we have $1 - e^{-\nu\tau} \geq \nu\tau - \frac{\nu^2\tau^2}{2} \geq \frac{\nu\tau}{2}$, so the sum (3.3.32) is bounded above by $2C\tau/\nu$, which is independent of M. Therefore, $\left|\sum_{n=1}^{\infty} a_{ij}^{(k)}(\tau)e^{-\nu n^2\tau}\right|$ is bounded uniformly for all $\tau \in (0,\infty)$, so we can apply Fubini-Tonelli's theorem to interchange sums and integrals and get

$$g_i^k(y,t) = \sum_j \int_0^t a_{ij}^{(k)}(\tau) \int_0^\pi g_j^0(\eta, t-\tau)\left(\frac{2}{\pi}\sum_{n\geq 1} e^{-\nu n^2\tau}\sin(n\eta)\sin(ny)\right) d\eta d\tau. \tag{3.3.33}$$

We now use the trigonometric identity $\sin(A)\sin(B) = \frac{1}{2}\cos(A-B) - \frac{1}{2}\cos(A+B)$ to rewrite (3.3.33) as

$$\begin{aligned}\frac{2}{\pi}\sum_{n\geq 1} e^{-\nu n^2\tau}\sin(n\eta)\sin(ny) &= \frac{1}{\pi}\sum_{n\geq 1} e^{-\nu n^2\tau}\left(\cos\left(n(\eta-y)\right) - \cos\left(n(\eta+y)\right)\right)\\ &= \frac{1}{2\pi}\sum_{n\in\mathbb{Z}} e^{-\nu n^2\tau}\left(\exp\left(in(\eta-y)\right) - \exp\left(in(\eta+y)\right)\right).\end{aligned} \tag{3.3.34}$$

Now we recall the following well-known properties of the Fourier transform*: for a Fourier transform pair (f, F),

P1. if both f and F are continuous and decay sufficiently rapidly, then $\sum_{n\in\mathbb{Z}} f(n) = \sum_{k\in\mathbb{Z}} F(2k\pi)$ (*Poisson summation formula*),

P2. $\mathcal{F}\left\{f(x)e^{iw_0x}\right\}(w) = F(w-w_0)$ (*Shifting property*),

P3. $\mathcal{F}\left\{e^{-x^2\tau}\right\}(w) = \sqrt{\frac{\pi}{\tau}}e^{-w^2/4\tau}$.

Thus, using the properties P2 and P3 and the Poisson summation formula (P1), we obtain from (3.3.34)

$$\frac{2}{\pi}\sum_{n\geq 1} e^{-\nu n^2\tau}\sin(n\eta)\sin(ny) = \frac{1}{\sqrt{4\pi\nu\tau}}\sum_{k\in\mathbb{Z}}\left(e^{-(2k\pi-\eta+y)^2/4\tau\nu} - e^{-(2k\pi-\eta-y)^2/4\tau\nu}\right).$$

*for a formal discussion on Fubini's theorems and Fourier transform, see Appendix A

Interchanging the sum and the integral, (3.3.33) gives

$$g_i^k(y,t) = \sum_j \int_0^t \frac{a_{ij}^{(k)}(\tau)}{\sqrt{4\pi\nu\tau}} \left(\sum_{k\in\mathbb{Z}} \int_0^\pi g_j^0(\eta, t-\tau) \left(e^{-\frac{(y-(\eta-2k\pi))^2}{4\tau\nu}} - e^{-\frac{(y+(\eta-2k\pi))^2}{4\tau\nu}} \right) d\eta \right) d\tau. \tag{3.3.35}$$

Now splitting the two integrals and performing the change of variables $\zeta = \eta - 2k\pi$ in the first integral and $\zeta = 2k\pi - \eta$ in the second, (3.3.35) gives

$$\sum_{k\in\mathbb{Z}} \int_{-2k\pi}^{(1-2k)\pi} g_j^0(\zeta+2k\pi, t-\tau) e^{-(y-\zeta)^2/4\tau\nu} d\zeta - \sum_{k\in\mathbb{Z}} \int_{(2k-1)\pi}^{2k\pi} g_j^0(2k\pi-\zeta, t-\tau) e^{-(y-\zeta)^2/4\tau\nu} d\zeta.$$

Letting $m = -k$ in the first integral, we obtain

$$\sum_{m\in\mathbb{Z}} \int_{2m\pi}^{(2m+1)\pi} g_j^0(\zeta-2m\pi, t-\tau) e^{-(y-\zeta)^2/4\tau\nu} d\zeta - \sum_{m\in\mathbb{Z}} \int_{(2m-1)\pi}^{2m\pi} g_j^0(2m\pi-\zeta, t-\tau) e^{-(y-\zeta)^2/4\tau\nu} d\zeta.$$

Defining the 2π-periodic odd extension of g_j^0 as

$$\bar{g}_j^0(y,t) = \begin{cases} g_j^0(y-2m\pi, t), & 2m\pi < y < (2m+1)\pi, \\ -g_j^0(2m\pi - y, t), & (2m-1)\pi < y < 2m\pi \quad (m \in \mathbb{Z}), \end{cases}$$

we can rewrite (3.3.35) as

$$g_i^k(y,t) = \sum_j \int_0^t \int_{-\infty}^{\infty} \frac{a_{ij}^{(k)}(\tau)}{\sqrt{4\pi\nu\tau}} \bar{g}_j^0(\zeta, t-\tau) e^{-(y-\zeta)^2/4\tau\nu} d\zeta d\tau. \tag{3.3.36}$$

Now since $\bar{g}_j^0$ and g_j^0 have the same maxima and minima, we have

$$|g_i^k(y,t)| \le \sum_j \|g_j^0\| \int_0^t \int_{-\infty}^{\infty} |a_{ij}^{(k)}(\tau)| \frac{1}{\sqrt{4\pi\nu\tau}} e^{-(y-\zeta)^2/4\tau\nu} d\zeta d\tau, \tag{3.3.37}$$

where $\|g_j^0\| = \max_{0<y<\pi} \max_{0<t<T} |g_j^0(y,t)|$ is the L^∞ norm of the initial guess. Also note

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{4\pi\nu\tau}} e^{-(y-\zeta)^2/4\tau\nu} d\zeta = \frac{1}{\sqrt{4\pi\nu\tau}} \int_{-\infty}^{\infty} e^{-\zeta^2/4\tau\nu} d\zeta = 1,$$

so that

$$\|g_i^k\| \le \sum_j \|g_j^0\| \int_0^t |a_{ij}^{(k)}(\tau)| d\tau,$$

which means we have the same bounds as in the 1D case. □

3.3.3.2 Numerical illustration

We show an experiment for the NNWR algorithm in two dimension for the model problem

$$\partial_t u - (\partial_{xx} u + \partial_{yy} u) = 0, u(x,y,0) = \sin(2\pi x)\sin(3y).$$

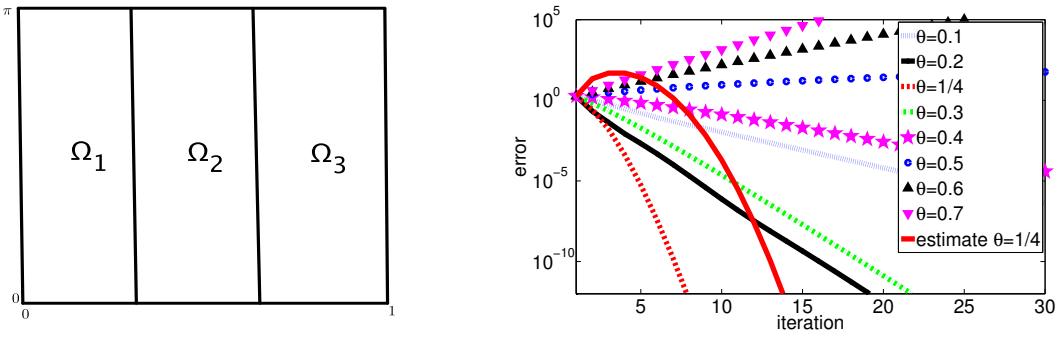


Figure 3.3.6: Decomposition of 2D domain into strips on the left, and convergence of NNWR using various relaxation parameters θ for $T = 0.2$ on the right

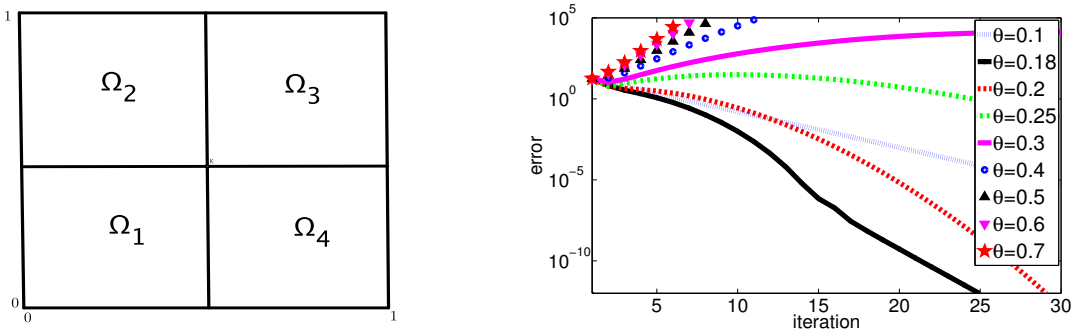


Figure 3.3.7: Decomposition of 2D domain with a crosspoint on the left, and convergence of NNWR using various relaxation parameters θ for $T = 0.2$ on the right

We decompose our domain $\Omega := (0, 1) \times (0, \pi)$ into three non-overlapping subdomains $\Omega_1 = (0, 2/5) \times (0, \pi)$, $\Omega_2 = (2/5, 3/4) \times (0, \pi)$, $\Omega_3 = (3/4, 1) \times (0, \pi)$, see Figure 3.3.6 on the left. We consider homogeneous Dirichlet boundary conditions along the physical boundaries. On the right, we plot the numerical errors of the NNWR algorithm for various θ and the theoretical estimates from Theorem 26 for $\theta = 1/4$ and again observe superlinear convergence.

We conclude this section with a numerical experiment not covered by our analysis: we decompose the two dimensional domain $\Omega := (0, 1) \times (0, 1)$ into four non-overlapping subdomains $\Omega_1 = (0, 1/2) \times (0, 1/2)$, $\Omega_2 = (0, 1/2) \times (1/2, 1)$, $\Omega_3 = (1/2, 1) \times (1/2, 1)$, $\Omega_4 = (1/2, 1) \times (0, 1/2)$, such that a cross point is present, see Figure 3.3.7 on the left. We take the initial condition $u(x, y, 0) = \sin(2\pi x) \sin(3\pi y)$. The right panel of the figure shows that the convergence of the NNWR algorithm remains superlinear, despite the presence of the cross point.

3.4 NNWR for Hyperbolic problems

We now present the Neumann-Neumann Waveform Relaxation (NNWR) algorithm for hyperbolic problems in space time. The method, as earlier, is based on a non-overlapping spatial domain decomposition, and the iteration involves subdomain solves

in space time with corresponding interface condition, followed by a correction step.

3.4.1 NNWR for two subdomains

To explain this algorithm for hyperbolic problems, we take the same model problem (2.4.1) and the same spatial decomposition of Ω into Ω_1 and Ω_2, as in Figure 2.3.1 in Subsection 2.3.1 from previous chapter.

The NNWR algorithm starts with an initial guess $w^0(x,t)$, $t \in (0,T)$ along the interface $\Gamma \times (0,T)$ and then computes simultaneously for $i = 1, 2$ with $k = 1, 2, ...$

$$\begin{aligned}
\partial_{tt} u_i^k - c^2(\boldsymbol{x})\triangle u_i^k &= f, && \text{in } \Omega_i, \\
u_i^k(\boldsymbol{x},0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_i, \\
\partial_t u_i^k(\boldsymbol{x},0) &= v_0(\boldsymbol{x}), && \text{in } \Omega_i, \\
u_i^k &= g, && \text{on } \partial\Omega_i \setminus \Gamma, \\
u_i^k &= w^{k-1}, && \text{on } \Gamma,
\end{aligned}
\qquad
\begin{aligned}
\partial_{tt} \psi_i^k - c^2(\boldsymbol{x})\triangle \psi_i^k &= 0, && \text{in } \Omega_i, \\
\psi_i^k(\boldsymbol{x},0) &= 0, && \text{in } \Omega_i, \\
\partial_t \psi_i^k(\boldsymbol{x},0) &= 0, && \text{in } \Omega_i, \\
\partial_{n_i} \psi_i^k &= \partial_{n_1} u_1^k + \partial_{n_2} u_2^k, && \text{on } \Gamma, \\
\psi_i^k &= 0, && \text{on } \partial\Omega_i \setminus \Gamma,
\end{aligned}$$

$$w^k(x,t) = w^{k-1}(x,t) - \theta \left[\psi_1^k \big|_{\Gamma\times(0,T)} + \psi_2^k \big|_{\Gamma\times(0,T)} \right], \tag{3.4.1}$$

where $\theta \in (0,1]$ is again a relaxation parameter.

Like for the DNWR method, we present here the one dimensional case $c(\boldsymbol{x}) = c$, with $\Omega = (-a,b)$, $\Omega_1 = (-a,0)$ and $\Omega_2 = (0,b)$. By linearity, it suffices to study the error equations, $f(\boldsymbol{x},t) = 0$, $g(\boldsymbol{x},t) = 0$, $u_0(\boldsymbol{x}) = v_0(\boldsymbol{x}) = 0$ in (3.4.1).

3.4.1.1 Convergence analysis

The Laplace transform of the NNWR algorithm in (3.4.1) for the error equations yields for the subdomain solutions

$$\hat{u}_1^k(x,s) = \frac{\hat{w}^{k-1}(s)}{\sinh(as/c)} \sinh\left((x+a)\frac{s}{c}\right), \quad \hat{u}_2^k(x,s) = -\frac{\hat{w}^{k-1}(s)}{\sinh(bs/c)} \sinh\left((x-b)\frac{s}{c}\right),$$

$$\hat{\psi}_1^k(x,s) = \frac{\hat{w}^{k-1}(s)\Psi(s)}{\cosh(as/c)} \sinh\left((x+a)\frac{s}{c}\right), \quad \hat{\psi}_2^k(x,s) = \frac{\hat{w}^{k-1}(s)\Psi(s)}{\cosh(bs/c)} \sinh\left((x-b)\frac{s}{c}\right),$$

where $\Psi(s) = [\coth(as/c) + \coth(bs/c)]$. Therefore, in Laplace space the updating condition in (3.4.1) becomes

$$\hat{w}^k(s) = \left[1 - \theta\left(2 + \frac{\coth(as/c)}{\coth(bs/c)} + \frac{\coth(bs/c)}{\coth(as/c)}\right)\right]^k \hat{w}^0(s), \quad k = 1, 2, \ldots \tag{3.4.2}$$

Theorem 27 (Convergence of NNWR, symmetric decomposition)**.** *For a symmetric decomposition, $a = b$, convergence is linear for the NNWR (3.4.1) with $\theta \in (0, 1/2)$, $\theta \neq 1/4$. If $\theta = 1/4$, convergence is achieved in two iterations.*

Proof. Inserting $a = b$ into equation (3.4.2), we obtain $\hat{w}^k(s) = (1-4\theta)^k \hat{w}^0(s)$, which has the simple back transform $w^k(t) = (1-4\theta)^k w^0(t)$. Thus for the NNWR method, the convergence is linear for $0 < \theta < 1/2, \theta \neq 1/4$. For $\theta = 1/4$, we have $w^1(t) = 0$. Hence, one more iteration produces the desired solution on the entire domain. □

We next analyze the case of an asymmetric decomposition, $a \neq b$. We recall the expression $G_b^a(s)$ from (2.4.5) to rewrite (3.4.2) in the form

$$\hat{w}^k(s) = \left\{(1-4\theta) - \theta\left(G_b^a(s) + G_a^b(s)\right)\right\}^k \hat{w}^0(s), \quad k = 1, 2, \ldots, \tag{3.4.3}$$

and we see that for NNWR, the choice $\theta = 1/4$ removes the linear factor.

Theorem 28 (Convergence of NNWR, asymmetric decomposition)**.** *Let $\theta = 1/4$. Then the NNWR algorithm (3.4.1) converges in at most $k+1$ iterations for two subdomains of lengths $a \neq b$, if the time window length T satisfies $T/k \leq 4\min\{a/c, b/c\}$, c being again the wave speed.*

Proof. With $\theta = 1/4$ we obtain from (3.4.3) with a similar calculation as in (2.4.7)

$$\hat{w}^k(s) = \left(-\frac{1}{4}\right)^k \left[G_b^a(s) + G_a^b(s)\right]^k \hat{w}^0(s) = \Bigg[-\sum_{m=1}^{\infty}\left(e^{-\frac{4ams}{c}} + e^{-\frac{4bms}{c}}\right) + \sum_{m=1}^{\infty}\sum_{n=1}^{\infty}(-1)^{n-1}\left(e^{-\frac{2(am+bn)s}{c}} + e^{-\frac{2(an+bm)s}{c}}\right)\Bigg]^k \hat{w}^0(s) = (-1)^k e^{-\frac{4aks}{c}}\hat{w}^0(s)$$
$$+ (-1)^k e^{-\frac{4bks}{c}}\hat{w}^0(s) + \left(\sum_{l>k}^{\infty} d_l^{(k)} e^{-\frac{4als}{c}} + \sum_{l>k}^{\infty} z_l^{(k)} e^{-\frac{4bls}{c}} + \sum_{m+n\geq 2k} j_{m,n}^{(k)} e^{-\frac{2(am+bn)s}{c}}\right)\hat{w}^0(s),$$

where $d_l^{(k)}, z_l^{(k)}, j_{m,n}^{(k)}$ are the corresponding coefficients. We use L2 from Appendix A

$$\mathcal{L}^{-1}\left\{e^{-\alpha s}\hat{g}(s)\right\} = H(t-\alpha)g(t-\alpha), \tag{3.4.4}$$

$H(t)$ being the Heaviside step function, to back transform and obtain

$$w^k(t) = (-1)^k w^0(t-4ak/c)H(t-4ak/c) + (-1)^k w^0(t-4bk/c)H(t-4bk/c) + \sum_{l>k}^{\infty} d_l^{(k)} w^0(t-4al/c)H(t-4al/c) + \sum_{l>k}^{\infty} z_l^{(k)} w^0(t-4bl/c)H(t-4bl/c) + \sum_{m+n\geq 2k} j_{m,n}^{(k)} w^0(t-2(am+bn)/c)H(t-2(am+bn)/c).$$

So for $T \leq 4k\min\left\{\frac{a}{c}, \frac{b}{c}\right\}$, we get $w^k(t) = 0$, and the conclusion follows. □

3.4.1.2 Numerical illustration

We perform numerical experiments to measure the actual convergence rate of the NNWR algorithm for the model problem

$$\begin{aligned} \partial_{tt}u - \partial_{xx}u &= 0, && x \in (-3,2), t > 0, \\ u(x,0) = 0,\ u_t(x,0) &= xe^{-x}, && -3 < x < 2, \\ u(-3,t) = -3te^3,\ u(2,t) &= 2te^{-2}, && t > 0, \end{aligned} \tag{3.4.5}$$

with $\Omega_1 = (-3,0)$ and $\Omega_2 = (0,2)$, so that $a = 3$ and $b = 2$ in (3.4.2). We again discretize the wave equation as in the previous chapter using centered finite differences in both space and time (Leapfrog scheme) on a grid with $\Delta x = \Delta t = 2\times 10^{-2}$. Using the initial guess $w^0(t) = t^2, t \in (0,T]$, we show in Figure 3.4.1 on the left the convergence curves for different values of the parameter θ for $T = 16$, and on the right the results for the best parameter $\theta = 1/4$ for different time window length T.

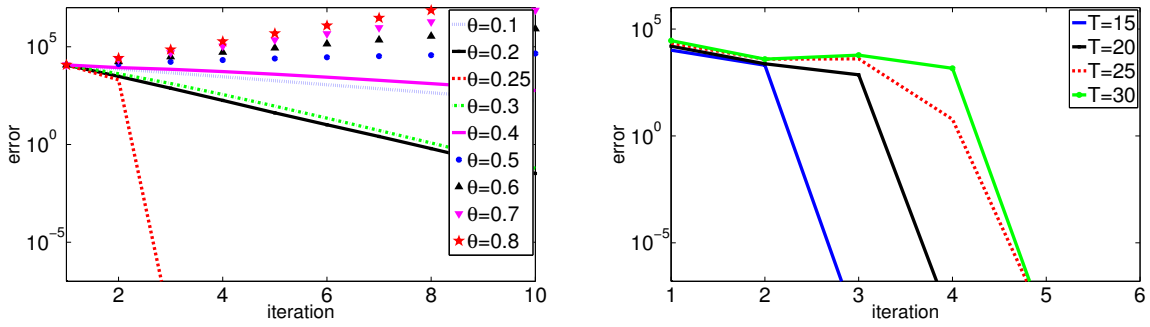


Figure 3.4.1: Convergence of NNWR with various values of θ for $T = 16$ on the left, and for various lengths T of the time window and $\theta = 1/4$ on the right

3.4.2 NNWR for multiple subdomains

We now introduce the NNWR algorithm for many subdomains for the model problem (2.4.1). Suppose Ω is partitioned into non-overlapping subdomains $\{\Omega_i,\ 1 \le i \le N\}$, exactly as in Subsection 3.3.2. The NNWR algorithm starts with an initial guess $g_{ij}^0(\boldsymbol{x}, t)$ along the interfaces $\Gamma_{ij} \times (0, T)$, $j \in \Lambda_i$, $i = 1, \ldots, N$, and then performs the following two-step iteration: one first solves Dirichlet subproblems on each Ω_i in parallel,

$$\begin{aligned} \partial_{tt} u_i^k - c^2 \Delta u_i^k &= f, && \text{in } \Omega_i, \\ u_i^k(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_i, \\ \partial_t u_i^k(\boldsymbol{x}, 0) &= v_0(\boldsymbol{x}), && \text{in } \Omega_i, \\ u_i^k &= g, && \text{on } \partial\Omega_i \setminus \Gamma_i, \\ u_i^k &= g_{ij}^{k-1}, && \text{on } \Gamma_{ij}, j \in \Lambda_i. \end{aligned} \tag{3.4.6}$$

One then solves Neumann subproblems on all subdomains,

$$\begin{aligned} \partial_{tt} \varphi_i^k - c^2 \Delta \varphi_i^k &= 0, && \text{in } \Omega_i, \\ \varphi_i^k(\boldsymbol{x}, 0) &= 0, && \text{in } \Omega_i, \\ \partial_t \varphi_i^k(\boldsymbol{x}, 0) &= 0, && \text{in } \Omega_i, \\ \varphi_i^k &= 0, && \text{on } \partial\Omega_i \setminus \Gamma_i, \\ \partial_{\boldsymbol{n}_{ij}} \varphi_i^k &= \partial_{\boldsymbol{n}_{ij}} u_i^k + \partial_{\boldsymbol{n}_{ji}} u_j^k, && \text{on } \Gamma_{ij}, j \in \Lambda_i. \end{aligned} \tag{3.4.7}$$

with the updating condition

$$g_{ij}^k(\boldsymbol{x}, t) = g_{ij}^{k-1}(\boldsymbol{x}, t) - \theta \left(\varphi_i^k \big|_{\Gamma_{ij} \times (0,T)} + \varphi_j^k \big|_{\Gamma_{ij} \times (0,T)} \right), \tag{3.4.8}$$

where $\theta \in (0, 1]$ is a relaxation parameter.

3.4.2.1 Convergence analysis

We prove our results for the one dimensional wave equation on the domain $\Omega := (0, L)$ with boundary conditions $u(0, t) = g_0(t)$ and $u(L, t) = g_L(t)$. We decompose Ω into non-overlapping subdomains $\Omega_i := (x_{i-1}, x_i)$, $i = 1, \ldots, N$, as shown in Figure 3.3.3, and define the subdomain length $h_i := x_i - x_{i-1}$, and $h_{\min} := \min_{1 \le i \le N} h_i$. Our initial

guess is denoted by $\{g_i^0(t)\}_{i=1}^{N-1}$ on the interfaces x_i. By linearity, we can assume $f = 0$, $g_0 = g_L = 0$ and $u_0 = 0$, which leads with $g_0^k(t) = g_N^k(t) = 0$ for all k to

$$\begin{aligned}
\partial_{tt}u_i^k - c^2\partial_{xx}u_i^k &= 0, \quad \text{in } \Omega_i, & \partial_{tt}\varphi_i^k - c^2\partial_{xx}\varphi_i^k &= 0, \quad \text{in } \Omega_i,\\
u_i^k(x,0) &= 0, \quad \text{in } \Omega_i, & \varphi_i^k(x,0) &= 0, \quad \text{in } \Omega_i,\\
\partial_t u_i^k(x,0) &= 0, \quad \text{in } \Omega_i, & \partial_t\varphi_i^k(x,0) &= 0, \quad \text{in } \Omega_i,\\
u_i^k(x_{i-1},t) &= g_{i-1}^{k-1}(t), & -\partial_x\varphi_i^k(x_{i-1},t) &= (\partial_x u_{i-1}^k - \partial_x u_i^k)(x_{i-1},t),\\
u_i^k(x_i,t) &= g_i^{k-1}(t), & \partial_x\varphi_i^k(x_i,t) &= (\partial_x u_i^k - \partial_x u_{i+1}^k)(x_i,t),
\end{aligned} \tag{3.4.9}$$

except for the first and last subdomains, where the Neumann conditions in the Neumann step are replaced by homogeneous Dirichlet conditions along the physical boundaries. The updated interface values for the next step will be

$$g_i^k(t) = g_i^{k-1}(t) - \theta\left(\varphi_i^k(x_i,t) + \varphi_{i+1}^k(x_i,t)\right). \tag{3.4.10}$$

After applying the Laplace transform to the NNWR algorithm in (3.4.9), we obtain for $i = 2,\ldots,N-2$ the updating condition in (3.4.10) as

$$\hat{g}_i^k = \hat{g}_i^{k-1} - \theta\left[\hat{g}_i^{k-1}\left(2 + \frac{2\gamma_i\gamma_{i+1}}{\sigma_i\sigma_{i+1}}\right) + \frac{\hat{g}_{i+1}^{k-1}}{\sigma_{i+1}}\left(\frac{\gamma_{i+2}}{\sigma_{i+2}} - \frac{\gamma_i}{\sigma_i}\right) + \frac{\hat{g}_{i-1}^{k-1}}{\sigma_i}\left(\frac{\gamma_{i-1}}{\sigma_{i-1}} - \frac{\gamma_{i+1}}{\sigma_{i+1}}\right) - \frac{\hat{g}_{i+2}^{k-1}}{\sigma_{i+1}\sigma_{i+2}} - \frac{\hat{g}_{i-2}^{k-1}}{\sigma_i\sigma_{i-1}}\right], \tag{3.4.11}$$

and for $i = 1, N-1$ as

$$\hat{g}_1^k = \hat{g}_1^{k-1} - \theta\left[\hat{g}_1^{k-1}\left(2 + \frac{\gamma_1\gamma_2}{\sigma_1\sigma_2} + \frac{\sigma_1\gamma_2}{\gamma_1\sigma_2}\right) + \frac{\hat{g}_2^{k-1}}{\sigma_2}\left(\frac{\gamma_3}{\sigma_3} - \frac{\sigma_1}{\gamma_1}\right) - \frac{\hat{g}_3^{k-1}}{\sigma_2\sigma_3}\right], \tag{3.4.12}$$

$$\hat{g}_{N-1}^k = \hat{g}_{N-1}^{k-1} - \theta\left[\hat{g}_{N-1}^{k-1}\left(2 + \frac{\gamma_{N-1}\gamma_N}{\sigma_{N-1}\sigma_N} + \frac{\sigma_N\gamma_{N-1}}{\gamma_N\sigma_{N-1}}\right) + \frac{\hat{g}_{N-2}^{k-1}}{\sigma_{N-1}}\left(\frac{\gamma_{N-2}}{\sigma_{N-2}} - \frac{\sigma_N}{\gamma_N}\right) - \frac{\hat{g}_{N-3}^{k-1}}{\sigma_{N-1}\sigma_{N-2}}\right], \tag{3.4.13}$$

where $\gamma_i = \cosh(h_i s/c)$, $\sigma_i = \sinh(h_i s/c)$.

Theorem 29 (Convergence of NNWR for multiple subdomains)**.** *Let $\theta = 1/4$. Then the NNWR algorithm (3.4.9)-(3.4.10) converges in $k+1$ iterations, if the time window length T satisfies $T/k \le 2\,h_{\min}/c$, c being the wave speed.*

Proof. With $\theta = 1/4$ the updating condition (3.4.11) becomes

$$\hat{g}_i^k(s) = -\frac{1}{4}\left(\hat{t}_{i,i}\hat{g}_i^{k-1}(s) + \hat{t}_{i,i+1}\hat{g}_{i+1}^{k-1}(s) + \hat{t}_{i,i-1}\hat{g}_{i-1}^{k-1}(s) - \hat{t}_{i,i+2}\hat{g}_{i+2}^{k-1}(s) - \hat{t}_{i,i-2}\hat{g}_{i-2}^{k-1}(s)\right), \tag{3.4.14}$$

where we defined

$$t_{i,i} := \frac{2}{\sigma_i\sigma_{i+1}}(\gamma_i\gamma_{i+1} - \sigma_i\sigma_{i+1}),\ \hat{t}_{i,i+1} := \frac{(\sigma_i\gamma_{i+2} - \gamma_i\sigma_{i+2})}{\sigma_i\sigma_{i+1}\sigma_{i+2}},$$
$$\hat{t}_{i,i-1} := \frac{(\sigma_{i+1}\gamma_{i-1} - \gamma_{i+1}\sigma_{i-1})}{\sigma_i\sigma_{i-1}\sigma_{i+1}},\ \hat{t}_{i,i+2} := \frac{1}{\sigma_{i+1}\sigma_{i+2}},\ \hat{t}_{i,i-2} := \frac{1}{\sigma_i\sigma_{i-1}}.$$

Similarly, we obtain for (3.4.12)

$$\hat{g}_1^k(s) = -\frac{1}{4}\left(\hat{t}_{1,1}\hat{g}_1^{k-1}(s) + \hat{t}_{1,2}\hat{g}_2^{k-1}(s) - \hat{t}_{1,3}\hat{g}_3^{k-1}(s)\right), \tag{3.4.15}$$

where we defined $\hat{t}_{1,1} := \left(\frac{\sigma_1\gamma_2}{\gamma_1\sigma_2} + \frac{\gamma_1\gamma_2}{\sigma_1\sigma_2} - 2\right)$, $\hat{t}_{1,2} = \frac{1}{\sigma_2}\left(\frac{\gamma_3}{\sigma_3} - \frac{\sigma_1}{\gamma_1}\right)$ and $\hat{t}_{1,3} = \frac{1}{\sigma_2\sigma_3}$. From (3.4.13), we obtain

$$\hat{g}_{N-1}^k(s) = -\frac{1}{4}\left(\hat{t}_{N-1,N-1}\hat{g}_{N-1}^{k-1}(s) + \hat{t}_{N-1,N-2}\hat{g}_{N-2}^{k-1}(s) - \hat{t}_{N-1,N-3}\hat{g}_{N-3}^{k-1}(s)\right), \tag{3.4.16}$$

where $\hat{t}_{N-1,N-1} = \left(\frac{\sigma_{N-1}\gamma_{N-2}}{\gamma_{N-1}\sigma_{N-2}} + \frac{\gamma_{N-1}\gamma_{N-2}}{\sigma_{N-1}\sigma_{N-2}} - 2\right)$, $\hat{t}_{N-1,N-2} = \frac{1}{\sigma_{N-2}}\left(\frac{\gamma_{N-3}}{\sigma_{N-3}} - \frac{\sigma_{N-1}}{\gamma_{N-1}}\right)$ and $\hat{t}_{N-1,N-3} = \frac{1}{\sigma_{N-2}\sigma_{N-3}}$. Note that $\hat{t}_{i,i+2} = \hat{t}_{i+2,i}, \hat{t}_{i,i+1} = -\hat{t}_{i+1,i}$. So by induction on (3.4.14)-(3.4.15) we can write

$$\hat{g}_i^k(s) = \sum_{j=-2n}^{2n}\left(-\frac{1}{4}\right)^n p_{i+j}^n\left(\hat{t}_{i+j,i+j-2}, \hat{t}_{i+j,i+j-1}, \ldots, \hat{t}_{i,i}, \ldots, \hat{t}_{i+j,i+j+1}, \hat{t}_{i+j,i+j+2}\right)\hat{g}_{i+j}^{k-n}(s), \tag{3.4.17}$$

and

$$\hat{g}_1^k(s) = \sum_{j=0}^{2n}\left(-\frac{1}{4}\right)^n b_{1+j}^n\left(\hat{t}_{1,1}, \ldots, \hat{t}_{1+j,2+j}, \hat{t}_{1+j,3+j}\right)\hat{g}_{1+j}^{k-n}(s), \tag{3.4.18}$$

where the coefficients p_{i+j}^n and b_{1+j}^n are homogeneous polynomials of degree n. A similar expression holds for $\hat{g}_{N-1}^k(s)$. A similar calculation as in (2.4.7) yields

$$\hat{t}_{i,i} = \frac{2\cosh\left((h_i - h_{i+1})s/c\right)}{\sinh(h_i s/c)\sinh(h_{i+1}s/c)} = 4\left(e^{-2h_i s/c} + e^{-2h_{i+1}s/c}\right)\left[1 + \sum_{m=1}^{\infty} e^{-2h_i m s/c} + \sum_{n=1}^{\infty} e^{-2h_{i+1}ns/c} + \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} e^{-2(mh_i + nh_{i+1})s/c}\right],$$

$$\hat{t}_{i,i+1} = \frac{\sinh\left((h_i - h_{i+2})s/c\right)}{\sinh(h_i s/c)\sinh(h_{i+1}s/c)\sinh(h_{i+2}s/c)} = 4\left[1 + \sum_{l=1}^{\infty} e^{-2lsh_i/c} + \sum_{m=1}^{\infty} e^{-2msh_{i+1}/c} + \sum_{n=1}^{\infty} e^{-2nsh_{i+2}/c} + \sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\left\{e^{-2(mh_i + nh_{i+1})s/c} + e^{-2(mh_i + nh_{i+2})s/c} + e^{-2(mh_{i+1} + nh_{i+2})s/c}\right\} + \sum_{l=1}^{\infty}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty} e^{-2(lh_i + mh_{i+1} + nh_{i+2})s/c}\right]\left(e^{-(h_{i+1} + 2h_{i+2})s/c} - e^{-(h_{i+1} + 2h_i)s/c}\right),$$

$$\hat{t}_{i,i+2} = \frac{1}{\sinh(h_{i+1}s/c)\sinh(h_{i+2}s/c)} = 4e^{-(h_{i+1} + h_{i+2})s/c}\left[1 + \sum_{m=1}^{\infty} e^{-2msh_{i+1}/c} + \sum_{n=1}^{\infty} e^{-2nsh_{i+2}/c} + \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} e^{-2(mh_{i+1} + nh_{i+2})s/c}\right],$$

$$\hat{t}_{1,1} = \frac{2\cosh\left((2h_1 - h_2)s/c\right)}{\sinh(2h_1 s/c)\sinh(h_2 s/c)} = 4\left(e^{-4h_1 s/c} + e^{-2h_2 s/c}\right)\left[1 + \sum_{m=1}^{\infty} e^{-4msh_1/c} + \sum_{n=1}^{\infty} e^{-2nsh_2/c} + \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} e^{-2(2mh_1+nh_2)s/c}\right],$$

$$\hat{t}_{1,2} = \frac{\cosh\left((h_1 - h_3)s/c\right)}{\cosh(h_1 s/c)\sinh(h_2 s/c)\sinh(h_3 s/c)} = 4\left[1 + \sum_{l=1}^{\infty}(-1)^l e^{-2lsh_1/c} + \sum_{m=1}^{\infty} e^{-2msh_2/c} + \sum_{n=1}^{\infty} e^{-2nsh_3/c} + \sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\left\{(-1)^m e^{-2(mh_1+nh_2)s/c} + (-1)^m e^{-2(mh_1+nh_3)s/c} + e^{-2(mh_2+nh_3)s/c}\right\} + \sum_{l=1}^{\infty}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}(-1)^l e^{-2(lh_1+mh_2+nh_3)s/c}\right]\left(e^{-(2h_1+h_2)s/c} + e^{-(h_2+2h_3)s/c}\right).$$

The argument also holds for the terms $\hat{t}_{i,i-1}, \hat{t}_{i,i-2}, \hat{t}_{1,3}, \hat{t}_{N-1,N-1}, \hat{t}_{N-1,N-2}, \hat{t}_{N-1,N-3}$ in a similar fashion. Now using these expressions we can write (3.4.17)-(3.4.18) as

$$\hat{g}_i^k(s) = (-1)^k\left[\left(e^{-2ksh_i/c} + e^{-2ksh_{i+1}/c}\right)\hat{g}_i^0(s) + \sum_{j=-2k}^{2k} q_{i+j}^k(s)\,\hat{g}_{i+j}^0(s)\right], \tag{3.4.19}$$

and

$$\hat{g}_1^k(s) = (-1)^k\left[\left(e^{-4h_1 ks/c} + e^{-2h_2 ks/c}\right)\hat{g}_1^0(s) + \sum_{j=0}^{2k} r_{1+j}^k(s)\,\hat{g}_{1+j}^0(s)\right], \tag{3.4.20}$$

where $q_{i+j}^k(s)$ and $r_{1+j}^k(s)$ are linear combinations of terms of the form e^{-sm} with $m \geq 2kh_l/c$ for some $l \in \{1, 2, \ldots N\}$. A similar expression holds for $\hat{g}_{N-1}^k(s)$. Now we use (3.4.4) to back transform (3.4.19)-(3.4.20) and obtain

$$g_i^k(t) = (-1)^k\left[g_i^0\left(t - \frac{2kh_i}{c}\right)H\left(t - \frac{2kh_i}{c}\right) + g_i^0\left(t - \frac{2kh_{i+1}}{c}\right)H\left(t - \frac{2kh_{i+1}}{c}\right) + \text{other terms}\right],$$

$$g_1^k(t) = (-1)^k\left[g_1^0\left(t - \frac{4kh_1}{c}\right)H\left(t - \frac{4kh_1}{c}\right) + g_1^0\left(t - \frac{2kh_2}{c}\right)H\left(t - \frac{2kh_2}{c}\right) + \text{other terms}\right]$$

and a similar expression for $g_{N-1}^k(t)$. So for $T \leq 2kh_{\min}/c$, we get $g_i^k(t) = 0$ for all i, and the conclusion follows. □

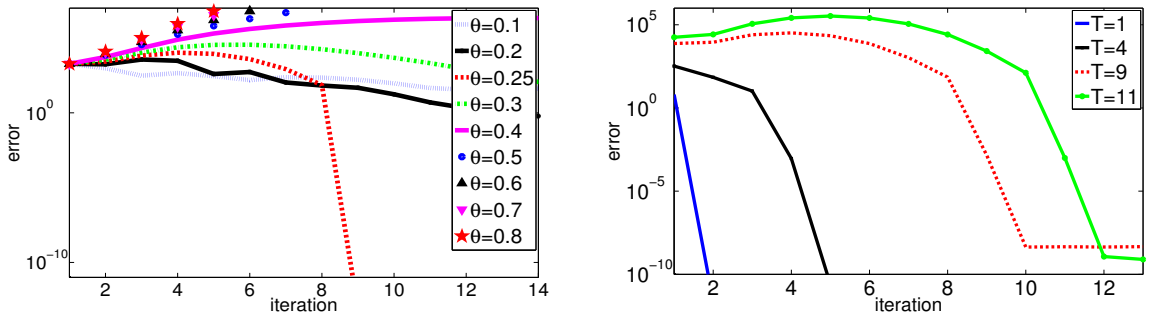


Figure 3.4.2: Convergence of NNWR with various values of θ for $T = 8$ on the left, and for various lengths T of the time window and $\theta = 1/4$ on the right

3.4.2.2 Numerical illustration

We perform numerical experiments to measure the actual convergence rate of the NNWR algorithm for the model problem

$$\begin{aligned} \partial_{tt}u - \partial_{xx}u &= 0, && x \in (0,5), t > 0, \\ u(x,0) = 0,\ u_t(x,0) &= 0, && 0 < x < 5, \\ u(0,t) = -3t^2,\ u(5,t) &= 2t^3, && t > 0, \end{aligned}$$

discretized using centered finite differences in both space and time on a grid with $\Delta x = \Delta t = 2\times 10^{-2}$. We use the initial guesses $g_j^0(t) = t^2, t \in (0,T]$ for $1 \le j \le 4$, and consider a decomposition of $(0,5)$ into five unequal subdomains, whose widths h_i are $0.6, 0.6, 0.5, 2.3, 1$ respectively. On the left panel of Figure 3.4.2, we show the convergence for different values of the parameter θ for $T = 8$, and on the right the results for the best parameter $\theta = 1/4$ for different time window length T. We see two-step convergence for $\theta = 1/4$ as expected, choosing the time window T small enough.

3.4.3 NNWR in 2D

We now formulate and analyze the NNWR algorithm for the 2D wave equation

$$\partial_{tt}u - c^2\Delta u = f(x,y,t), \quad (x,y) \in \Omega = (l,L)\times(0,\pi),\ t \in (0,T]$$

with initial condition $u(x,y,0) = u_0(x,y), \partial_t u(x,y,0) = v_0(x,y)$ and Dirichlet boundary conditions. To define the NNWR algorithm, we decompose Ω into strips of the form $\Omega_i = (x_{i-1}, x_i)\times(0,\pi)$, $l = x_0 < x_1 < \cdots < x_N = L$. We define the subdomain width $h_i := x_i - x_{i-1}$, and $h_{\min} := \min_{1\le i\le N} h_i$. Also we directly consider the error equations with $f(x,y,t) = 0, u_0(x,y) = 0 = v_0(x,y)$ and homogeneous Dirichlet boundary conditions. Given initial guesses $\{w_i^0(y,t)\}_{i=1}^{N-1}$ along the interface $\{x = x_i\}$, the NNWR algorithm, as a particular case of (3.4.6)-(3.4.7)-(3.4.8), is given by performing

iteratively for $k = 1, 2, \ldots$ and for $i = 1, \ldots, N$ the Dirichlet and Neumann steps

$$\begin{aligned}
\partial_{tt} u_i^k - c^2 \Delta u_i^k &= 0, \quad \text{in } \Omega_i, & \partial_{tt} \psi_i^k - c^2 \Delta \psi_i^k &= 0, \quad \text{in } \Omega_i, \\
u_i^k(x, y, 0) &= 0, & \psi_i^k(x, y, 0) &= 0, \\
\partial_t u_i^k(x, y, 0) &= 0, & \partial_t \psi_i^k(x, y, 0) &= 0, \\
u_i^k(x_{i-1}, y, t) &= w_{i-1}^{k-1}(y, t), & \partial_{n_i} \psi_i^k(x_{i-1}, y, t) &= (\partial_{n_{i-1}} u_{i-1}^k + \partial_{n_i} u_i^k)(x_{i-1}, y, t), \\
u_i^k(x_i, y, t) &= w_i^{k-1}(y, t), & \partial_{n_i} \psi_i^k(x_i, y, t) &= (\partial_{n_{i-1}} u_{i-1}^k + \partial_{n_i} u_i^k)(x_i, y, t), \\
u_i^k(x, 0, t) &= u_i^k(x, \pi, t) = 0, & \psi_i^k(x, 0, t) &= \psi_i^k(x, \pi, t) = 0,
\end{aligned} \tag{3.4.21}$$

except for the first and last subdomain, where in the Neumann step the Neumann conditions are replaced by homogeneous Dirichlet conditions along the physical boundaries. The update conditions are defined as

$$w_i^k(y, t) = w_i^{k-1}(y, t) - \theta \left(\psi_i^k(x_i, y, t) + \psi_{i+1}^k(x_i, y, t) \right).$$

3.4.3.1 Convergence analysis

In a similar way as in the case of the heat equation in Subsection 3.3.3.1, we reduce the problem to a collection of 1D problems by performing a Fourier transform along the y direction. Using a Fourier sine series along the y-direction, we obtain

$$u_i^k(x, y, t) = \sum_{n \geq 1} U_i^k(x, n, t) \sin(ny)$$

where

$$U_i^k(x, n, t) = \frac{2}{\pi} \int_0^\pi u_i^k(x, \eta, t) \sin(n\eta) d\eta.$$

The NNWR algorithm (3.4.21) therefore becomes a sequence of 1D problems indexed by n,

$$\frac{\partial^2 U_i^k}{\partial t^2}(x, n, t) - c^2 \frac{\partial^2 U_i^k}{\partial x^2}(x, n, t) + c^2 n^2 U_i^k(x, n, t) = 0, \tag{3.4.22}$$

with the boundary conditions for $U_i^k(x, n, t)$. We now define

$$\chi(\alpha, \beta, t) := \mathcal{L}^{-1} \left\{ \exp\left(-\beta \sqrt{s^2 + \alpha^2} \right) \right\}, \quad \text{Re}(s) > 0, \tag{3.4.23}$$

with s being the Laplace variable. Before presenting the main convergence theorem of this subsection, we prove the following Lemma.

Lemma 30. *We have the identity:*

$$\chi(\alpha, \beta, t) = \begin{cases} \delta(t - \beta) - \frac{\alpha\beta}{\sqrt{t^2 - \beta^2}} J_1\left(\alpha \sqrt{t^2 - \beta^2} \right), & t \geq \beta, \\ 0, & 0 < t < \beta, \end{cases}$$

where δ is the dirac delta function and J_1 is the Bessel function of first order given by

$$J_1(z) = \frac{1}{\pi} \int_0^\pi \cos\left(z \sin\varphi - \varphi \right) d\varphi.$$

Proof. Using the change of variable $r = \sqrt{s^2 + \alpha^2}$ we write

$$e^{-\beta r} = e^{-\beta s} - (e^{-\beta s} - e^{-\beta r}).$$

From the table [72, p. 245] we get

$$\mathcal{L}^{-1}\left\{e^{-\beta s}\right\} = \delta(t - \beta). \tag{3.4.24}$$

Also on page 263 of [72] we find

$$\mathcal{L}^{-1}\left\{e^{-\beta s} - e^{-\beta r}\right\} = \begin{cases} \frac{\alpha\beta}{\sqrt{t^2-\beta^2}} J_1\left(\alpha\sqrt{t^2-\beta^2}\right), & t > \beta, \\ 0, & 0 < t < \beta. \end{cases} \tag{3.4.25}$$

Thus subtracting (3.4.25) from (3.4.24) we obtain the expected inverse Laplace transform. □

Theorem 31 (Convergence of NNWR in 2D)**.** *Let $\theta = 1/4$. For $T > 0$ fixed, the NNWR algorithm (3.4.21) converges in $k+1$ iterations, if the time window length T satisfies $T/k < 2\,h_{\min}/c$, c being the wave speed.*

Proof. We take a Laplace transform in t of (3.4.22) to get

$$(s^2 + c^2n^2)\hat{U}_i^k - c^2\frac{d^2\hat{U}_i^k}{dx^2} = 0,$$

and now treat each n as in the one-dimensional analysis in the proof of Theorem 29, where the recurrence relations (3.4.11), (3.4.12) and (3.4.13) of the form

$$\hat{g}_i^k(s) = \sum_j A_{ij}^{(k)}(s)\hat{g}_j^0(s) \tag{3.4.26}$$

now become for each $n = 1, 2, \ldots$

$$\hat{W}_i^k(n,s) = \sum_j A_{ij}^{(k)}\left(\sqrt{s^2 + c^2n^2}\right)\hat{W}_j^0(n,s). \tag{3.4.27}$$

The equation (3.4.26) is of the form (3.4.19)-(3.4.20), that means $A_{ij}^{(k)}(s)$ are linear combination of terms of the form $e^{-\varrho s}$ for $\varrho \geq 2kh_l/c$ for some $l \in \{1, 2, \ldots N\}$. Therefore the coefficients $A_{ij}^{(k)}\left(\sqrt{s^2 + c^2n^2}\right)$ are sums of exponential functions of the form $e^{-\varrho\sqrt{s^2+c^2n^2}}$ for $\varrho \geq 2kh_l/c$. Hence we use the definition of χ in (3.4.23) to take the inverse Laplace transform of (3.4.27), and obtain

$$W_i^k(n,t) = \sum_j \sum_m \chi(cn, \varrho_{m,j}, t) * W_j^0(n,t),$$

with $\varrho_{m,j} \geq 2kh_{\min}/c$. So it is straightforward that for $t < 2kh_{\min}/c$, $W_i^k(n,t) = 0$ for each n, since the function χ is zero there by Lemma 30. Therefore the interface functions $w_i^k(y,t)$, given by $w_i^k(y,t) = \sum_{n\geq 1} W_i^k(n,t)\sin(ny)$ are also zero for all $i \in \{1, \ldots, N-1\}$. Hence one more iteration produces the desired solution on the entire domain. □

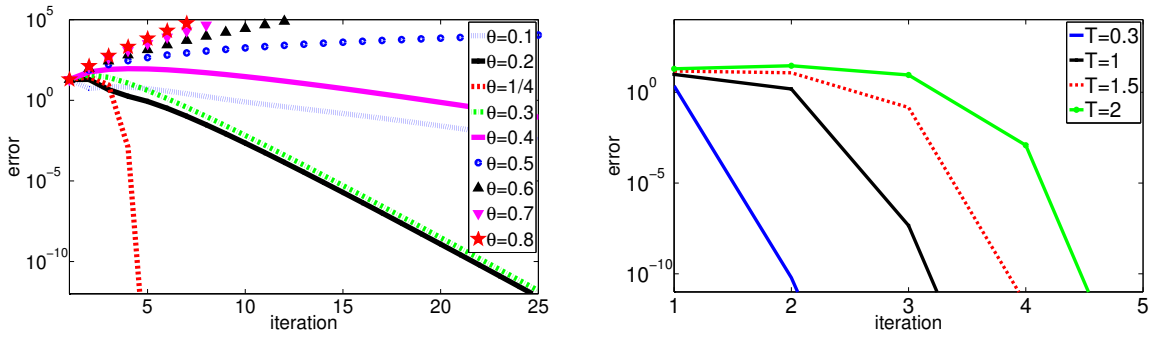


Figure 3.4.3: Convergence of NNWR with various values of θ for $T = 2$ on the left, and for various lengths T of the time window and $\theta = 1/4$ on the right

3.4.3.2 Numerical illustration

We show an experiment for the NNWR algorithm in two dimension for the model problem

$$\partial_{tt}u - (\partial_{xx}u + \partial_{yy}u) = 0, u(x, y, 0) = xy(x - 1)(y - \pi)(5x - 2)(4x - 3), u_t(x, y, 0) = 0,$$

with homogeneous Dirichlet boundary conditions. We discretize the wave equation using centered finite difference in both space and time (Leapfrog scheme) on a grid with $\Delta x = 5{\times}10^{-2}, \Delta y = 16{\times}10^{-2}, \Delta t = 4{\times}10^{-2}$. We decompose our domain $\Omega := (0, 1) \times (0, \pi)$ into three non-overlapping subdomains $\Omega_1 = (0, 2/5) \times (0, \pi)$, $\Omega_2 = (2/5, 3/4) \times (0, \pi)$, $\Omega_3 = (3/4, 1) \times (0, \pi)$. As initial guesses, we take $w_i^0(y, t) = t \sin(y)$. In Figure 3.4.3 we plot the convergence for different values of the parameter θ for $T = 2$ on the left panel, and on the right the results for the best parameter $\theta = 1/4$ for different time window length, T.

3.5 Conclusion

We have introduced another new class of space-time parallel algorithms, called Neumann-Neumann waveform relaxation (NNWR) for parabolic as well as hyperbolic problems. For the one-dimensional heat equation, we proved superlinear convergence for this algorithm for a particular choice of the relaxation parameter $\theta = 1/4$. Our convergence estimate holds for a decomposition into many subdomains, and we also gave an extension to two spatial dimensions. We have also defined the NNWR algorithm for the second order wave equation, and analyzed its convergence properties for 1D problems. We showed that for a particular choice of the relaxation parameter $\theta = 1/4$, convergence can be achieved in a finite number of steps for two as well as many subdomains. Like for the DNWR algorithm, convergence can be achieved in two iterations by choosing the time window lengh T small enough.

4 A Comparative Study with Schwarz Waveform Relaxation Methods

This chapter is devoted to the comparison of the performances of the DNWR and NNWR algorithms with the Schwarz Waveform Relaxation (SWR) algorithms with and without overlap. Like DNWR and NNWR, SWR methods can also be seen as an extension of Domain decomposition methods for elliptic PDEs to solve evolution problems. These algorithms are iterative methods solving the underlined PDE at each iteration on smaller subdomains, over the whole time window. We begin with an overview of various SWR algorithms applied to both heat and wave equations. Dirichlet transmission conditions are used in the classical SWR algorithm on the artificial boundaries. This classical iteration is convergent, but only with an overlap, and furthermore the convergence is slow. Absorbing boundary conditions can be used to make convergence faster, or to obtain convergence also for non-overlapping subdomains. These special types of boundary conditions have been introduced by Engquist and Majda [22] and Bayliss et al. [2] for solving PDEs on unbounded domains. The idea is to truncate the domain to a bounded one by introducing artificial boundaries to limit the area of computation. The use of these transmission conditions results in a convergent algorithm that is called Optimized SWR method. It can be applied with or without overlap, and the convergence is much faster than for the classical SWR in both cases. We finally present numerical results illustrating a comparison amongst various types of SWR algorithms and the newly proposed DNWR and NNWR for both heat and wave equations.

4.1 Schwarz Waveform Ralaxation Method

The name Schwarz Waveform Relaxation comes from the fact that these methods are of Schwarz domain decomposition type, for the decomposition of the spatial domain into overlapping subdomains, and of Waveform Relaxation type, for solving the equation over the entire time window. To solve time dependent problems of parabolic or hyperbolic type with SWR, we formulate algorithms without discretizing the original problems. We first subdivide the spatial domain into subdomains with or without overlap, and then solve iteratively time dependent subproblems, exchanging information along the artificial boundaries.

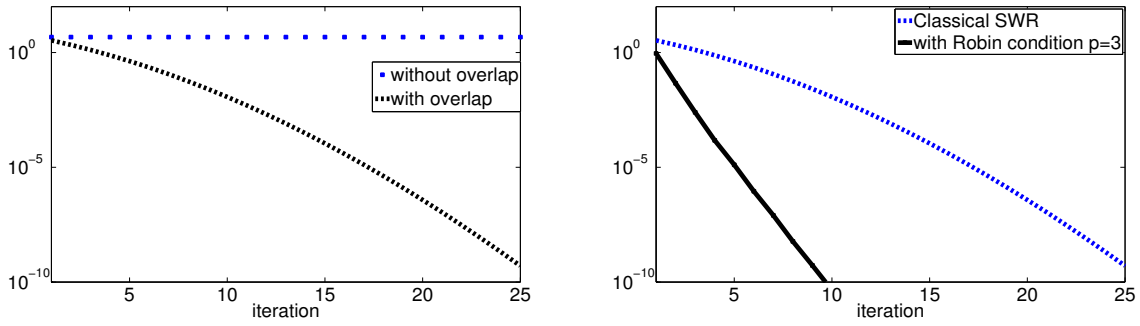


Figure 4.1.1: Convergence of the classical SWR algorithm for the one dimensional heat equation on the left, and comparison of convergence speed with respect to various transmission conditions on the right

We illustrate the method applied to the following evolution equations $\mathcal{L}_i(u) = f$, where $\mathcal{L}_1(u) := \partial_t u - \nu\Delta u$ and $\mathcal{L}_2(u) := \partial_{tt} u - c^2\Delta u$ in the spatial domain $\Omega = (0, L)$ on the time domain $(0, T]$. The spatial domain Ω is decomposed into subdomains $\Omega_j, 1 \le j \le N$ with or without overlap. For $i = 1, \ldots, N$, let Λ_i be the set of indices of all neighbouring subdomains to Ω_i and $\Gamma_{ij} := (\partial\Omega_i \setminus \partial\Omega) \cap \bar{\Omega}_j, j \in \Lambda_i$, $\Gamma_i := \partial\Omega_i \setminus \partial\Omega$. The SWR is the following Jacobi type iterative method: for $m = 1, 2$

$$\begin{array}{rcll} \mathcal{L}_m u_i^k & = & f, & \text{in } \Omega_i \times (0,T], \\ u_i^k(\cdot,0) & = & u_0,\ (\partial_t u_i^k(\cdot,0) = v_0 \text{ for } m=2), & \text{in } \Omega_i, \\ u_i^k & = & g, & \text{on } \partial\Omega_i \setminus \Gamma_i \times (0,T], \\ \mathcal{B}_{ij} u_i^k & = & \mathcal{B}_{ij} u_j^{k-1}, & \text{on } \Gamma_{ij} \times (0,T], j \in \Lambda_i. \end{array} \tag{4.1.1}$$

with an initial guess $\mathcal{B}_{ij} u_j^0$ along the interfaces ($\mathcal{B}_{ij}$ is a differential operator). This algorithm (4.1.1) is called parallel SWR method, whereas the corresponding Gauss-Seidel iteration is called alternating SWR method. In the following subsections we discuss the various choices of transmission conditions, and corresponding convergence results.

4.1.1 Classical SWR method

The most natural choice of the transmission conditions in (4.1.1) is to take Dirichlet conditions $u_i^k = u_j^{k-1}$, i.e., $\mathcal{B}_{ij} \equiv I$. The classical SWR algorithm was first presented in [40, 41] and further discussed in [30] for parabolic and hyperbolic equations. It is shown that the classical algorithm is convergent for a decomposition with overlap, and the larger the overlap, the faster the convergence. However, the method does not converge without overlap. For an illustration of this fact, see the left panel of Figure 4.1.1. A linear convergence result of this overlapping SWR iteration for the heat equation is shown in [40] on unbounded time intervals. Giladi and Keller [41] proved superlinear convergence of the method on bounded time intervals for the convection-diffusion equation. A similar behavior in higher dimension is analyzed in [65].

However, if the SWR method is applied to the wave equation, then convergence occurs in a finite number of iterations $k > cT/\delta$, where c is the wave propagation speed, and δ is the size of the overlap; for more details see [32, 31].

4.1.2 Optimized SWR method

In [30], Gander et al. showed that the use of Dirichlet transmission conditions at the artificial interfaces makes the convergence slow, which is also true for steady problems [69]. Therefore to make convergence faster, or to get convergence even for non-overlapping domain decomposition, we need to impose better transmission conditions along the artificial boundaries. For the heat or wave equation the transparent boundary conditions are obtained by introducing the Dirichlet-to-Neumann map. So if we choose $\mathcal{B}_{ij} \equiv \left(\partial_{\boldsymbol{n}_{ij}} + \mathrm{DtN}_{ji}\right)$ for $m = 1$, where $\boldsymbol{n}_{ij}$ denotes the unit outward normal for Ω_i and DtN_{ji} the Dirichlet-to-Neumann operator of the subdomain Ω_j on the interface Γ_{ij}, the algorithm (4.1.1) converges in N iterations with N subdomains, irrespective of the size of the overlap. But the computational cost of the DtN operators (non-local operators) is in general very high. So in optimized SWR methods, these exact conditions are approximated using local operators which result in absorbing boundary conditions. For the heat equation, a zeroth order Taylor approximation leads to Robin transmission conditions, with which we see much better convergence behaviors than with Dirichlet conditions; for the heat equation the method converges without overlap, and for the overlapping case, the convergence is much faster than with classical SWR. So unlike the classical algorithm, we can take Robin transmission conditions $\mathcal{B}_{ij} \equiv \left(\partial_{\boldsymbol{n}_{ij}} + p\right)$ for $m = 1$ to enhance the speed of the convergence with p being a real positive variable. A significant improvement in the convergence rate for the 1D heat equation with Robin transmission conditions can be seen in the right panel of Figure 4.1.1. For an optimized choice of the parameter p, see [27]. On the other hand, for the 1D wave equation, the exact transmission condition is given by $\mathcal{B}_{ij} \equiv (\partial_t \pm c\partial_x)$. In [42], Laurence Halpern characterized optimized SWR by the following paragraph:

'The purpose is to solve the space-time partial differential equation in each subdomain in parallel, and to transmit domain boundary information to the neighbors at the end of the time interval. The basic idea comes from the world of absorbing boundary conditions: for a model problem, approximations of the Dirichlet-Neumann map are developed, which can be written in the Fourier variables. These approximations lead to transmission conditions which involve time and tangential derivatives. The coefficients in these transmission conditions are in turn computed so as to optimize the convergence factor in the algorithm. This process can be written as a complex best approximation problem of a homographic type, and solved either explicitly or asymptotically. This gives a convergent algorithm that we call Optimized Schwarz Waveform Relaxation algorithm.'

For more details on optimized SWR for the advection-reaction-diffusion equations see [8, 29], and for the wave equation, see [32, 28].

4.2 Comparison with DNWR and NNWR

We present a comparison of the performance of the DNWR and NNWR algorithms for two subdomains with the SWR algorithms both for heat and wave equations. Both DNWR and NNWR are defined on a domain decomposition without overlap, whereas the classical SWR converges only for overlapping subdomains. However, the SWR method can be made convergent even for non-overlapping decomposition by introducing optimized transmission conditions. Using numerical experiments we now show better

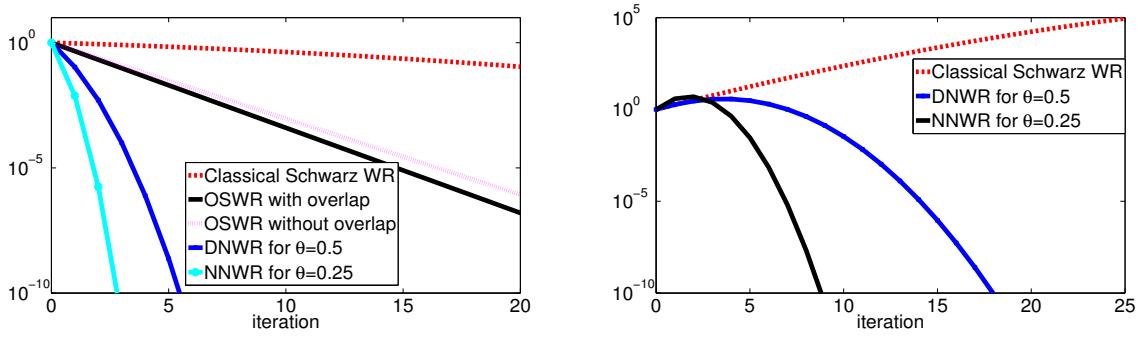


Figure 4.2.1: Comparison of estimates of various WR algorithms on the left for two subdomains, and on the right for many subdomains

Table 4.1: Comparison of estimates of various known WR methods for the heat equation.

Methods	2 subdomains	N equal subdomains
Classical Schwarz WR, overlap size δ	$\operatorname{erfc}\left(\frac{k\delta}{\sqrt{\nu T}}\right)$	$2^k \operatorname{erfc}\left(\frac{k\delta}{2\sqrt{\nu T}}\right)$
OSWR with overlap	$1-C(\Delta x)^{\frac{1}{3}}$ if $\beta \geq \frac{4}{3}$ $1-C(\Delta x)^{\frac{\beta}{4}}$ if $\beta < \frac{4}{3}$	no result so far
OSWR without overlap	$1-C(\Delta t)^{\frac{1}{4}}$	no result so far
DNWR	$\left(\frac{a-b}{a}\right)^k \operatorname{erfc}\left(\frac{kb}{2\sqrt{\nu T}}\right)$	$(N-2)^k \operatorname{erfc}\left(\frac{kh}{2\sqrt{\nu T}}\right)$
NNWR	$\left(\frac{(a-b)^2}{ab}\right)^k \operatorname{erfc}\left(\frac{kb}{\sqrt{\nu T}}\right)$	$\left(\frac{\sqrt{6}}{1-e^{-\frac{(2k+1)h^2}{\nu T}}}\right)^{2k} e^{-k^2h^2/\nu T}$

convergence behaviors of the substructuring WR methods in comparison to the SWR algorithms.

4.2.1 Heat equation

We first give a summary of the known estimates in Table 4.1 for other WR algorithms for the heat equation to compare the effectiveness of the new DNWR and NNWR algorithms. In Figure 4.2.1 we compare the theoretical estimates of various known WR methods with that of the DNWR and NNWR methods from Table 4.1, where $\Delta t = (\Delta x)^\beta$ and $\nu = 1$. We use an overlap δ of length $4\Delta x$, where $\Delta x = 1/50$, $\beta \approx 1.4114$, subdomain lengths $a = 3, b = 2$ and $C = 2$. For the many subdomains case, we take $\delta = 12\Delta x, N = 5, h = 1$. We observe that the estimates of the DNWR and NNWR algorithms indicate faster convergence than the other methods for this particular choice.

We run two numerical experiments for the model problem $\partial_t u = \nu u_{xx}$ on the spatial domain $\Omega = (0, 5)$, first with zero initial condition, $u_0(x) = 0$ and boundary conditions $u(0,t) = t\sin(\pi t), u(5,t) = t(t+1)$, and then with the initial condition $u_0(x) = x(x-3)(5-x)$ and boundary conditions $u(0,t) = t^2, u(5,t) = te^{-t}$. We split the domain into $\Omega_1 = (0,3)$ and $\Omega_2 = (3,5)$ for both experiments. In the left panel of Figure 4.2.2 we compare the performance of the DNWR and NNWR algorithms with

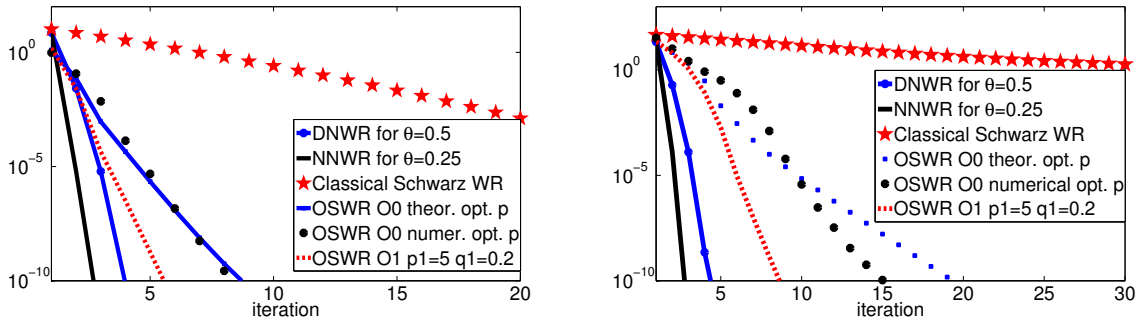


Figure 4.2.2: Comparison of DNWR and NNWR with SWR for the first experiment on the left, and second experiment on the right

Table 4.2: Comparison of steps needed for convergence for 1D wave equation.

Methods	2 subdomains	Many subdomains
DNWR	$T \leq 2kh_{\min}/c$	$T \leq kh_{\min}/c$
NNWR	$T \leq 4kh_{\min}/c$	$T \leq 2kh_{\min}/c$

the SWR algorithms with overlap. Here we use $\nu = 2$, $T = 1$ and an overlap of length $4\Delta x$, where $\Delta x = 1/40$. For the second experiment see the right panel of Figure 4.2.2, where we consider $\nu = 1$, $T = 2$ and an overlap of length $2\Delta x$, where $\Delta x = 1/50$. We observe that the DNWR and NNWR algorithms converge faster than the overlapping Schwarz WR. Only a higher order optimized Schwarz waveform relaxation algorithm comes close to the performance of the DNWR algorithm in this experiment.

4.2.2 Wave equation

We compare in Figure 4.2.3 the performance of the DNWR and NNWR algorithms with the SWR algorithms with and without overlap. We consider the model problem

$$\begin{aligned} \partial_{tt}u - \partial_{xx}u &= 0, & x \in (-3,2), t > 0, \\ u(x,0) = 0,\ u_t(x,0) &= xe^{-x}, & -3 < x < 2, \\ u(-3,t) = -3e^3t,\ u(2,t) &= 2te^{-2}, & t > 0, \end{aligned}$$

and for the overlapping Schwarz variant we use an overlap of length $24\Delta x$, where $\Delta x = 1/50$. For the DNWR, NNWR and non-overlapping SWR we consider a domain decomposition into two subdomains $\Omega_1 = (-3,0)$ and $\Omega_2 = (0,2)$. We observe that the DNWR and NNWR algorithms converge as fast as the Schwarz WR algorithms for smaller time windows T. Due to the local nature of the Dirichlet-to-Neumann operator in 1D [32], SWR converges in a finite number of iterations just like DNWR and NNWR. In higher dimensions, however, non-overlapping SWR will no longer converge in a finite number of steps, but DNWR and NNWR will; see Figure 4.2.4. In Table 4.2, we compare from Chapter 2 and 3 the theoretical results, that give the maximum number of iterations needed for the 1D wave equation to converge of the DNWR and the NNWR

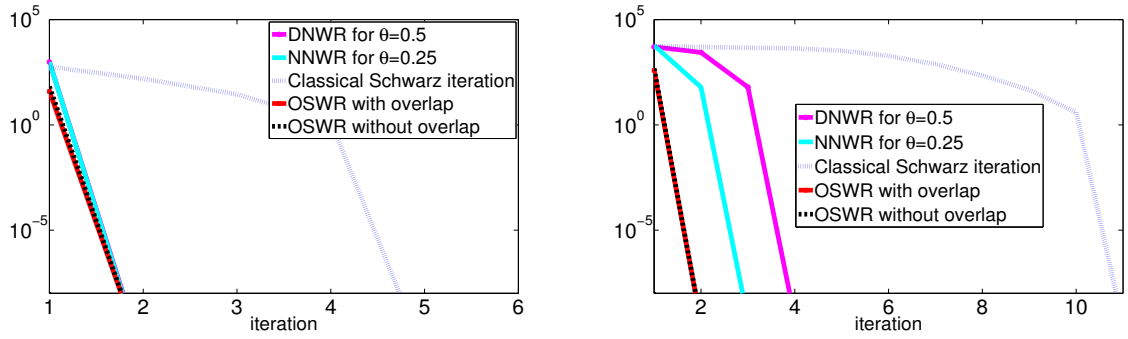


Figure 4.2.3: Comparison of DNWR, NNWR, and SWR in 1D for $T = 4$ on the left, and $T = 10$ on the right

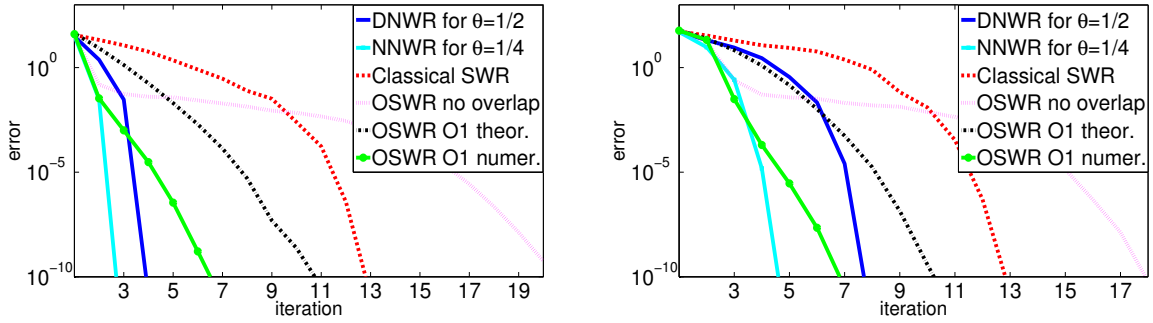


Figure 4.2.4: Comparison of DNWR, NNWR, and SWR for $T = 2$ in 2D for two subdomains on the left, and three subdomains on the right

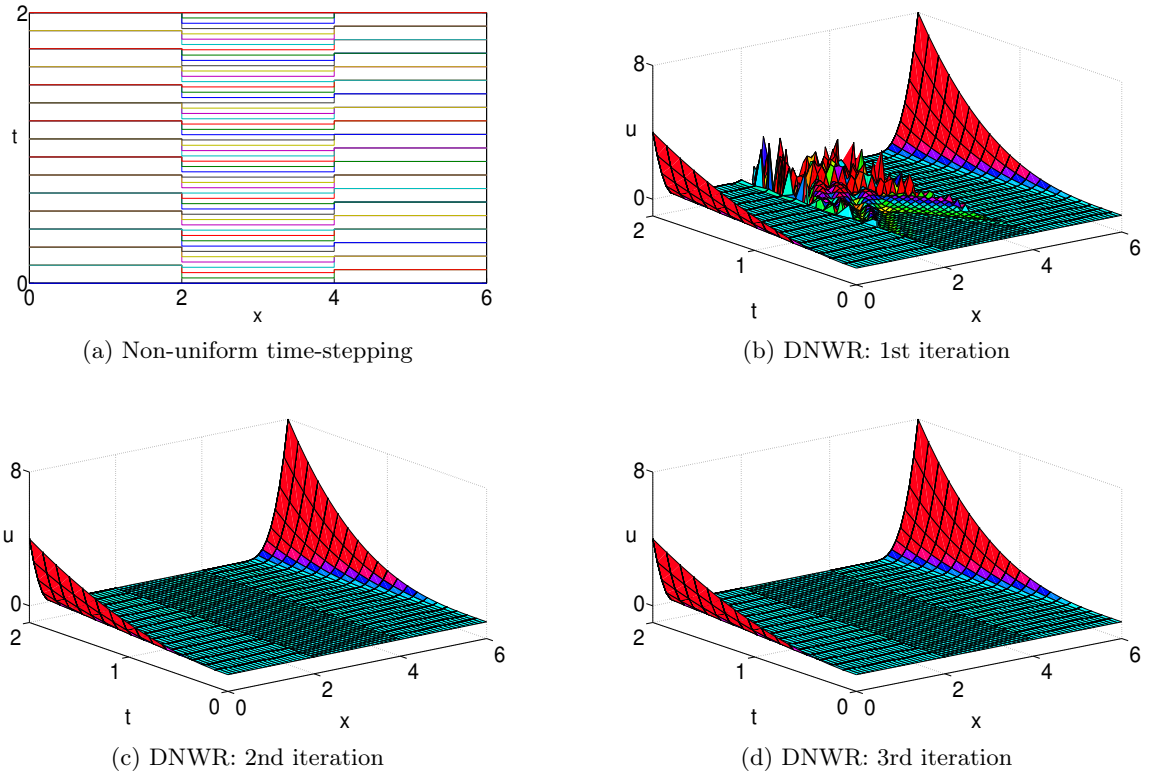
methods to the exact solution. For the 2D experiment we consider the model problem

$$\partial_{tt}u - (\partial_{xx}u + \partial_{yy}u) = 0, u(x, y, 0) = 0 = u_t(x, y, 0),$$

with Dirichlet boundary conditions $u(0, y, t) = t^2 \sin(y), u(1, y, t) = y(y - \pi)t^3$ and $u(x, 0, t) = 0 = u(x, \pi, t)$. We discretize the wave equation using centered finite difference in both space and time (Leapfrog scheme) as earlier on a grid with $\Delta x = 5\times10^{-2}, \Delta y = 16\times10^{-2}, \Delta t = 4\times10^{-2}$. We decompose our domain $\Omega := (0, 1) \times (0, \pi)$ for the two subdomains experiment into $\Omega_1 = (0, 3/5)\times(0, \pi)$ and $\Omega_2 = (3/5, 1)\times(0, \pi)$, and for the three subdomains experiment into $\Omega_1 = (0, 2/5) \times (0, \pi)$, $\Omega_2 = (2/5, 3/4) \times (0, \pi)$, $\Omega_3 = (3/4, 1) \times (0, \pi)$. We take a random initial guess to start the iteration, and for the overlapping SWR we use an overlap of length $2\Delta x$ in all the experiments. As optimized SWR methods, we implement first order methods with one parameter; for more details see [28]. On the left panel of Figure 4.2.4 we plot the convergence curves for two subdomains, and the same for three subdomains on the right.

4.3 Efficiency in parallelization

Both in the DNWR and NNWR algorithms, a significant speed up in computational time can be obtained if the subdomain problems are solved in parallel on supercomput-



(a) Non-uniform time-stepping

(b) DNWR: 1st iteration

(c) DNWR: 2nd iteration

(d) DNWR: 3rd iteration

Figure 4.3.1: Convergence of the DNWR method applied to the wave equation for non-uniform time steps for $\theta = 1/2$ for $T = 2$

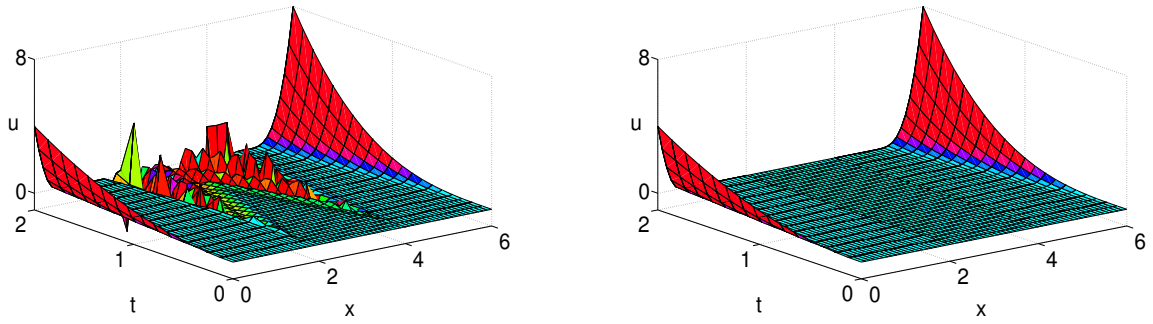
ers having many processors. At each iteration, one solves the space-time subproblem over the entire time window, before communicating interface data across subdomains. Here, one considers each subdomain problem as posed in space-time, so that a different spatial and time grid can be chosen for each subdomain. For such non-uniform mesh grid, boundary data is transmitted from one subdomain to a neighboring subdomain by introducing a suitable time projection. For two dimensional problems, the interface is one dimensional. Using ideas of merge sort one can compute the projection with linear cost, see [33] and the references therein. Even for higher dimensional interfaces, such an algorithm with linear complexity is still possible, see [34]. At this point, we show some numerical experiments for the DNWR and NNWR algorithms with different time grids for different subdomains and discontinuous wave speed across interfaces. We consider the model problem

$$\partial_{tt}u - c^2\partial_{xx}u = 0, u(x,0) = 0 = u_t(x,0),$$

with Dirichlet boundary conditions $u(0,t) = t^2, u(6,t) = t^3$. Suppose the spatial domain $\Omega := (0,6)$ is decomposed into three equal subdomains $\Omega_i, i = 1,2,3$, and the random initial guesses are used to start the DNWR or the NNWR iteration. For the spatial discretization, we take a uniform mesh with size $\Delta x = 1\times10^{-1}$, and for the time discretization, we use non-uniform time grids $\Delta t_i, i = 1,2,3$, as given in Table 4.3. In Figure 4.3.1 (a) we show the non-uniform time steps for different subdomains. For a particular relaxation parameter $\theta = 1/2$, the DNWR algorithm for the arrangement

Table 4.3: Propagation speed and time steps for different subdomains.

	Ω_1	Ω_2	Ω_3
wave speed c	$1/4$	2	$1/2$
time grids Δt_i	13×10^{-2}	39×10^{-3}	1×10^{-1}



(a) NNWR: 1st iteration

(b) NNWR: 2nd iteration

Figure 4.3.2: Convergence of the NNWR method applied to the wave equation for non-uniform time steps for $\theta = 1/4$ for $T = 2$

A3 as in Subsection 2.4.2 converges in three iterations for $T = 2$; see Figure 4.3.1 (b), (c) and (d). Figure 4.3.2 gives the convergence behavior of the NNWR algorithm for $T = 2$ with the same non-uniform time grids as in Table 4.3.

The above approach is different from the classical technique, where one discretizes first in time to obtain a sequence of steady problems, and then applies domain decomposition algorithms to solve the resulting elliptic problems at each time step in parallel. In this approach a uniform time step is enforced across the whole domain, which is not computationally efficient when the problem contains variable coefficients, or in case of simulations with multiple time scales. Another drawback of this approach is that it is completely sequential in the time direction, because one cannot start the solution of later time steps until the solution at the earlier time steps are known. However, although the iterations in NNWR method are parallel in the true sense, a DNWR iteration can only be performed if the boundary data from the neighboring subdomains are known. As a remedy we can reformulate the waveform iterations in a pipeline-parallel fashion, after an initial start-up cost. For a similar type of pipelined parallel computation that reformulates classical SWR, see [73].

We now raise the issue of scalability of the NNWR algorithm by giving some numerical examples both for the heat and wave equations. From Theorem 25 in Chapter 3, one can say that as long as h^2/T is constant, we expect similar convergence behavior for the heat equation. On the left panel of Figure 4.3.3, we plot convergence curves of NNWR by doubling the number of subdomains and then dividing the time window length by four. Similarly from Theorem 29, we expect identical convergence behavior of the NNWR algorithm for the wave equation as long as h/T is constant. We plot on the right of Figure 4.3.3 the convergence curves by doubling the number of subdomains and making the time window length half. One can therefore conclude that the algo-

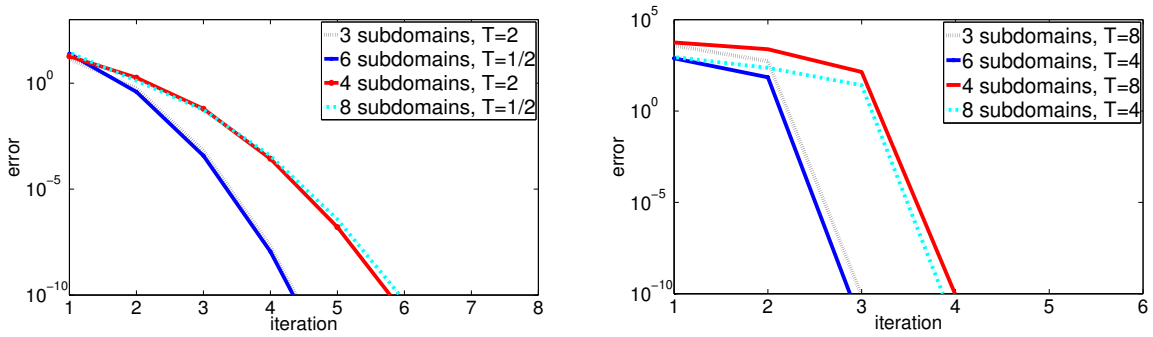


Figure 4.3.3: Graphs for comparing scalability of NNWR method for the heat equation on the left, and for the wave equation on the right

rithm is scalable for short time windows, but a coarse grid correction would give better convergence results for parabolic problems.

Although DNWR and NNWR are mainly space-time parallel algorithms (e.g. the problems to be solved in parallel are posed in space-time), each subdomain problem in a later stage can be parallelized in time using parareal or similar techniques with the expected gain in efficiency.

Another gain in efficiency for hyperbolic equations can be obtained as follows. Suppose the total number of degrees of freedom in space-time for the single domain problem is P. Then to solve the discretized hyperbolic PDE, we typically need $\mathcal{O}(P^n)$ operations for $n \geq 1$, where $n = 1$ if an explicit scheme is used, and otherwise $n > 1$ in higher dimensions, since one has to solve a linear system at each time step. If we decompose the domain into m subdomains, then the parallel running time for each iteration will be $\mathcal{O}(P^n/m^n)$. In Subsection 3.4.2 of Chapter 3 we have shown a convergence result in Theorem 29, which says that for m subdomains, our methods converge to the exact solution in 2 iterations for T small enough, so that the total running time becomes less than $\mathcal{O}(2P^n/m^n)$. Thus we expect to obtain meaningful speed up for 2D or 3D problems, where n is significantly larger than 1 in general.

4.4 Conclusion

We observe that for the heat equation both DNWR and NNWR algorithms converge faster than the overlapping Schwarz WR iteration. Only a higher order optimized SWR algorithm comes close to the performance of the DNWR algorithm in the numerical experiment. For the wave equation, we also get faster convergence in the DNWR and NNWR algorithms. It is also evident from the above numerical experiments that among the two new substructuring WR methods, NNWR converges faster for both heat and wave equations. But in comparison to DNWR, the NNWR has to solve twice the number of subproblems (once for Dirichlet subproblems, and once for Neumann subproblems) on each subdomain at each iteration. Therefore the computational cost is almost double for the NNWR than for the DNWR algorithm at each step. However, we get better convergence behavior with the NNWR in terms of iteration numbers. Also the convergence is symmetric in NNWR with respect to the size of the subdomains, unlike

in DNWR. In addition, a natural extension of the NNWR method for two subdomains is possible to multiple subdomains, or even to higher dimensions, as shown in Chapter 3. But for the DNWR method, the extension to multiple subdomains is tricky in the sense that, one can use different types of arrangements of interface conditions. The DNWR method is analyzed in Chapter 2 for one of these arrangements.

5

Applications to Optimal Control Problems

In this chapter, we focus the application of substructuring methods, both for steady and evolution problems, to solve optimal control problems. As discussed in Chapter 1, there are several types of PDEs that can act as constraints within optimization problems, for example elliptic, hyperbolic or parabolic ones. So, one needs to focus on the development of fast and effective methods for their numerical solution. DD methods are often used as powerful parallel computing tools for large-scale optimization problems governed by PDEs. Heinkenschloss et al. [44] presented the numerical behavior of a particular type of non-overlapping spatial DD method for the solution of linear–quadratic parabolic optimal control problems, that arose in the determination of groundwater pollutant sources from measurements of concentrations, and identification of sources, or cooling processes in metallurgy. In Section 5.2 we analyze the behavior of such spatial domain decomposition methods of substructuring type, and present a convergence analysis in detail for a model control problem with PDE constraint of elliptic type. The analysis for both Dirichlet-Neumann and Neumann-Neumann algorithms is presented with numerical tests. We also define the DNWR and the NNWR methods for parabolic optimal control problems and present numerical results for time-dependent models in Section 5.3.

5.1 Optimal Control Problems

To begin with, we briefly explain the mathematical structure of optimal control problems. We have the following essential elements in an optimal control problem:

- a control function u to be chosen freely from an admissible set U_{ad}.
- a state variable y and a state equation which connects the state with the control. In our discussion, we only consider this state equation to be a partial differential equation (PDE). The state y is the solution of the PDE that depends on the control. So, any change in the control u leads to a change in the solution state y. The state is therefore uniquely determined by the PDE for a particular choice of u.

- a functional J to be minimized. It is called the cost function or the objective function, that depends on both state y and control u. One has to choose u in such a way that $J(y,u)$ is minimized. Such controls u are called optimal.

For a simple set up of a finite dimensional problem, we explain the optimality conditions and other necessary framework. Suppose $J : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ denotes a cost functional to be minimized and the matrices $A_{n\times n}$, $B_{n\times m}$ are given. Without loss of generality, we can consider the finite dimensional optimal control problem as:

$$\begin{aligned} &\min\ J(y,u), \\ &Ay = Bu, \quad u \in U_{\text{ad}}, \end{aligned} \tag{5.1.1}$$

because for a more general system with a forcing term $Ay = Bu + f$, one can make a shift $z = y - y_{\text{par}}$, with y_{par} being a particular solution to $Ay_{\text{par}} = f$, and finally one obtains $Az = Bu$. So we may choose the control function u arbitrarily in U_{ad} to determine the state function y uniquely so that the cost function is minimized.

If A has an inverse, then $S : \mathbb{R}^m \to \mathbb{R}^n$, defined by $S = A^{-1}B$ is called the solution matrix. Thus $y = Su$ is determined for each control u. For more details about these basic notions, see [83, 56, 53]. The key tool to obtain the first order optimality conditions is given by the following result.

Theorem (First Order Optimality Condition)**.** *Suppose U_{ad} is a convex subset of $\mathbb{R}^m$. Then any optimal control $\bar{u}$ for (5.1.1) satisfies the variational inequality*

$$f'(\bar{u})\,(u - \bar{u}) \geq 0,\ \forall u \in U_{ad}, \tag{5.1.2}$$

where $f(u) := J\,(Su, u)\,.$

For a proof of the result, see [83, p. 63]. This first order condition is a necessary condition for local optimality. But if in addition J is a convex function, then this becomes also a sufficient condition for global optimality. However, using the first order optimality condition (5.1.2), we can write the optimality system as:

$$\begin{aligned} Ay &= Bu, && \forall u \in U_{\text{ad}} \\ A^{\mathrm{T}}p &= \nabla_y J(y,u), && \\ \left\langle B^{\mathrm{T}}p + \nabla_u J(y,u), v-u\right\rangle_{\mathbb{R}^m} &\geq 0, && \forall v \in U_{\text{ad}}. \end{aligned} \tag{5.1.3}$$

where p is the adjoint variable associated to (y,u).

The function $L : \mathbb{R}^{2n+m} \to \mathbb{R}$, $L(y,u,p) := J(y,u) - \langle Ay - Bu, p\rangle_{\mathbb{R}^n}$, is called the Lagrangian function or Lagrangian corresponding to the problem. In terms of L, we can rewrite the optimality conditions (5.1.3) as:

$$\begin{aligned} \nabla_p L(y,u,p) &= 0, \\ \nabla_y L(y,u,p) &= 0, \\ \langle \nabla_u L(y,u,p), v-u\rangle_{\mathbb{R}^m} &\geq 0,\ \forall v \in U_{\text{ad}}. \end{aligned} \tag{5.1.4}$$

If $U_{\text{ad}} = \mathbb{R}^m$, then the last variational inequality in (5.1.3) or (5.1.4) will be reduced to: $B^{\mathrm{T}}p + \nabla_u J(y,u) = 0$ or $\nabla_u L(y,u,p) = 0$. We use this system as a reference in the rest of our discussion.

We now consider linear-quadratic optimal control problems with PDE constraints and apply Domain Decomposition methods, more specifically the substructuring methods to solve the state and corresponding adjoint equations. For similar applications of substructuring methods to solve linear–quadratic elliptic optimal control problems, see [45, 46]. Our application also includes the formulation of DNWR and NNWR algorithms to solve space-time problems. Although our techniques can be extended for multiple subdomains, we only consider a decomposition into two non-overlapping subdomains for the sake of simplicity. For further details on DD methods applied to optimal control problems, see [4, 5, 6, 7].

5.2 Steady problems

We now formulate and analyze substructuring algorithms for the following model elliptic optimal control problem, which originates from the optimal stationary heating example with controlled heat source, on a bounded domain $\Omega \subset \mathbb{R}$:

$$\text{Minimize } J(y,u) = \tfrac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2\,dx + \tfrac{\lambda}{2}\int_\Omega u^2(x)dx, \tag{5.2.1}$$

$$\text{subject to } \begin{cases} -\nabla\cdot(\kappa(x)\nabla y(x)) = u(x), & x\in\Omega,\\ y(x)=0, & x\in\partial\Omega, \end{cases} \tag{5.2.2}$$

where $\Omega=(0,1)$ and $u\in U_{\text{ad}}$. In this setting, y denotes the temperature at a particular point, $\kappa(x)$ is the thermal conductivity of Ω, and $\lambda>0$ is a regularization parameter. Here we consider $U_{\text{ad}}=\mathbb{R}$ as the set of all feasible controls.

Let us consider the Lagrangian,

$$L(y,u,p) = J(y,u) + \int_0^1 p(x)\left(u(x)+\nabla\cdot(\kappa(x)\nabla y(x))\right)dx,$$

with p being the adjoint variable. Then from the conditions (5.1.4), the adjoint equation corresponding to the problem (5.2.1)-(5.2.2) becomes

$$\begin{cases} -\nabla\cdot(\kappa(x)\nabla p(x)) = y(x)-\hat{y}(x), & x\in\Omega,\\ p(x)=0, & x\in\partial\Omega, \end{cases} \tag{5.2.3}$$

together with the optimality condition:

$$p(x)+\lambda u(x)=0. \tag{5.2.4}$$

The underlying elliptic PDEs (5.2.2) and (5.2.3) then can be solved using Dirichlet-Neumann or Neumann-Neumann Domain Decomposition methods. By linearity it suffices to consider the homogeneous problems, $\hat{y}(x)=0$, and to analyze convergence to zero, since the corresponding error equations coincide with these homogeneous equations.

5.2.1 Dirichlet-Neumann algorithm

We first apply the Dirichlet-Neumann algorithm to solve the PDEs (5.2.2) and (5.2.3), coupled through the condition (5.2.4). Suppose the domain $\Omega=(0,1)$ is decomposed

into two non-overlapping subdomains, $\Omega_1 = (0, \alpha)$ and $\Omega_2 = (\alpha, 1)$. We denote by y_i, u_i, p_i the restriction of y, u, p to Ω_i, $i = 1, 2$. Then given two initial guesses h_y^0 and h_p^0 at the interface $\{x = \alpha\}$, we write the Dirichlet-Neumann algorithm for both state and adjoint equations as: for $k = 1, 2, \ldots$ compute

$$\begin{array}{rcll} -\nabla \cdot \left(\kappa(x)\nabla y_1^k\right) & = & u_1^k, & \text{in } \Omega_1, \\ y_1^k(0) & = & 0, & \\ y_1^k(\alpha) & = & h_y^{k-1}, & \end{array} \qquad \begin{array}{rcll} -\nabla \cdot \left(\kappa(x)\nabla p_1^k\right) & = & y_1^k, & \text{in } \Omega_1, \\ p_1^k(0) & = & 0, & \\ p_1^k(\alpha) & = & h_p^{k-1}, & \end{array} \tag{5.2.5}$$

$$\begin{array}{rl} -\nabla \cdot \left(\kappa(x)\nabla y_2^k\right) = u_2^k, & \text{in } \Omega_2, \\ \partial_x y_2^k(\alpha) = \partial_x y_1^k(\alpha), & \\ y_2^k(1) = 0, & \end{array} \qquad \begin{array}{rl} -\nabla \cdot \left(\kappa(x)\nabla p_2^k\right) = y_2^k, & \text{in } \Omega_2, \\ \partial_x p_2^k(\alpha) = \partial_x p_1^k(\alpha), & \\ p_2^k(1) = 0, & \end{array} \tag{5.2.6}$$

and the update conditions

$$\begin{array}{rcl} h_y^k & = & \theta_y y_2^k(\alpha) + (1 - \theta_y) h_y^{k-1}, \\ h_p^k & = & \theta_p p_2^k(\alpha) + (1 - \theta_p) h_p^{k-1}, \end{array} \tag{5.2.7}$$

where $\theta_y, \theta_p \in (0, 1)$ are two relaxation parameters, one for the state equation (5.2.2) and another for the adjoint equation (5.2.3).

5.2.1.1 Convergence analysis

We analyze the convergence for the DN algorithm (5.2.5)-(5.2.6)-(5.2.7) for the special case $\kappa(x) = 1$. By (5.2.4), we can write $u_i^k = -p_i^k/\lambda$ for $i = 1, 2$. We denote by $D^{(m)} := \frac{d^m}{dx^m}$. Now eliminating p_1^k from (5.2.5), we obtain

$$D^{(4)} y_1^k + \frac{1}{\lambda} y_1^k = 0, \tag{5.2.8}$$

with the boundary conditions

$$\begin{array}{cccc} (i) & y_1^k(0) = 0, & (ii) & y_1^k(\alpha) = h_y^{k-1}, \\ (iii) & D^{(2)} y_1^k(0) = 0, & (iv) & D^{(2)} y_1^k(\alpha) = h_p^{k-1}/\lambda. \end{array}$$

The third and fourth conditions come from the equation $D^{(2)} y_1^k = -u_1^k = p_1^k/\lambda$, which is a combination of (5.2.5) and (5.2.4). Now since $\lambda > 0$, we set $\mu^4 := 1/\lambda$. Then the general solution of (5.2.8) is

$$y_1^k(x) = \frac{1}{2}\left(A_1 e^{\mu x/\sqrt{2}} + A_2 e^{-\mu x/\sqrt{2}}\right) \cos\left(x\mu/\sqrt{2}\right) + \frac{1}{2}\left(B_1 e^{\mu x/\sqrt{2}} + B_2 e^{-\mu x/\sqrt{2}}\right) \sin\left(x\mu/\sqrt{2}\right).$$

Using the boundary conditions $(i), (ii)$, we get: $A_1 + A_2 = 0$ and

$$A_1 \sinh\left(\mu\alpha/\sqrt{2}\right) \cos\left(\mu\alpha/\sqrt{2}\right) + \frac{1}{2}\left(B_1 e^{\mu\alpha/\sqrt{2}} + B_2 e^{-\mu\alpha/\sqrt{2}}\right) \sin\left(\alpha\mu/\sqrt{2}\right) = h_y^{k-1}. \tag{5.2.9}$$

Finally using $(iii), (iv)$, we obtain $B_1 - B_2 = 0$ and

$$-A_1\mu^2 \cosh\left(\mu\alpha/\sqrt{2}\right) \sin\left(\mu\alpha/\sqrt{2}\right) + B_1\mu^2 \sinh\left(\mu\alpha/\sqrt{2}\right) \cos\left(\mu\alpha/\sqrt{2}\right) = h_p^{k-1}/\lambda. \tag{5.2.10}$$

Set

$$\gamma_1 = \cosh\left(\frac{\mu\alpha}{\sqrt{2}}\right),\ \gamma_2 = \cosh\left(\frac{\mu(1-\alpha)}{\sqrt{2}}\right),\ \sigma_1 = \sinh\left(\frac{\mu\alpha}{\sqrt{2}}\right),\ \sigma_2 = \sinh\left(\frac{\mu(1-\alpha)}{\sqrt{2}}\right),$$
$$\eta_1 = \cos\left(\frac{\mu\alpha}{\sqrt{2}}\right),\quad \eta_2 = \cos\left(\frac{\mu(1-\alpha)}{\sqrt{2}}\right),\quad \rho_1 = \sin\left(\frac{\mu\alpha}{\sqrt{2}}\right),\quad \rho_2 = \sin\left(\frac{\mu(1-\alpha)}{\sqrt{2}}\right).$$

Solving (5.2.9)-(5.2.10) we get

$$y_1^k = A_1 \sinh\left(\mu x/\sqrt{2}\right)\cos\left(\mu x/\sqrt{2}\right) + B_1 \cosh\left(\mu x/\sqrt{2}\right)\sin\left(\mu x/\sqrt{2}\right), \tag{5.2.11}$$

where

$$A_1 = \frac{h_y^{k-1}\sigma_1\eta_1 - \mu^2 h_p^{k-1}\gamma_1\rho_1}{\sigma_1^2+\rho_1^2}, \qquad B_1 = \frac{h_y^{k-1}\gamma_1\rho_1 + \mu^2 h_p^{k-1}\sigma_1\eta_1}{\sigma_1^2+\rho_1^2}.$$

Similarly, eliminating p_2^k from (5.2.6), we obtain

$$\begin{aligned} D^{(4)}y_2^k + \frac{1}{\lambda}y_2^k &= 0, \\ D^{(1)}y_2^k(\alpha) = D^{(1)}y_1^k(\alpha), &\qquad D^{(3)}y_2^k(\alpha) = D^{(3)}y_1^k(\alpha), \\ y_2^k(1) = 0, &\qquad D^{(2)}y_2^k(1) = 0. \end{aligned} \tag{5.2.12}$$

Solving the boundary value problem (5.2.12) yields

$$y_2^k(x) = C_1 \sinh\left(\frac{\mu(1-x)}{\sqrt{2}}\right)\cos\left(\frac{\mu(1-x)}{\sqrt{2}}\right) + D_1 \cosh\left(\frac{\mu(1-x)}{\sqrt{2}}\right)\sin\left(\frac{\mu(1-x)}{\sqrt{2}}\right), \tag{5.2.13}$$

where

$$\begin{aligned} C_1 &= -A_1\frac{\sigma_1\sigma_2\rho_1\rho_2 + \gamma_1\gamma_2\eta_1\eta_2}{\eta_2^2+\sigma_2^2} + B_1\frac{\gamma_1\eta_1\sigma_2\rho_2 - \sigma_1\rho_1\gamma_2\eta_2}{\eta_2^2+\sigma_2^2}, \\ D_1 &= -A_1\frac{\gamma_1\eta_1\sigma_2\rho_2 - \sigma_1\rho_1\gamma_2\eta_2}{\eta_2^2+\sigma_2^2} - B_1\frac{\sigma_1\sigma_2\rho_1\rho_2 + \gamma_1\gamma_2\eta_1\eta_2}{\eta_2^2+\sigma_2^2}. \end{aligned}$$

Thus using (5.2.11) and (5.2.13) the update conditions (5.2.7) reduce to

$$\begin{aligned} h_y^k &= (1-\theta_y)\,h_y^{k-1} + \theta_y\left(C_1\sigma_2\eta_2 + D_1\gamma_2\rho_2\right), \\ h_p^k &= (1-\theta_p)\,h_p^{k-1} - \frac{\theta_p}{\mu^2}\left(C_1\gamma_2\rho_2 - D_1\sigma_2\eta_2\right), \end{aligned}$$

from where more simplification gives

$$\begin{aligned} h_y^k &= (1-\theta_y)\,h_y^{k-1} + \theta_y\left(h_y^{k-1}v - \mu^2 h_p^{k-1}w\right), \\ h_p^k &= (1-\theta_p)\,h_p^{k-1} + \theta_p\left(\tfrac{h_y^{k-1}}{\mu^2}w + h_p^{k-1}v\right), \end{aligned} \tag{5.2.14}$$

with

$$v(\alpha,\mu) = -\frac{\rho_1\rho_2\eta_1\eta_2 + \sigma_1\sigma_2\gamma_1\gamma_2}{(\sigma_1^2+\rho_1^2)(\eta_2^2+\sigma_2^2)}, w(\alpha,\mu) = \frac{\gamma_1\sigma_1\rho_2\eta_2 - \rho_1\eta_1\gamma_2\sigma_2}{(\sigma_1^2+\rho_1^2)(\eta_2^2+\sigma_2^2)}.$$

We are now in a position to present the convergence results for the DN algorithm.

Theorem 32 (Convergence in the symmetric case)**.** *When the subdomains are of the same size, $\alpha = 1/2$ in (5.2.5)-(5.2.6)-(5.2.7), the DN algorithm for the coupled PDEs converges linearly for $0 < \theta_y, \theta_p < 1$, $\theta_y \neq 1/2, \theta_p \neq 1/2$. For $\theta_y, \theta_p = 1/2$, it converges in two iterations. Convergence is independent of the value of λ.*

Proof. For $\alpha = 1/2$, the updating terms (5.2.14) reduce to

$$\begin{aligned} h_y^k &= (1 - 2\theta_y)\, h_y^{k-1}, \\ h_p^k &= (1 - 2\theta_p)\, h_p^{k-1}, \end{aligned}$$

since $v(\alpha, \mu) = -1$ and $w(\alpha, \mu) = 0$ for $\alpha = 1/2$. Therefore the convergence is linear for $0 < \theta_y, \theta_p < 1$, $\theta_y \neq 1/2, \theta_p \neq 1/2$. If $\theta_y, \theta_p = 1/2$, we have $h_y^1 = 0 = h_p^1$, and hence the desired convergence occurs after one more iteration. □

We now focus on the more interesting case, where we consider asymmetric subdomains, i.e., $\alpha \neq 1/2$.

Theorem 33 (Convergence in the asymmetric case)**.** *Suppose $\alpha \neq 1/2$. Then the DN algorithm (5.2.5)-(5.2.6)-(5.2.7) for the coupled PDEs converges in at most three iterations if one chooses (θ_y, θ_p) between any one of the pairs*

$$\left(\Lambda^+, \Lambda^-\right), \left(\Lambda^-, \Lambda^+\right),$$

where $\Lambda^+ := \frac{1}{(1-v)} + \frac{|w|}{(1-v)\sqrt{(1-v)^2+w^2}}, \Lambda^- := \frac{1}{(1-v)} - \frac{|w|}{(1-v)\sqrt{(1-v)^2+w^2}}$.

Proof. For $\alpha \neq 1/2$, we set $\bar{h}_p^k := \mu h_p^k, \bar{h}_y^k := \frac{h_y^k}{\mu}$. Then we can write the updating terms (5.2.14) in the matrix form

$$\begin{pmatrix} \bar{h}_y^k \\ \bar{h}_p^k \end{pmatrix} = \left[\begin{pmatrix} 1-\theta_y & 0 \\ 0 & 1-\theta_p \end{pmatrix} + \begin{pmatrix} \theta_y v(\alpha,\mu) & -\theta_y w(\alpha,\mu) \\ \theta_p w(\alpha,\mu) & \theta_p v(\alpha,\mu) \end{pmatrix} \right] \begin{pmatrix} \bar{h}_y^{k-1} \\ \bar{h}_p^{k-1} \end{pmatrix}.$$

Note that the above matrix on the right (which we call S) is never zero for any particular set of values θ_y, θ_p. So we can not get two step convergence for $\alpha \neq 1/2$. But if for some positive integer n, $S^n = 0$, then we can write

$$\begin{pmatrix} \bar{h}_y^n \\ \bar{h}_p^n \end{pmatrix} = S^n \begin{pmatrix} \bar{h}_y^0 \\ \bar{h}_p^0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

so that the DN algorithm converges in $n+1$ iterations. Now the spectral radius of S is

$$\Upsilon\left(\theta_y, \theta_p, \alpha, \mu\right) := \max\left\{ \left| 1 - \frac{1}{2}\left(\theta_y + \theta_p\right)(1-v) \pm \frac{1}{2}\sqrt{\left(\theta_y - \theta_p\right)^2 (1-v)^2 - 4\theta_y\theta_p w^2} \right| \right\}.$$

For each $\alpha \in (0,1)$ and $\mu > 0$, we now solve for θ_y, θ_p the system

$$1 - \frac{1}{2}\left(\theta_y + \theta_p\right)(1-v) = 0, \left(\theta_y - \theta_p\right)^2 (1-v)^2 - 4\theta_y\theta_p w^2 = 0$$

simultaneously to obtain (Λ^+, Λ^-), where

$$\Lambda^+ = \frac{1}{(1-v)} + \frac{|\, w \,|}{(1-v)\sqrt{(1-v)^2 + w^2}}, \Lambda^- = \frac{2}{(1-v)} - \Lambda^+.$$

But Υ is symmetric with respect to θ_y and θ_p, so (Λ^-, Λ^+) is also a solution of the system of equations. Therefore $\Upsilon\left(\Lambda^\pm, \Lambda^\mp, \alpha, \mu\right) = 0$, resulting in $S^2 = 0$ and hence three step convergence to the exact solution. □

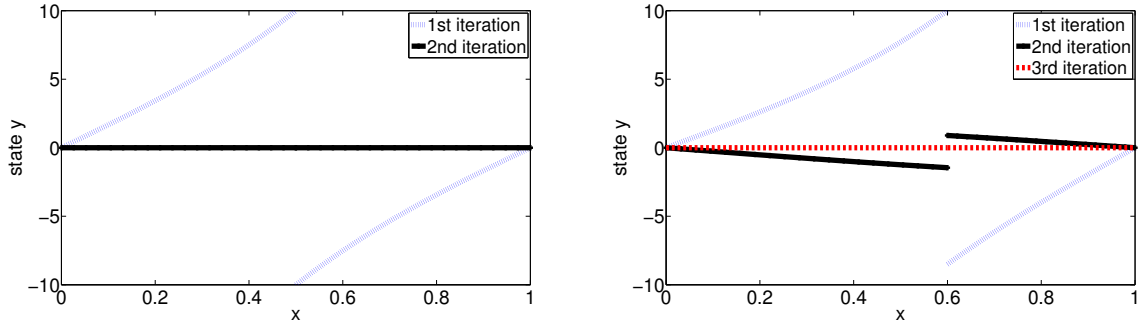


Figure 5.2.1: Convergence of the iterative solution of the DN method: in two iterations for the symmetric case on the left, and in three iterations for $\alpha = 0.6$ on the right

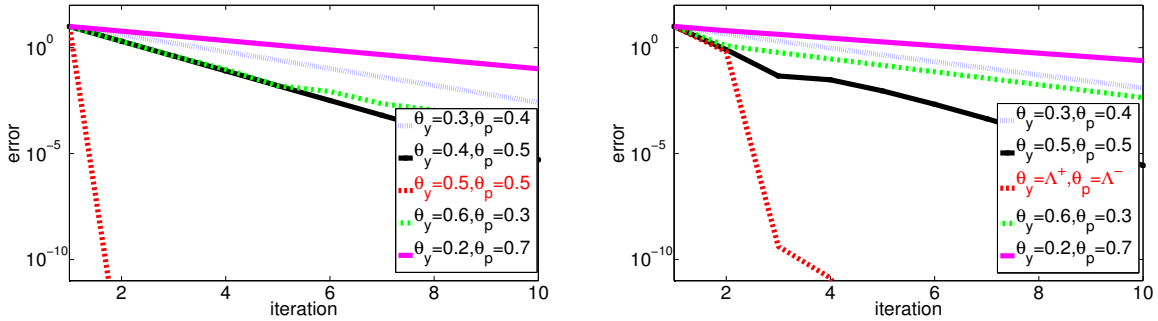


Figure 5.2.2: Comparison of convergence curves for the DN method with various values of θ_y, θ_p for $\alpha = 1/2$ on the left, and for $\alpha = 0.6$ on the right

Remark 34. As $v(\alpha, \mu) \leq 0$ (since $\rho_1\rho_2\eta_1\eta_2 + \sigma_1\sigma_2\gamma_1\gamma_2 \geq 0$, which is due to the facts: $\gamma_i \geq |\eta_i|, \sigma_i \geq |\rho_i|$ for all α, μ) and $w \in \mathbb{R}$, we have $\Lambda^- \in (0, 1)$ and apparently $\Lambda^+ \in (0, 2)$. But $\Lambda^+ \geq 1$ if and only if

$$v^2\left(1 + \frac{(1-v)^2}{w^2}\right) < 1,$$

which is not true for these particular v, w. So, we have both $\Lambda^+, \Lambda^- \in (0, 1)$.

5.2.1.2 Numerical illustration

We perform numerical experiments to measure the actual convergence rate of the DN algorithm for the model problem (5.2.1)-(5.2.2) with $\lambda = 1/2, \hat{y}(x) = 0$. For the DN algorithm to start, we take two initial guesses $h_y^0 = 10, h_p^0 = 20$. We consider two cases: first the symmetric case, $\alpha = 1/2$, and then the asymmetric case, $\alpha = 0.6$, for which the theoretical optimal values of the parameters for the DN method from Theorem 33 are $\Lambda^+ = 0.62, \Lambda^- = 0.57$. In Figure 5.2.1 we show the two step convergence of the DN method for $\alpha = 1/2$ on the left, and the three step convergence for $\alpha = 0.6$ on the right. The plots in Figure 5.2.2 represent the convergence behavior of the DN algorithm for different choices of parameters θ_y, θ_p. On the left panel we get $\theta_y = \theta_p = 1/2$ to be the

best parameters for the symmetric case, whereas on the right (Λ^+, Λ^-) produces the best performance for $\alpha = 0.6$ as expected.

5.2.2 Neumann-Neumann algorithm

To write the NN algorithm for both state and adjoint equations (5.2.2) and (5.2.3), we again subdivide $\Omega = (0,1)$ into two non-overlapping subdomains, $\Omega_1 = (0,\alpha)$ and $\Omega_2 = (\alpha,1)$. Then given two initial guesses g_y^0 and g_p^0 at the interface $\{x = \alpha\}$, the Neumann-Neumann algorithm is: for $k = 1, 2, \ldots$ compute the approximations

$$\begin{array}{rcll} -\nabla\cdot\left(\kappa(x)\nabla y_1^k\right) & = & u_1^k, & \text{in } \Omega_1, \\ y_1^k(0) & = & 0, & \\ y_1^k(\alpha) & = & g_y^{k-1}, & \end{array} \qquad \begin{array}{rcll} -\nabla\cdot\left(\kappa(x)\nabla y_2^k\right) & = & u_2^k, & \text{in } \Omega_2, \\ y_2^k(\alpha) & = & g_y^{k-1}, & \\ y_2^k(1) & = & 0, & \end{array} \tag{5.2.15}$$

and the correction step

$$\begin{array}{rl} -\nabla\cdot\left(\kappa(x)\nabla\psi_1^k\right)=0, & \text{in } \Omega_1, \\ \psi_1^k(0)=0, & \\ \partial_x\psi_1^k(\alpha)=\partial_x y_1^k(\alpha) - \partial_x y_2^k(\alpha), & \end{array} \qquad \begin{array}{rl} -\nabla\cdot\left(\kappa(x)\nabla\psi_2^k\right)=0, & \text{in } \Omega_2, \\ \partial_x\psi_2^k(\alpha)=\partial_x y_1^k(\alpha) - \partial_x y_2^k(\alpha), & \\ \psi_2^k(1)=0, & \end{array} \tag{5.2.16}$$

and similarly for the adjoint equation

$$\begin{array}{rcll} -\nabla\cdot\left(\kappa(x)\nabla p_1^k\right) & = & y_1^k, & \text{in } \Omega_1, \\ p_1^k(0) & = & 0, & \\ p_1^k(\alpha) & = & g_p^{k-1}, & \end{array} \qquad \begin{array}{rcll} -\nabla\cdot\left(\kappa(x)\nabla p_2^k\right) & = & y_2^k, & \text{in } \Omega_2, \\ p_2^k(\alpha) & = & g_p^{k-1}, & \\ p_2^k(1) & = & 0, & \end{array} \tag{5.2.17}$$

$$\begin{array}{rl} -\nabla\cdot\left(\kappa(x)\nabla\varphi_1^k\right)=0, & \text{in } \Omega_1, \\ \varphi_1^k(0)=0, & \\ \partial_x\varphi_1^k(\alpha)=\partial_x p_1^k(\alpha) - \partial_x p_2^k(\alpha), & \end{array} \qquad \begin{array}{rl} -\nabla\cdot\left(\kappa(x)\nabla\varphi_2^k\right)=0, & \text{in } \Omega_2, \\ \partial_x\varphi_2^k(\alpha)=\partial_x p_1^k(\alpha) - \partial_x p_2^k(\alpha), & \\ \varphi_2^k(1)=0, & \end{array} \tag{5.2.18}$$

with the update conditions

$$\begin{array}{rcl} g_y^k & = & g_y^{k-1} - \theta_y\left(\psi_1^k(\alpha) - \psi_2^k(\alpha)\right), \\ g_p^k & = & g_p^{k-1} - \theta_p\left(\varphi_1^k(\alpha) - \varphi_2^k(\alpha)\right), \end{array} \tag{5.2.19}$$

where $\theta_y, \theta_p \in (0,1)$ are relaxation parameters.

5.2.2.1 Convergence analysis

We present a convergence analysis for the NN algorithm (5.2.15)-(5.2.16)-(5.2.17)-(5.2.18)-(5.2.19) again for the special case $\kappa(x) = 1$. By (5.2.4), we have $u_i^k = -p_i^k/\lambda$ for $i = 1, 2$. Eliminating p_1^k, p_2^k from (5.2.15)-(5.2.17), we obtain

$$\begin{cases} D^{(4)}y_1^k + \frac{1}{\lambda}y_1^k = 0, \\ y_1^k(0) = 0, y_1^k(\alpha) = g_y^{k-1}, \\ D^{(2)}y_1^k(0) = 0, D^{(2)}y_1^k(\alpha) = g_p^{k-1}/\lambda, \end{cases} \qquad \begin{cases} D^{(4)}y_2^k + \frac{1}{\lambda}y_2^k = 0, \\ y_2^k(1) = 0, y_2^k(\alpha) = g_y^{k-1}, \\ D^{(2)}y_2^k(1) = 0, D^{(2)}y_2^k(\alpha) = g_p^{k-1}/\lambda. \end{cases} \tag{5.2.20}$$

Set again $\mu^4 := 1/\lambda$. Then the solutions of (5.2.20) are

$$y_1^k(x) = E_1 \sinh\left(\mu x/\sqrt{2}\right)\cos\left(\mu x/\sqrt{2}\right) + E_2\cosh\left(\mu x/\sqrt{2}\right)\sin\left(\mu x/\sqrt{2}\right),$$

with

$$E_1 = \frac{g_y^{k-1}\sigma_1\eta_1 - \mu^2 g_p^{k-1}\gamma_1\rho_1}{\sigma_1^2+\rho_1^2}, \qquad E_2 = \frac{g_y^{k-1}\gamma_1\rho_1 + \mu^2 g_p^{k-1}\sigma_1\eta_1}{\sigma_1^2+\rho_1^2},$$

and

$$y_2^k(x) = F_1 \sinh\left(\frac{\mu(1-x)}{\sqrt{2}}\right)\cos\left(\frac{\mu(1-x)}{\sqrt{2}}\right) + F_2\cosh\left(\frac{\mu(1-x)}{\sqrt{2}}\right)\sin\left(\frac{\mu(1-x)}{\sqrt{2}}\right),$$

with

$$F_1 = \frac{g_y^{k-1}\sigma_2\eta_2 - \mu^2 g_p^{k-1}\gamma_2\rho_2}{\sigma_2^2+\rho_2^2}, \qquad F_2 = \frac{g_y^{k-1}\gamma_2\rho_2 + \mu^2 g_p^{k-1}\sigma_2\eta_2}{\sigma_2^2+\rho_2^2}.$$

Finally solving ψ_i^k, φ_i^k in (5.2.16)-(5.2.18) and replacing them in (5.2.19) we get

$$g_y^k = g_y^{k-1} - \theta_y\frac{\mu}{\sqrt{2}}\left\{E_1\left(\gamma_1\eta_1 - \sigma_1\rho_1\right) + E_2\left(\sigma_1\rho_1 + \gamma_1\eta_1\right) + F_1\left(\gamma_2\eta_2 - \sigma_2\rho_2\right) + F_2\left(\sigma_2\rho_2 + \gamma_2\eta_2\right)\right\},$$

$$g_p^k = g_p^{k-1} - \frac{\theta_p}{\mu\sqrt{2}}\left\{E_2\left(\gamma_1\eta_1 - \sigma_1\rho_1\right) - E_1\left(\sigma_1\rho_1 + \gamma_1\eta_1\right) + F_2\left(\gamma_2\eta_2 - \sigma_2\rho_2\right) - F_1\left(\sigma_2\rho_2 + \gamma_2\eta_2\right)\right\},$$

from where more simplifications give

$$\begin{aligned} g_y^k &= g_y^{k-1} - \theta_y\left(g_y^{k-1}z_1 + \mu^2 g_p^{k-1}z_2\right), \\ g_p^k &= g_p^{k-1} - \theta_p\left(g_p^{k-1}z_1 - \tfrac{1}{\mu^2}g_y^{k-1}z_2\right), \end{aligned} \tag{5.2.21}$$

where

$$z_1(\alpha,\mu) = \frac{\mu}{\sqrt{2}}\left(\frac{\sigma_1\gamma_1+\rho_1\eta_1}{\sigma_1^2+\rho_1^2} + \frac{\sigma_2\gamma_2+\rho_2\eta_2}{\sigma_2^2+\rho_2^2}\right), z_2(\alpha,\mu) = \frac{\mu}{\sqrt{2}}\left(\frac{\sigma_1\gamma_1-\rho_1\eta_1}{\sigma_1^2+\rho_1^2} + \frac{\sigma_2\gamma_2-\rho_2\eta_2}{\sigma_2^2+\rho_2^2}\right).$$

Theorem 35 (Convergence of the NN algorithm)**.** *The NN algorithm for the coupled PDEs* (5.2.15)-(5.2.16)-(5.2.17)-(5.2.18)-(5.2.19) *converges in at most three iterations if one chooses* (θ_y, θ_p) *between any one of the pairs*

$$\left(\Theta^+, \Theta^-\right), \left(\Theta^-, \Theta^+\right),$$

where $\Theta^+ := \frac{1}{z_1} + \frac{|z_2|}{z_1\sqrt{z_1^2+z_2^2}}, \Theta^- := \frac{1}{z_1} - \frac{|z_2|}{z_1\sqrt{z_1^2+z_2^2}}$.

Proof. Setting $\bar{g}_p^k := \mu g_p^k, \bar{g}_y^k := \frac{g_y^k}{\mu}$, we write the updating terms (5.2.21) in the matrix form

$$\begin{pmatrix} \bar{g}_y^k \\ \bar{g}_p^k \end{pmatrix} = \begin{pmatrix} 1-\theta_y z_1 & -\theta_y z_2 \\ \theta_p z_2 & 1-\theta_p z_1 \end{pmatrix}\begin{pmatrix} \bar{g}_y^{k-1} \\ \bar{g}_p^{k-1} \end{pmatrix}.$$

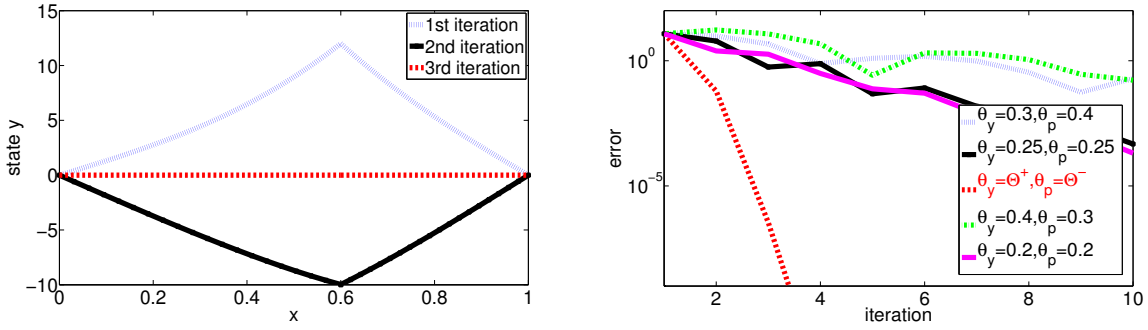


Figure 5.2.3: Convergence of NN: convergence of the iterative solutions with optimal parameters in three iterations on the left, and convergence for various values of θ_y, θ_p on the right

Now the matrix on the right (which we call P) is never zero for any particular set of values θ_y, θ_p. But like in the DN method, if we see $P^n = 0$, for some positive integer n, then we can write

$$\begin{pmatrix} \bar{g}_y^n \\ \bar{g}_p^n \end{pmatrix} = P^n \begin{pmatrix} \bar{g}_y^0 \\ \bar{g}_p^0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

resulting in convergence in $n+1$ iterations. We calculate the spectral radius of P as $\Phi(\theta_y, \theta_p, \alpha, \mu) := \max\left\{\left|1 - \frac{1}{2}(\theta_y + \theta_p) z_1 \pm \frac{1}{2}\sqrt{(\theta_y - \theta_p)^2 z_1^2 - 4\theta_y\theta_p z_2^2}\right|\right\}$. We now solve the system

$$1 - \frac{1}{2}(\theta_y + \theta_p) z_1 = 0, (\theta_y - \theta_p)^2 z_1^2 - 4\theta_y\theta_p z_2^2 = 0$$

simultaneously for each $\alpha \in (0,1)$ and $\mu > 0$ to get a solution (Θ^+, Θ^-), where

$$\Theta^+ := \frac{1}{z_1} + \frac{|z_2|}{z_1\sqrt{z_1^2 + z_2^2}}, \quad \Theta^- := \frac{2}{z_1} - \Theta^+.$$

Again due to the symmetric nature of Φ, (Θ^-, Θ^+) is another solution pair of the system of equations. Thus $\Phi(\Theta^\pm, \Theta^\mp, \alpha, \mu) = 0$, resulting in $P^2 = 0$ and therefore three step convergence to the exact solution. This completes the result. □

Remark 36. Note that, even for the symmetric case, $\alpha = 1/2$, we do not get two step convergence for any relaxation parameters θ_y, θ_p.

5.2.2.2 Numerical illustration

We test our theoretical results for the NN method by numerical experiments for the model problem (5.2.1)-(5.2.2) with $\lambda = 1/2^4, \hat{y}(x) = 0$. We take two initial guesses as $g_y^0 = 12, g_p^0 = 4$. We consider the asymmetric case, $\alpha = 0.6$, for which the theoretical optimal values of the parameters from Theorem 35 are $\Theta^+ = 0.30, \Theta^- = 0.16$. In Figure 5.2.3 we plot on the left panel the convergence of the iterative solution of the state equation, and on the right the convergence curves for various values of the parameters θ_y, θ_p.

5.3 Evolution problems

Next we formulate the DNWR and NNWR algorithms for linear-quadratic parabolic optimal control problems, originating from the examples of transient optimal heating with distributed control, or identification of a source of pollution in a waterbody; see Chapter 1. To illustrate our ideas, we consider a simple model problem on a bounded spatial domain $\Omega \subset \mathbb{R}$ with an objective functional to be minimized

$$J(y,u) = \frac{1}{2}\int_0^T \int_\Omega (y(x,t) - \hat{y}(x,t))^2 \, dxdt + \frac{\lambda}{2}\int_0^T \int_\Omega u^2(x,t) dxdt, \tag{5.3.1}$$

subject to the parabolic PDE with distributed control

$$\begin{cases} \partial_t y(x,t) - \nu\Delta y(x,t) = u(x,t), & x \in \Omega, t \in (0,T), \\ y(x,0) = y_0(x), & x \in \Omega, \\ y(x,t) = 0, & x \in \partial\Omega \times (0,T). \end{cases} \tag{5.3.2}$$

We take $\Omega = (0,1)$ and $U_{\text{ad}} = \mathbb{R}$. For more such space-time problems, see [1, 56].

Consider the Lagrangian,

$$L(y,u,p) = J(y,u) + \int_0^T \int_0^1 p(x,t)\left(u(x,t) + \nu\Delta y(x,t) - \partial_t y(x,t)\right) dxdt,$$

with p being the Lagrange multiplier or the adjoint variable. From the conditions (5.1.4), the adjoint equation corresponding to the problem (5.3.1)-(5.3.2) becomes

$$\begin{cases} -\partial_t p(x,t) - \nu\Delta p(x,t) = y(x,t) - \hat{y}(x,t), & x \in \Omega, t \in (0,T), \\ p(x,T) = 0, & x \in \Omega, \\ p(x,t) = 0, & x \in \partial\Omega \times (0,T), \end{cases} \tag{5.3.3}$$

with the optimality condition

$$p(x,t) + \lambda u(x,t) = 0. \tag{5.3.4}$$

Remark 37. It is well known that the backward heat equation is ill-posed, but the adjoint equation (5.3.3) is a terminal value problem, and thus not a backward heat equation. The problem is thus well-posed, which can be seen as follows: set $U(x,t) = p(x,T-t)$. Then U satisfies the following initial-boundary value equation:

$$\partial_t U - \nu\Delta U = y(x,T-t) - \hat{y}(x,T-t) \quad \text{in } \Omega, \;\; U(x,0) = 0 \quad \text{in } \Omega, \;\; U(x,t) = 0 \quad \text{on } \partial\Omega,$$

for $t \in (0,T)$, and it is a well-posed problem. The above form is also more suitable for numerical implementation.

Now we apply the DNWR and the NNWR methods to solve the underlying space-time PDEs (5.3.2) and (5.3.3). We only consider the error equations in the rest of our discussion; so we take $\hat{y}(x,t) = 0$ and $y_0(x) = 0$ in (5.3.1)-(5.3.2)-(5.3.3)-(5.3.4).

5.3.1 DNWR and NNWR algorithms

To explain the new algorithms, we assume for simplicity that the spatial domain $\Omega = (0,1)$ is partitioned into two non-overlapping subdomains $\Omega_1 = (0,\alpha)$ and $\Omega_2 = (\alpha,1)$. We denote by y_i, p_i, u_i the restriction of y, p, u to Ω_i, $i = 1,2$.

The DNWR for the coupled PDEs consists of the following steps: given two initial guesses $w_y^0(t)$ and $w_p^0(t)$ along the interface $\{x=\alpha\}\times(0,T)$, compute for $k = 1, 2, ...$

$$\begin{array}{rlll} \partial_t y_1^k - \nu\Delta y_1^k = u_1^k, & \text{in } \Omega_1, & -\partial_t p_1^k - \nu\Delta p_1^k = y_1^k, & \text{in } \Omega_1, \\ y_1^k(x,0) = 0, & \text{in } \Omega_1, & p_1^k(x,T) = 0, & \text{in } \Omega_1, \\ y_1^k(0,t) = 0, & t\in(0,T), & p_1^k(0,t) = 0, & t\in(0,T), \\ y_1^k(\alpha,t) = w_y^{k-1}(t), & t\in(0,T), & p_1^k(\alpha,t) = w_p^{k-1}(t), & t\in(0,T), \end{array} \tag{5.3.5}$$

and

$$\begin{array}{rlll} \partial_t y_2^k - \nu\Delta y_2^k = u_2^k, & \text{in } \Omega_2, & -\partial_t p_2^k - \nu\Delta p_2^k = y_2^k, & \text{in } \Omega_1, \\ y_2^k(x,0) = 0, & \text{in } \Omega_2, & p_2^k(x,T) = 0, & \text{in } \Omega_1, \\ \partial_x y_2^k(\alpha,t) = \partial_x y_1^k(\alpha,t), & t\in(0,T), & \partial_x p_2^k(\alpha,t) = \partial_x p_1^k(\alpha,t), & t\in(0,T), \\ y_1^k(1,t) = 0, & t\in(0,T), & p_2^k(1,t) = 0, & t\in(0,T), \end{array} \tag{5.3.6}$$

together with the update conditions

$$\begin{array}{rcl} w_y^k(t) & = & \theta_y y_2^k(\alpha,t) + (1-\theta_y) w_y^{k-1}(t), \\ w_p(t) & = & \theta_p p_2^k(\alpha,t) + (1-\theta_p) w_p^{k-1}(t), \end{array} \tag{5.3.7}$$

where $\theta_y, \theta_p \in (0,1)$ are two relaxation parameters.

The NNWR starts with two initial guesses $\vartheta_y^0(t)$ and $\vartheta_p^0(t)$ along the interface $\{x=\alpha\}\times(0,T)$ and then computes simultaneously in $t\in(0,T)$ for $i = 1,2$ with $k = 1, 2, ...$

$$\begin{array}{rlll} \partial_t y_1^k - \nu\Delta y_1^k = u_1^k, & \text{in } \Omega_1, & \partial_t y_2^k - \nu\Delta y_2^k = u_2^k, & \text{in } \Omega_2, \\ y_1^k(x,0) = 0, & \text{in } \Omega_1, & y_2^k(x,0) = 0, & \text{in } \Omega_2, \\ y_1^k(0,t) = 0, & & y_2^k(\alpha,t) = \vartheta_y^{k-1}(t), & \\ y_1^k(\alpha,t) = \vartheta_y^{k-1}(t), & & y_2^k(1,t) = 0, & \end{array}$$

$$\begin{array}{rlll} \partial_t \psi_1^k - \nu\Delta\psi_1^k = 0, & \text{in } \Omega_1, & \partial_t \psi_2^k - \nu\Delta\psi_2^k = 0, & \text{in } \Omega_2, \\ \psi_1^k(x,0) = 0, & \text{in } \Omega_1, & \psi_2^k(x,0) = 0, & \text{in } \Omega_2, \\ \psi_1^k(0,t) = 0, & & \partial_x\psi_2^k(\alpha,t) = \left(\partial_x y_1^k - \partial_x y_2^k\right)\big|_{x=\alpha}, & \\ \partial_x\psi_1^k(\alpha,t) = \left(\partial_x y_1^k - \partial_x y_2^k\right)\big|_{x=\alpha}, & & \psi_2^k(1,t) = 0, & \end{array}$$

and for the adjoint equation

$$\begin{array}{rlll} -\partial_t p_1^k - \nu\Delta p_1^k = y_1^k, & \text{in } \Omega_1, & -\partial_t p_2^k - \nu\Delta p_2^k = y_2^k, & \text{in } \Omega_2, \\ p_1^k(x,T) = 0, & \text{in } \Omega_1, & p_2^k(x,T) = 0, & \text{in } \Omega_2, \\ p_1^k(0,t) = 0, & & p_2^k(\alpha,t) = \vartheta_p^{k-1}(t), & \\ p_1^k(\alpha,t) = \vartheta_p^{k-1}(t), & & p_2^k(1,t) = 0, & \end{array}$$

$$\begin{array}{rlll} -\partial_t \varphi_1^k - \nu\Delta\varphi_1^k = 0, & \text{in } \Omega_1, & -\partial_t \varphi_2^k - \nu\Delta\varphi_2^k = 0, & \text{in } \Omega_2, \\ \varphi_1^k(x,T) = 0, & \text{in } \Omega_1, & \varphi_2^k(x,T) = 0, & \text{in } \Omega_2, \\ \varphi_1^k(0,t) = 0, & & \partial_x\varphi_2^k(\alpha,t) = \left(\partial_x p_1^k - \partial_x p_2^k\right)\big|_{x=\alpha}, & \\ \partial_x\varphi_1^k(\alpha,t) = \left(\partial_x p_1^k - \partial_x p_2^k\right)\big|_{x=\alpha}, & & \varphi_2^k(1,t) = 0, & \end{array}$$

with the update conditions

$$\begin{aligned} \vartheta_y^k(t) &= \vartheta_y^{k-1}(t) - \theta_y \left(\psi_1^k(\alpha,t) - \psi_2^k(\alpha,t)\right), \\ \vartheta_p^k(t) &= \vartheta_p^{k-1}(t) - \theta_p \left(\varphi_1^k(\alpha,t) - \varphi_2^k(\alpha,t)\right), \end{aligned}$$

where $\theta_y, \theta_p \in (0,1)$ are relaxation parameters.

5.3.1.1 Calculation for convergence analysis

To analyze the convergence behavior of the DNWR method (5.3.5)-(5.3.6)-(5.3.7) (similarly for the NNWR method), we need to solve the coupled problems (5.3.5) and (5.3.6). But from (5.3.4), we have $u_i^k = -p_i^k/\lambda$ for $i = 1, 2$. So eliminating p_1^k from (5.3.5), we get

$$\nu^2 \frac{\partial^4 y_1^k}{\partial x^4} - \frac{\partial^2 y_1^k}{\partial t^2} + \frac{1}{\lambda} y_1^k = 0, \tag{5.3.8}$$

with the initial conditions

$$(i)\ y_1^k(x,0) = 0, \quad (ii)\ \left(\partial_t y_1^k - \nu \partial_{xx} y_1^k\right)\big|_{(x,T)} = 0,$$

and the boundary conditions

$$\begin{aligned} &(a) \quad y_1^k(0,t) = 0, && (b) \quad \left(\partial_t y_1^k - \nu \partial_{xx} y_1^k\right)\big|_{(0,t)} = 0, \\ &(c) \quad y_1^k(\alpha,t) = w_y^{k-1}(t), && (d) \quad \left(\partial_t y_1^k - \nu \partial_{xx} y_1^k\right)\big|_{(\alpha,t)} = -w_p^{k-1}/\lambda. \end{aligned}$$

We now use separation of variables to solve (5.3.8). Dropping the iteration number in calculation, let us suppose $y_1(x,t) = H(x)V(t)$. Therefore substituting y_1 back into equation (5.3.8), we obtain

$$\frac{H^{(4)}}{H} + \frac{1}{\lambda\nu^2} = \frac{1}{\nu^2}\frac{V^{(2)}}{V}, \tag{5.3.9}$$

with the initial and boundary conditions (considering only non-trivial solutions)

$$\text{IC}: \ (i)\ V(0) = 0, \quad (ii)\ H(x)V^{(1)}(T) = \nu H^{(2)}(x)V(T),\ \forall x \in \Omega_1$$

$$\begin{aligned} \text{BC}: \quad &(iii)\ H(0) = 0, && (iv) \quad H^{(2)}(0) = 0, \\ &(v)\ H(\alpha)V(t) = w_y(t), && (vi) \quad H(\alpha)V^{(1)}(t) - \nu H^{(2)}(\alpha)V(t) = -w_p(t)/\lambda. \end{aligned}$$

Note that, the symbol $f^{(m)}$ denotes the mth derivative of the function f with respect to the corresponding variable. One can have the following possibilities:

CASE 1: If both sides of (5.3.9) are equal to a constant $-\kappa^2$, then we get a solution set as:

$$\begin{aligned} H(x) &= A_1 \sinh\left(\omega_1 x/\sqrt{2}\right)\cos\left(\omega_1 x/\sqrt{2}\right) + B_1 \cosh\left(\omega_1 x/\sqrt{2}\right)\sin\left(\omega_1 x/\sqrt{2}\right), \\ V(t) &= C_1 \sin\left(\nu\kappa t\right), \end{aligned}$$

together with $\omega_1^4 = \kappa^2 + \frac{1}{\lambda\nu^2}$ and the conditions $(ii), (v), (vi)$. Now the initial condition (ii) yields

$$\frac{H^{(2)}(x)}{H(x)} = \frac{1}{\nu}\frac{V^{(1)}(T)}{V(T)} =: \beta_1,$$

and so one has $\beta_1 = \kappa \cot(\kappa\nu T)$ and $H^{(2)}(x) - \beta_1 H(x) = 0$. Therefore, the solution of (5.3.8) would be of the form

$$y_1(x,t) = \sum_{n\geq 1} \sin(\nu\kappa_n t) \Big(A_n \sinh(\omega_{1,n} x/\sqrt{2}) \cos(\omega_{1,n} x/\sqrt{2}) + B_n \cosh(\omega_{1,n} x/\sqrt{2}) \sin(\omega_{1,n} x/\sqrt{2}) \Big).$$

CASE 2: If the ratio in (5.3.9) is equal to zero, then we obtain a solution set as:

$$\begin{aligned} H(x) &= A_2 \sinh\left(\omega_2 x/\sqrt{2}\right) \cos\left(\omega_2 x/\sqrt{2}\right) + B_2 \cosh\left(\omega_2 x/\sqrt{2}\right) \sin\left(\omega_2 x/\sqrt{2}\right), \\ V(t) &= C_2 t, \end{aligned}$$

together with $\omega_2^4 = \frac{1}{\lambda\nu^2}$ and the conditions $(ii), (v), (vi)$. Also, the condition (ii) gives

$$\frac{H^{(2)}(x)}{H(x)} = \frac{1}{\nu} \frac{V^{(1)}(T)}{V(T)} =: \beta_2,$$

and so we have $\beta_2 = \frac{1}{\nu T}$ and $H^{(2)}(x) - \beta_2 H(x) = 0$. Therefore, the solution of (5.3.8) would be of the form

$$y_1(x,t) = \sum_{n\geq 1} t \Big(A_n \sinh(\omega_{2,n} x/\sqrt{2}) \cos(\omega_{2,n} x/\sqrt{2}) + B_n \cosh(\omega_{2,n} x/\sqrt{2}) \sin(\omega_{2,n} x/\sqrt{2}) \Big).$$

CASE 3: If the ratio in (5.3.9) is equal to ϖ^2, then we get a solution set as:

$$\begin{aligned} H(x) &= A_3 \sinh\left(\omega_3 x/\sqrt{2}\right) \cos\left(\omega_3 x/\sqrt{2}\right) + B_3 \cosh\left(\omega_3 x/\sqrt{2}\right) \sin\left(\omega_3 x/\sqrt{2}\right), \\ V(t) &= C_3 \sinh(\nu\varpi t), \end{aligned}$$

together with $\omega_3^4 = \frac{1}{\lambda\nu^2} - \varpi^2$ and the conditions $(ii), (v), (vi)$. In addition, we obtain from (ii)

$$\frac{H^{(2)}(x)}{H(x)} = \frac{1}{\nu} \frac{V^{(1)}(T)}{V(T)} =: \beta_3,$$

and so we get $\beta_3 = \varpi \coth(\varpi\nu T)$ and $H^{(2)}(x) - \beta_3 H(x) = 0$. Therefore, the solution of (5.3.8) would be of the form

$$y_1(x,t) = \sum_{n\geq 1} \sinh(\nu\varpi_n t) \Big(A_n \sinh(\omega_{3,n} x/\sqrt{2}) \cos(\omega_{3,n} x/\sqrt{2}) + B_n \cosh(\omega_{3,n} x/\sqrt{2}) \sin(\omega_{3,n} x/\sqrt{2}) \Big).$$

But it is difficult to get the closed form solutions to proceed further analysis. We keep this for future work.

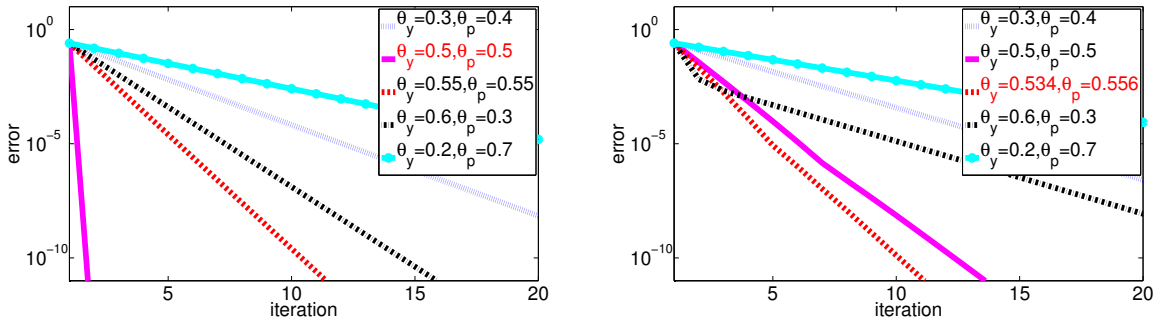


Figure 5.3.1: Convergence of DNWR with various values of θ_y, θ_p for $T = 2$ for the symmetric subdomains on the left, and $\alpha = 0.6$ on the right

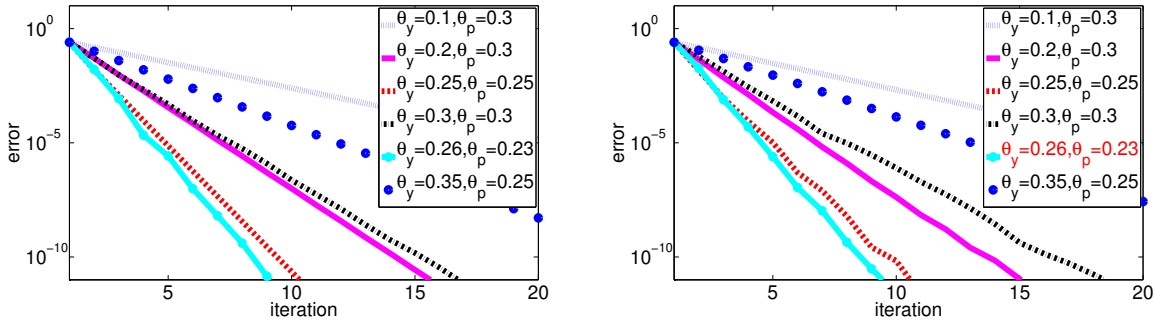


Figure 5.3.2: Convergence of NNWR with various values of θ_y, θ_p for $T = 2$ for the symmetric subdomains on the left, and $\alpha = 0.6$ on the right

5.3.1.2 Numerical illustration

We implement these two algorithms numerically for the model problem (5.3.1)-(5.3.2)-(5.3.3)-(5.3.4) with $\lambda = 1, \nu = 1, \hat{y}(x,t) = 0, y_0(x) = 0$, with a discretization using standard centered finite differences in space and backward Euler in time with $\Delta x = \Delta t = 4 \times 10^{-3}$. For the DNWR and the NNWR algorithms, we consider two cases: first the symmetric decomposition, where $\alpha = 1/2$, and then a domain decomposition with $\Omega_1 = (0, 0.6), \Omega_2 = (0.6, 1)$. In all our numerical experiments, we choose $w_y^0(t) = w_p^0(t) = t^2, t \in (0, T]$ as the initial guesses. Figure 5.3.1 gives the numerically measured convergence curves of the DNWR method for $T = 2$ and for different values of the parameters θ_y, θ_p for $\alpha = 1/2$ on the left, and $\alpha = 0.6$ on the right. We see that for the symmetric case, we get linear convergence for all relaxation parameters θ_y, θ_p, except for $(\theta_y, \theta_p) = (0.5, 0.5)$, when we observe two step convergence. But for the asymmetric decomposition we can not predict numerically the best value of the parameters. In Figure 5.3.2 we plot the convergence curves for $T = 2$ for the NNWR algorithm for $\alpha = 1/2$ on the left, and $\alpha = 0.6$ on the right. Unlike for the DNWR, we do not see any difference between the symmetric and asymmetric case.

5.4 Conclusion

We have applied substructuring methods to solve the state and adjoint PDEs in an elliptic optimal control problem, and then analyzed for a model problem to produce appropriate relaxation parameters which give at most three-step convergence to the exact solution for both DN and NN algorithms. We also formulate the DNWR and the NNWR algorithms to solve the underlying coupled forward and backward space-time problems in parabolic optimal control problems. With numerical experiments, we showed two-step convergence, choosing a particular set of relaxation parameters for the ideal symmetric case in DNWR. For the NNWR method, we also see finite step convergence, although we do not get two-step convergence even in symmetric case.

Conclusion and Future work

We introduced two new classes of space-time parallel methods, the Dirichlet-Neumann waveform relaxation (DNWR) and the Neumann-Neumann waveform relaxation (NNWR) algorithms for general parabolic as well as hyperbolic problems. For the one-dimensional heat equation, we proved superlinear convergence for both algorithms for a particular choice of the relaxation parameter. For the NNWR, our convergence estimate holds for a decomposition into many subdomains, and we also gave an extension to two spatial dimensions. We have also presented a detailed analysis of their convergence properties for the 1D second order wave equation. We showed that for a particular choice of the relaxation parameter, convergence can be achieved in a finite number of steps. Moreover, choosing the time window lengh carefully, these algorithms can be used to solve wave propagation problems in two iterations only. We have observed with numerical experiments that the DNWR and NNWR algorithms converge faster than the Schwarz WR iterations. Another straightforward advantage of the DNWR or the NNWR algorithm over SWR algorithm is that one needs not worry about choosing the appropriate transmission condition in order to obtain fast convergence.

The first question to arise here is whether it is possible to generalize the analysis of the new DNWR and NNWR methods for solving space-time problems in parallel, to more complex problems on irregular domains. We have shown with numerical experiments in Chapter 2 and Chapter 3 that, the DNWR and NNWR still lead to superlinear convergence for the heat equation with variational coefficient. But, our technique of analysis does not apply, and one should analyze the behavior of the algorithms, preferably with a variational framework, for problem with variational coefficient, and on subdomains having cross-points.

In this thesis we have only analyzed a special case, where we choose the ralaxation parameter $\theta = 1/2$ in DNWR, and $\theta = 1/4$ in NNWR. But a more detailed analysis would be required to analyze the behavior of the DNWR and the NNWR algorithms for other choices of the relaxation parameter θ. However, an ideal version of the DNWR algorithm for the model problem (2.3.1) on the spatial domain Ω, partitioned into two non-overlapping Ω_1 and Ω_2, would be as follows: given an initial guess $h^0(\boldsymbol{x}, t)$ along the interface $\Gamma \times (0, T)$, compute

$$
\begin{aligned}
\partial_t u_1^k - \nabla \cdot \big(\kappa(\boldsymbol{x},t)\nabla u_1^k\big) &= f, && \text{in } \Omega_1,\\
u_1^k(\boldsymbol{x},0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_1,\\
u_1^k &= g, && \text{on } \partial\Omega_1 \setminus \Gamma,\\
u_1^k &= h^{k-1}, && \text{on } \Gamma,
\end{aligned}
\qquad
\begin{aligned}
\partial_t u_2^k - \nabla \cdot \big(\kappa(\boldsymbol{x},t)\nabla u_2^k\big) &= f, && \text{in } \Omega_2,\\
u_2^k(\boldsymbol{x},0) &= u_0(\boldsymbol{x}), && \text{in } \Omega_2,\\
\partial_{\boldsymbol{n}_2} u_2^k &= -\partial_{\boldsymbol{n}_1} u_1^k, && \text{on } \Gamma,\\
u_2^k &= g, && \text{on } \partial\Omega_2 \setminus \Gamma,
\end{aligned}
\qquad (\star)
$$

with the update condition

$$h^k(\boldsymbol{x},t) = \theta(t)u_2^k\,\big|_{\Gamma\times(0,T)} + (1-\theta(t))\,h^{k-1}(\boldsymbol{x},t),$$

$\theta(t)$ being a relaxation parameter, that is given by a function $\theta : (0,T] \to (0,1]$. Clearly the above algorithm $(\star)$ reduces to the classical DNWR iteration (2.3.2)-(2.3.3) if $\theta(t)$ is chosen to be a constant. Similarly a new version for the NNWR (3.3.1)-(3.3.2)(3.3.3) can also be rewritten by choosing a time-dependent relaxation parameter $\theta(t)$. It would be very interesting to study the convergence behavior of these generalized algorithms.

Another idea is to use the DNWR or NNWR to solve scalar conservation laws numerically, and study the convergence behavior. As discussed earlier, the classical way of applying DD methods to space-time problems is to discretize uniformly in time first over the entire domain by an implicit scheme, and then to apply a suitable DD method to solve the sequence of steady problems at each time step. But due to the uniform nature of the time discretization, we cannot expect to have an optimal space-time discretization. So another way to consider the DD approach is to apply WR methods. This approach provides a nice framework for developing space-time adaptive schemes in which different time schemes and then steps are used in different subdomains. For a formal discussion on convergence analysis of SWR method for convection-dominated conservation laws, see [39]. While analyzing the DNWR or the NNWR for scalar conservation laws, the first goal would be getting continuous and discrete convergence estimates, so as to achieve at least the same rate of convergence as the classical methods, and then a systematic comparison in terms of performence with SWR methods.

I

Appendix A

We introduce the basic notions of the Laplace and the Fourier transform as two special cases of a general integral transform. We define all these mathematical operators for a function $u(x,t), (x,t) \in \Omega \times \mathcal{T} \subseteq \mathbb{R} \times \mathbb{R}$, so as to keep notational uniformity in the entire thesis.

I.1 Integral Transforms

Definition. An integral transform is an operator that maps a function u to another function $\mathcal{I}\{u\}$ as follows:

$$\mathcal{I}\{u\}(x,\xi) := \int_{t_1}^{t_2} K(t,\xi)u(x,t)\,dt,\ t_1, t_2 \in \overline{\mathcal{T}}.$$

The two-variables function K is called the kernel function or nucleus of the transform.

Each particular choice of K produces different useful integral transform. We discuss in this section two important examples, namely the Fourier transform and the Laplace transform. However, all integral transforms have the following linearity property:

$$\mathcal{I}\{c_1 u + c_2 v\} = c_1\mathcal{I}\{u\} + c_2\mathcal{I}\{v\}, \text{ for constants } c_1, c_2.$$

I.1.1 Laplace Transforms

Definition. We define the (unilateral) Laplace transform of a function $u(x,t)$ with respect to time t ($\mathcal{T} = (0,\infty)$) as

$$\hat{u}(x,s) := \mathcal{L}\{u(x,t)\} := \int_0^{\infty} e^{-st}u(x,t)\,dt,$$

where s is a complex variable.

Note that it is a particular integral transform where $K(t,s) := e^{-st}$. The hat symbol denotes the Laplace transform of a function in time.

Definition. The inverse Laplace transform of a function $\hat{u}(x,s)$ having no poles in the region $\{s : \mathrm{Re}(s) > \alpha\}$, is defined as follows:

$$u(x,t) := \mathcal{L}^{-1}\{\hat{u}(x,s)\} := \frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} e^{st}\hat{u}(x,s)\,ds.$$

In a more practical sense, if $\mathcal{L}\{u(x,t)\} = \hat{u}(x,s)$, then the inverse Laplace transform of $\hat{u}(x,s)$ is defined by

$$\mathcal{L}^{-1}\{\hat{u}(x,s)\} := u(x,t), \qquad t \geq 0,$$

and maps the Laplace transform of a function back to the original function. We state two results from [18] for the existence of a Laplace transform and an inverse Laplace transform respectively.

Theorem 38. *Suppose a function $u(x,t)$ is continuous or piecewise continuous in every finite interval in the range $t > 0$, and is $\mathcal{O}(e^{\alpha t})$ for each x, i.e.,*

$$|u(x,t)| \leq M(x)e^{\alpha t},\ t > T_0$$

for a bounded function M. Then the Laplace transform of $u(x,t)$ exists in the region $\{s : Re(s) > \alpha\}$.

Theorem 39. *Suppose $F(x,s)$ is any function of a real variable x and a complex variable s that is analytic, and for each x, $F(x,s)$ is $\mathcal{O}\left(s^{-k}\right)$ for all s in a half plane $\mathrm{Re}(s) \geq \alpha$, where $k > 1$. Also let $F(x, \mathrm{Re}(s))$ be real for $\mathrm{Re}(s) \geq \alpha$. Then the inverse Laplace transform of $F(x,s)$ exists in the region $\{s : \mathrm{Re}(s) > \alpha\}$, i.e., there exists a real valued function $u(x,t)$ such that*

$$\mathcal{L}\{u(x,t)\} = F(x,s),\ \mathrm{Re}(s) > \alpha.$$

Furthermore, $u(x,t)$ is $\mathcal{O}(e^{\alpha t})$, continuous for $t \in \mathbb{R}$, and $u(x,t) = 0$ for $t \leq 0$.

We now list some useful properties of the Laplace transform:

L1. $\mathcal{L}\left\{e^{\beta t}u(\cdot,t)\right\} = \hat{u}(\cdot, s-\beta)$ (*Frequency shifting*),

L2. $\mathcal{L}\{u(\cdot,t-\beta)H(t-\beta)\} = e^{-\beta s}\hat{u}(\cdot,s)$, where $H(t) := \begin{cases} 1, & t > 0, \\ 0, & t \leq 0. \end{cases}$ is the Heaviside step function (*Time shifting*),

L3. $\mathcal{L}\left\{u^{(n)}(\cdot,t)\right\} = s^n\hat{u}(\cdot,s) - \sum_{j=1}^{n} s^{j-1}u^{(n-j)}(\cdot,0)$, provided

$$\lim_{t\to\infty} u^{(m)}(\cdot,t)e^{-st} = 0,\ m = 0,\ldots n-1 \quad (\textit{General differentiation}),$$

L4. $\mathcal{L}\{(u * v)(\cdot,t)\} = \hat{u}(\cdot,s)\cdot\hat{v}(\cdot,s)$, where the convolution function is given by

$$(u*v)(\cdot,t) := \int_{-\infty}^{\infty} u(\cdot,\tau)v(\cdot,t-\tau)\,d\tau \qquad (\textit{Convolution property}),$$

L5. $\mathcal{L}\left\{\int_0^t u(\cdot,\tau)\,dt\right\} = \frac{1}{s}\hat{u}(\cdot,s)$ (*Integration property*).

For more information on Laplace transforms, see [79, 72].

I.1.2 Fourier Transforms

Definition. The Fourier transform of a function $u(x,t)$ with respect to the spatial variable x ($\Omega = \mathbb{R}$) is defined by

$$U(w,t) := \mathcal{F}\{u\}(w,t) := \int_{-\infty}^{\infty} e^{-iwx} u(x,t) dx,$$

with w being a real variable.

It is again a special integral transform where $K(x,w) := e^{-iwx}$. The upper case letter is used for the Fourier transform of a function in space. For the existence of such transforms, it is sufficient to assume $\int_{-\infty}^{\infty} |u(x,\cdot)|\, dx < \infty$, in other words $u(x,\cdot) \in L^1(\mathbb{R})$.

Definition. The inverse Fourier transform of a function $U(w,t)$ is defined as follows:

$$u(x,t) := \mathcal{F}^{-1}\{U\}(x,t) := \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iwx} U(w,t)\, dw.$$

We need the following definition before stating the Fourier inversion theorem.

Definition. The Schwarz class $\mathcal{S}$ is the collection of all C^∞ functions f such that

$$\left| x^\alpha f^{(m)}(x) \right| \to 0, \qquad \text{as } |x| \to \infty$$

for all $\alpha \geq 0$, and $m \in \mathbb{N} \cup \{0\}$.

So the class $\mathcal{S}$ consists of those infinitely many differentiable functions that decay sufficiently fast along with all its derivatives at infinity.

Theorem 40. *Suppose $U \in \mathcal{S}$ be a function and let*

$$\mathcal{F}^{-1}\{U\}(x,t) := \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iwx} U(w,t)\, dw.$$

Then for all $x \in \mathbb{R}$

$$\mathcal{F}^{-1}\{\mathcal{F}\{u\}\}(x,\cdot) = u(x,\cdot).$$

We now list some important properties of Fourier transform:

F1. $\mathcal{F}\{e^{iw_0 x} u(x,\cdot)\}(w,\cdot) = U(w - w_0, \cdot)$ (*Shifting property*),

F2. $\mathcal{F}\{u(x-\beta,\cdot)\}(w,\cdot) = e^{-i\beta w} U(w,\cdot)$ (*Translation property*),

F3. $\mathcal{F}\left\{u^{(n)}(x,\cdot)\right\}(w,\cdot) = (iw)^n U(w,\cdot)$, for $n = 0, 1, \ldots$ (*General differentiation*),

F4. $\mathcal{F}\{(u * v)(x,\cdot)\}(w,\cdot) = U(w,\cdot) \cdot V(w,\cdot)$, where the convolution function is given by

$$(u * v)(x,\cdot) := \int_{-\infty}^{\infty} u(y,\cdot) v(x-y,\cdot)\, dy \qquad (\textit{Convolution property}),$$

F5. If $f \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$, then $F := \mathcal{F}\{f\} \in L^2(\mathbb{R})$ and

$$\|f\|^2_{L^2(\mathbb{R})} = \frac{1}{2\pi} \|F\|^2_{L^2(\mathbb{R})} \quad (\textit{Parseval's identity}),$$

F6. If $f \in \mathcal{S}$ and $F := \mathcal{F}\{f\}$, then

$$\sum_{n\in\mathbb{Z}} f(n) = \sum_{k\in\mathbb{Z}} F(2k\pi) \quad (\textit{Poisson summation formula}).$$

Remark. In general we apply the Laplace transform in time, and the Fourier transform in space. However, the Laplace transform and the Fourier transform are closely connected in a number of ways. When the Laplace variable s is purely imaginary, i.e., $s = iw$, the Laplace transform (with the underlying function being extended to the entire real line, or the bilateral Laplace transform) reduces to the Fourier transform. Thus the Laplace transform is nothing but the Fourier transform of a function together with an exponential weighting. Due to this exponential weighting, the Laplace transform converges for more functions than the Fourier transform.

I.2 Fubini theorems

We state a few important results that give conditions under which a double integral can be equal to two iterated integrals, and the rule for interchanging those integrals; for more details see [75].

Theorem 41 (Fubini's theorem)**.** *If $f \in L^1(\mathbb{R}^2)$, then for almost every $x \in \mathbb{R}$, $\int_{-\infty}^{\infty} |f(x,y)|\, dy < \infty$, i.e., $f(x,\cdot) \in L^1(\mathbb{R})$. Moreover, the following interchange of the integrals holds:*

$$\int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(x,y)\, dy \right) dx = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(x,y)\, dx \right) dy = \iint_{\mathbb{R}^2} f(x,y)\, dxdy.$$

We can have a similar result as below for interchanging summation, if we take the counting measure on $\mathbb{N}$.

Theorem 42 (Fubini's theorem for series)**.** *Suppose the doubly indexed sequence $\{f_{ij}\}$ is absolutely summable, i.e., $\sum_{i,j\geq 1} |f_{ij}| < \infty$, then*

$$\sum_{i,j\in\mathbb{N}} f_{ij} = \sum_i \sum_j f_{ij} = \sum_j \sum_i f_{i,j}.$$

We now state one of the useful results for interchanging iterated integrals.

Theorem 43 (Fubini-Tonelli theorem)**.** *Suppose f is measurable on $\mathbb{R}^2$, and one of the three integrals*

$$\int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} |f(x,y)|\, dy \right) dx, \quad \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} |f(x,y)|\, dx \right) dy, \quad \iint_{\mathbb{R}^2} |f(x,y)|\, dxdy$$

is finite. Then, $f \in L^1(\mathbb{R}^2)$ and the following interchange holds:

$$\int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(x,y)\, dy \right) dx = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(x,y)\, dx \right) dy = \iint_{\mathbb{R}^2} f(x,y)\, dxdy.$$

II

Appendix B

In this appendix, we present in detail the tools to get the precise expressions of the kernels from Chapters 2 and 3, and explain the intuitive idea behind getting superlinear convergence for the heat equation and finite step convergence to the exact solution for the wave equation in DNWR for $\theta = 1/2$, and in NNWR for $\theta = 1/4$.

II.1 Kernel for the Heat equation

We work on obtaining the precise closed-form formulas of the DNWR kernel $\mathcal{L}^{-1}\left\{G^k(s)\right\}$ from Subsection 2.3.1.1 and the NNWR kernel $\mathcal{L}^{-1}\left\{Y^k(s)\right\}$ from Subsection 3.3.1.2 for $k = 1, 2, \ldots$ where

$$G(s) = \frac{\sinh((b-a)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})\cosh(b\sqrt{s/\nu})}, \quad Y(s) = \frac{\sinh^2((a-b)\sqrt{s/\nu})}{\sinh(2a\sqrt{s/\nu})\sinh(2b\sqrt{s/\nu})}.$$

Based on these expressions, the theoretical convergence curves have been plotted in Subsection 2.3.1.4. One has to consider two different cases for the DNWR method: Dirchlet subdomain bigger than Neumann subdomain ($a > b$) and the other way around. However, we consider only the case $a > b$ for the NNWR kernel because of the symmetric nature of $Y(s)$ with respect to a, b.

Case-I : $a > b$

We rewrite $G(s)$ as $-\frac{1}{\cosh(b\sqrt{s/\nu})} \cdot \frac{\sinh((a-b)\sqrt{s/\nu})}{\sinh(a\sqrt{s/\nu})}$ and find closed-form formulas for

$$\mathcal{L}^{-1}\left\{\operatorname{sech}^k(b\sqrt{s/\nu})\right\} \text{ and } \mathcal{L}^{-1}\left\{\frac{\sinh^k(\lambda\sqrt{s/\nu})}{\sinh^k(a\sqrt{s/\nu})}\right\}$$

with $\lambda = a - b$, followed by a convolution to obtain $(-1)^k\,\mathcal{L}^{-1}\left\{G^k(s)\right\}$. Note that the negative sign is absorbed in the updating term (2.3.10) for $\theta = 1/2$.

Proposition 44. *For $k = 1, 2, 3, \ldots$, we have the identities*

$$\mathcal{L}^{-1}\left\{\operatorname{sech}^k(b\sqrt{s/\nu})\right\} = \frac{2^k b}{\sqrt{\pi\nu t^3}}\sum_{m=0}^{\infty}(-1)^m\binom{m+k-1}{m}\left(m+\tfrac{k}{2}\right)e^{-\frac{(2m+k)^2b^2}{4\nu t}},$$
$$\mathcal{L}^{-1}\left\{\frac{\sinh^k(\lambda\sqrt{s/\nu})}{\sinh^k(a\sqrt{s/\nu})}\right\} = \frac{1}{\sqrt{\pi\nu t^3}}\sum_{m=0}^{\infty}\sum_{l=0}^{k}(-1)^l\binom{m+k-1}{m}\binom{k}{l}\left(\tfrac{b}{2}k+l\lambda+ma\right)e^{-\frac{(2ma+kb+2l\lambda)^2}{4\nu t}}. \tag{II.1.1}$$

Moreover, both functions are positive for $t > 0$.

Proof. Since $\left|e^{-2b\sqrt{s/\nu}}\right| < 1$ for $\operatorname{Re}(s) > 0$, we can expand sech into an infinite binomial series as

$$\begin{aligned}\operatorname{sech}^k(b\sqrt{s/\nu}) &= \left(\frac{2}{e^{b\sqrt{s/\nu}}+e^{-b\sqrt{s/\nu}}}\right)^k = 2^k e^{-kb\sqrt{s/\nu}}\left(1+e^{-2b\sqrt{s/\nu}}\right)^{-k}\\ &= 2^k\sum_{m=0}^{\infty}(-1)^m\binom{m+k-1}{m}e^{-(2m+k)b\sqrt{s/\nu}}.\end{aligned}$$

Now using the inverse Laplace transform (see Oberhettinger [72])

$$\mathcal{L}^{-1}\left(e^{-\alpha\sqrt{s}}\right) = \frac{\alpha}{\sqrt{4\pi t^3}}e^{-\alpha^2/4t}, \quad \alpha > 0, \tag{II.1.2}$$

we obtain

$$\begin{aligned}\mathcal{L}^{-1}\left(\operatorname{sech}^k(b\sqrt{s/\nu})\right) &= 2^k\sum_{m=0}^{\infty}(-1)^m\binom{m+k-1}{m}\mathcal{L}^{-1}\left(e^{-(2m+k)b\sqrt{s/\nu}}\right)\\ &= 2^k\sum_{m=0}^{\infty}(-1)^m\binom{m+k-1}{m}\frac{(2m+k)b}{\sqrt{4\pi\nu t^3}}e^{-\frac{(2m+k)^2b^2}{4\nu t}}\end{aligned} \tag{II.1.3}$$

To justify taking the inverse Laplace transform term by term, we show that the Laplace transform of the right-hand side of (II.1.3) indeed gives $\operatorname{sech}^k(b\sqrt{s/\nu})$. Let

$$y_m(t) = 2^k(-1)^m\binom{m+k-1}{m}\frac{(2m+k)b}{\sqrt{4\pi\nu t^3}}e^{-(2m+k)^2b^2/4\nu t}$$

be the mth term of the series. Then for any $s_0 > 0$, we have for $\operatorname{Re}(s) > s_0$

$$\begin{aligned}\int_0^\infty |e^{-st}y_m(t)|\,dt &\le 2^k\int_0^\infty e^{-s_0t}\binom{m+k-1}{m}\frac{(2m+k)b}{\sqrt{4\pi\nu t^3}}e^{-(2m+k)^2b^2/4\nu t}\,dt\\ &= 2^k\binom{m+k-1}{m}e^{-(2m+k)b\sqrt{s_0/\nu}}.\end{aligned}$$

We now recall (2.3.14) to write

$$\sum_{m=0}^{\infty}\int_0^\infty |e^{-st}y_m(t)|\,dt \le \operatorname{cosech}^k(b\sqrt{s_0/\nu}) < \infty.$$

So by Fubini-Tonelli's theorem, we obtain for all $\mathrm{Re}(s) \geq s_0$

$$\int_0^\infty e^{-st} \sum_{m=0}^{\infty} y_m(t)\, dt = \sum_{m=0}^{\infty} \int_0^\infty e^{-st} y_m(t)\, dt = \mathrm{sech}^k(b\sqrt{s/\nu}).$$

The first identity of (II.1.1) thus follows by taking the inverse Laplace transform on both sides. Also the positivity follows from Lemma 9.

For the second identity, we again recall the identity (2.3.14) to write

$$\mathrm{cosech}^k(a\sqrt{s/\nu}) = 2^k \sum_{m=0}^{\infty} \binom{m+k-1}{m} e^{-(2m+k)a\sqrt{s/\nu}},$$

which means, for $0 < \lambda < a$

$$\begin{aligned}
\frac{\sinh^k(\lambda\sqrt{s/\nu})}{\sinh^k(a\sqrt{s/\nu})} &= 2^k \sinh^k(\lambda\sqrt{s/\nu}) \sum_{m=0}^{\infty} \binom{m+k-1}{m} e^{-(2m+k)a\sqrt{s/\nu}} \\
&= e^{k\lambda\sqrt{s/\nu}} \left(1 - e^{-2\lambda\sqrt{s/\nu}}\right)^k \sum_{m=0}^{\infty} \binom{m+k-1}{m} e^{-(2m+k)a\sqrt{s/\nu}} \\
&= \sum_{m=0}^{\infty} \sum_{l=0}^{k} (-1)^l \binom{m+k-1}{m} \binom{k}{l} e^{-(2ma+k(a-\lambda)+2l\lambda)\sqrt{s/\nu}}. \qquad \text{(II.1.4)}
\end{aligned}$$

We now apply (II.1.2) to obtain the inverse Laplace transform as

$$\mathcal{L}^{-1}\left\{\frac{\sinh^k(\lambda\sqrt{s/\nu})}{\sinh^k(a\sqrt{s/\nu})}\right\} = \sum_{m=0}^{\infty} \sum_{l=0}^{k} (-1)^l \binom{m+k-1}{m} \binom{k}{l} \frac{\frac{k}{2}(a-\lambda)+l\lambda+ma}{\sqrt{\pi\nu t^3}} e^{-\frac{(2ma+k(a-\lambda)+2l\lambda)^2}{4\nu t}},$$

where we justify the interchanging of sums and inverse transforms in the same way as above. The required result is obtained by replacing λ by $a - b$ in the above expression, and the positivity again follows from Lemma 9. □

We now compute the two series in the right hand side of (II.1.1), and plot them in Figure II.1.1 for different values of k. A convolution of these two functions in (II.1.1) gives the kernel $(-1)^k \mathcal{L}^{-1}\left\{G^k(s)\right\}$ for $a > b$; see Figure II.1.2. We observe that the curves shift to the right and at the same time, the peak decreases as the value of k increases. So, if one only considers a small time window, the peak will eventually exit the time window for k large enough and its contribution will be vanishingly small in the updating term (2.3.10). This is the intuitive explanation why we get superlinear convergence for $\theta = 1/2$ in small time windows.

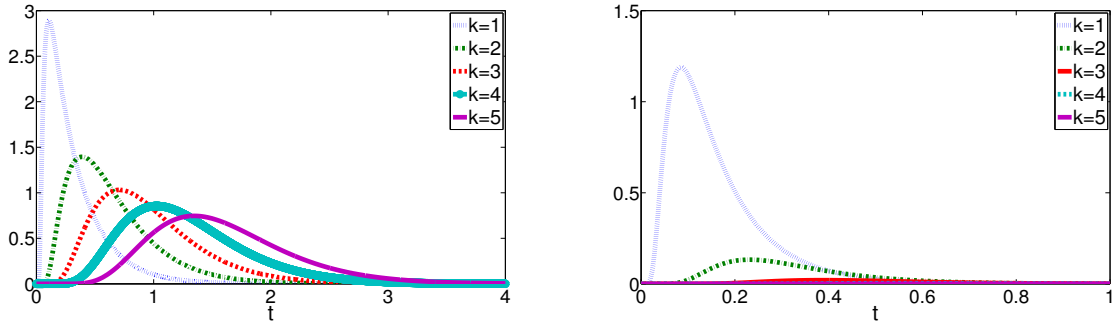


Figure II.1.1: For a particular set of choices $a = 1, b = 0.8, \nu = 1$, and as k varies, plot of the curves $\mathcal{L}^{-1}\left\{\text{sech}^k(b\sqrt{s/\nu})\right\}$ on the left, and $\mathcal{L}^{-1}\left\{\frac{\sinh^k((a-b)\sqrt{s/\nu})}{\sinh^k(a\sqrt{s/\nu})}\right\}$ on the right

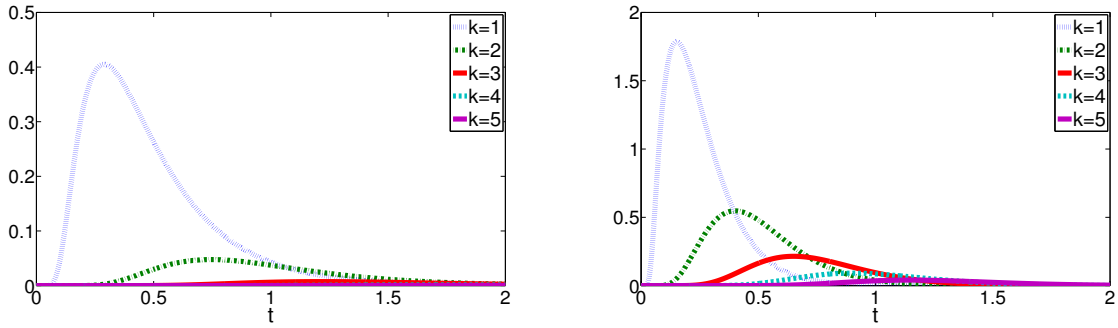


Figure II.1.2: The curves $(-1)^k \mathcal{L}^{-1}\left\{G^k(s)\right\}$: on the left for $a = 1, b = 0.8, \nu = 1$, and on the right for a different set $a = 1, b = 0.5, \nu = 1$

Case-II : $a < b$

For the other case, i.e., $a < b$, we rewrite $G(s)$ as $\frac{\sqrt{s}}{\sinh(a\sqrt{s/\nu})} \cdot \frac{\sinh((b-a)\sqrt{s/\nu})}{\sqrt{s}\cosh(b\sqrt{s/\nu})}$ and derive the closed-form formulas for

$$\mathcal{L}^{-1}\left\{(\sqrt{s})^k \text{cosech}^k(a\sqrt{s/\nu})\right\} \text{ and } \mathcal{L}^{-1}\left\{\frac{1}{(\sqrt{s})^k}\frac{\sinh^k(\mu\sqrt{s/\nu})}{\cosh^k(b\sqrt{s/\nu})}\right\}$$

with $\mu = b - a$. A convolution of these two functions finally produces $\mathcal{L}^{-1}\left\{G^k(s)\right\}$. We need the following Lemma before proceeding with the main results.

Lemma 45. *For $k = 1, 2, 3, ...$, $f_k(t;\alpha) = \mathcal{L}^{-1}\{s^{-k/2}e^{-\alpha\sqrt{s}}\}$ has the form:*

$$\begin{aligned}
f_1(t;\alpha) &= \frac{1}{\sqrt{\pi t}}\exp\left(-\frac{\alpha^2}{4t}\right), \; f_2(t;\alpha) = \text{erfc}\left(\frac{\alpha}{2\sqrt{t}}\right), \\
f_3(t;\alpha) &= 2\left(\frac{t}{\pi}\right)^{1/2}\exp\left(-\frac{\alpha^2}{4t}\right) - \alpha.\text{erfc}\left(\frac{\alpha}{2\sqrt{t}}\right), \\
f_k(t;\alpha) &= \frac{2t}{k-2}f_{k-2}(t;\alpha) - \frac{\alpha}{k-2}f_{k-1}(t;\alpha), \quad \text{for } k \geq 4.
\end{aligned}$$

Proof. Set $g_k(s) = s^{-k/2}e^{-\alpha\sqrt{s}}$. We want to have the inverse Laplace transform of $g_k(s)$ for different values of k. For $k = 0$, we have from [72]

$$f_0(t;\alpha) = \mathcal{L}^{-1}\{e^{-\alpha\sqrt{s}}\} = \frac{\alpha}{2\sqrt{\pi t^3}}e^{-\alpha^2/4t}.$$

Now to get $g_1(s)$, we integrate $g_0(s)$ with respect to α and use Fubini's theorem to write

$$\int_\alpha^\infty e^{-\tau\sqrt{s}}\,d\tau = \int_\alpha^\infty \int_0^\infty f_0(t;\tau)e^{-st}\,dt\,d\tau = \mathcal{L}\left\{\int_\alpha^\infty f_0(t;\tau)\,d\tau\right\}. \tag{II.1.5}$$

On the left hand side, we get a factor of $1/\sqrt{s}$ that comes out due to integration with respect to α, and on the right we have

$$\int_\alpha^\infty f_0(t;\tau)\,d\tau = \frac{1}{2\sqrt{\pi t^3}}\int_\alpha^\infty \tau e^{-\tau^2/4t}\,d\tau = \frac{1}{\sqrt{\pi t}}\left[-e^{-\tau^2/4t}\right]_\alpha^\infty = \frac{1}{\sqrt{\pi t}}e^{-\alpha^2/4t}.$$

So from (II.1.5), we get

$$f_1(t;\alpha) = \mathcal{L}^{-1}\{s^{-1/2}e^{-\alpha\sqrt{s}}\} = \int_\alpha^\infty f_0(t;\tau)\,d\tau = \frac{1}{\sqrt{\pi t}}e^{-\alpha^2/4t}.$$

Now we can recursively define

$$f_{k+1}(t;\alpha) = \int_\alpha^\infty f_k(t;\tau)\,d\tau.$$

Note that $\dfrac{\partial f_{k+1}}{\partial\alpha}(t;\alpha) = -f_k(t;\alpha)$. So to get a recurrence for $f_k(t;\alpha)$, we perform integration by parts,

$$\begin{aligned} f_{k+1}(t;\alpha) &= \int_\alpha^\infty f_k(t;\tau)\,d\tau \\ &= [\tau f_k(t;\tau)]_\alpha^\infty - \int_\alpha^\infty \tau f_k'(t;\tau)\,d\tau \\ &= -\alpha f_k(t;\alpha) + \int_\alpha^\infty \tau f_{k-1}(t;\tau)\,d\tau. \end{aligned}$$

Here one must assume that $\lim\limits_{\alpha\to\infty}\alpha f_k(t;\alpha) = 0$ for any fixed $t > 0$. In general, we will also need $\lim\limits_{\alpha\to\infty}\alpha^m f_k(t;\alpha) = 0$ for any $m > 0$ and $t > 0$, which is true for $f_0(t;\alpha)$ and $f_1(t;\alpha)$, and will be true for general $f_k(t;\alpha)$. If we continue to perform integration by parts, we will have

$$f_k(t;\alpha) = -\sum_{l=1}^{k-1}\frac{\alpha^l}{l!}f_{k-l}(t;\alpha) + \frac{1}{(k-1)!}\int_\alpha^\infty \tau^{k-1}f_0(t;\tau)\,d\tau.$$

In other words, we have

$$\sum_{l=0}^{k-1}\frac{\alpha^l}{l!}f_{k-l}(t;\alpha) = \frac{1}{(k-1)!}\cdot\frac{1}{2\sqrt{\pi t^3}}\int_\alpha^\infty \tau^k e^{-\tau^2/4t}\,d\tau. \tag{II.1.6}$$

Let us define the moments $\mu_k = \int_\alpha^\infty \tau^k e^{-\tau^2/4t}\,d\tau$. Then we can write (II.1.6) for $k = 1, 2, \dots$ as a linear system:

$$\begin{bmatrix} 1 & & & \\ \alpha & 1 & & \\ \vdots & \ddots & \ddots & \\ \frac{\alpha^{k-1}}{(k-1)!} & \cdots & \alpha & 1 \end{bmatrix} \begin{pmatrix} f_1(t;\alpha) \\ f_2(t;\alpha) \\ \vdots \\ f_k(t;\alpha) \end{pmatrix} = \frac{1}{2\sqrt{\pi t^3}} \begin{pmatrix} \mu_1/0! \\ \mu_2/1! \\ \vdots \\ \mu_k/(k-1)! \end{pmatrix}.$$

Notice that the matrix on the left (which we call A) can be written as

$$A = I + \alpha N + \frac{\alpha^2}{2!}N^2 + \cdots + \frac{\alpha^{k-1}}{(k-1)!}N^{k-1} = \exp(\alpha N),$$

where N is the nilpotent matrix

$$N = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ 0 & & 1 & 0 \end{bmatrix}.$$

This means

$$A^{-1} = \exp(-\alpha N) = \sum_{l=0}^{k-1} \frac{(-1)^l \alpha^l}{l!} N^l,$$

or, in other words for $k = 1, 2, \dots$

$$\begin{aligned} f_k(t;\alpha) &= \frac{1}{2\sqrt{\pi t^3}} \sum_{l=1}^{k} \frac{(-1)^{k-l}\alpha^{k-l}\mu_l}{(l-1)!(k-l)!} \\ &= \frac{1}{2\sqrt{\pi t^3}} \cdot \frac{1}{(k-1)!} \sum_{m=0}^{k-1} (-1)^m \binom{k-1}{m} \alpha^m \mu_{k-m}. \end{aligned}$$

Using the definition of μ_k and the binomial theorem, we can actually write

$$\begin{aligned} f_k(t;\alpha) &= \frac{1}{2\sqrt{\pi t^3}} \cdot \frac{1}{(k-1)!} \int_\alpha^\infty \tau \left[\sum_{m=0}^{k-1} \binom{k-1}{m} (-1)^m \alpha^m \tau^{k-1-m} \right] e^{-\tau^2/4t}\,d\tau \\ &= \frac{1}{2\sqrt{\pi t^3}} \cdot \frac{1}{(k-1)!} \int_\alpha^\infty \tau(\tau-\alpha)^{k-1} e^{-\tau^2/4t}\,d\tau. \end{aligned}$$

Integrating by parts once, we get

$$\begin{aligned} f_k(t;\alpha) &= \frac{1}{2\sqrt{\pi t^3}} \cdot \frac{1}{(k-1)!} \left\{ \left[-2t(\tau-\alpha)^{k-1} e^{-\frac{\tau^2}{4t}} \right]_{\tau=\alpha}^{\infty} + 2t(k-1) \int_\alpha^\infty (\tau-\alpha)^{k-2} e^{-\frac{\tau^2}{4t}}\,d\tau \right\} \\ &= \frac{1}{\sqrt{\pi t}} \cdot \frac{1}{(k-2)!} \int_\alpha^\infty (\tau-\alpha)^{k-2} e^{-\tau^2/4t}\,d\tau. \end{aligned} \tag{II.1.7}$$

So in particular for $k = 2$, we have

$$f_2(t;\alpha) = \frac{1}{\sqrt{\pi t}} \int_\alpha^\infty e^{-\tau^2/4t}\,d\tau = \operatorname{erfc}\Big(\frac{\alpha}{2\sqrt{t}}\Big).$$

For $k=3$, we get from (II.1.7)

$$f_3(t;\alpha) = \frac{1}{\sqrt{\pi t}}\int_\alpha^\infty \tau e^{-\tau^2/4t}\,d\tau - \frac{\alpha}{\sqrt{\pi t}}\int_\alpha^\infty e^{-\tau^2/4t}\,d\tau$$
$$= 2\Big(\frac{t}{\pi}\Big)^{1/2} e^{-\alpha^2/4t} - \alpha\cdot\operatorname{erfc}\Big(\frac{\alpha}{2\sqrt{t}}\Big).$$

For $k\geq 4$, we can calculate $f_k(t;\alpha)$ using the following recurrence relation:

$$f_k(t;\alpha) = \frac{1}{\sqrt{\pi t}}\frac{1}{(k-2)!}\left\{\int_\alpha^\infty \tau(\tau-\alpha)^{k-3}e^{-\tau^2/4t}\,d\tau - \alpha\int_\alpha^\infty (\tau-\alpha)^{k-3}e^{-\tau^2/4t}\,d\tau\right\}$$
$$= \frac{1}{\sqrt{\pi t}}\frac{1}{(k-2)!}\left\{2t(k-3)\int_\alpha^\infty (\tau-\alpha)^{k-4}e^{-\tau^2/4t}\,d\tau - \alpha\int_\alpha^\infty (\tau-\alpha)^{k-3}e^{-\tau^2/4t}\,d\tau\right\}$$
$$= \frac{1}{k-2}\Big[2t\,f_{k-2}(t;\alpha) - \alpha\,f_{k-1}(t;\alpha)\Big].$$

This completes the result. □

Proposition 46. *For $k=1,2,3,\ldots$, we have the following identities*

$$\mathcal{L}^{-1}\left\{\sqrt{s^k}\,\operatorname{cosech}^k(a\sqrt{s/\nu})\right\} = \frac{2^{\frac{k-1}{2}}}{\sqrt{\pi t^{k+2}}}\cdot\sum_{m=0}^\infty \binom{m+k-1}{m}\exp\left(-\frac{(2m+k)^2a^2}{4\nu t}\right)\mathrm{He}_{k+1}\left(\frac{(2m+k)a}{\sqrt{2\nu t}}\right),$$
$$\mathcal{L}^{-1}\left\{\frac{1}{(\sqrt{s})^k}\sinh^k(\mu\sqrt{s/\nu})\operatorname{sech}^k(b\sqrt{s/\nu})\right\} = \sum_{m=0}^\infty\sum_{l=0}^k (-1)^{m+l}\binom{m+k-1}{m}\binom{k}{l} f_k(t;\xi_{l,m}/\sqrt{\nu}), \tag{II.1.8}$$

where $\mathrm{He}_n(x)$ are Hermite polynomials of degree n, $\xi_{l,m}=ka+2l\mu+2mb, \mu=b-a$ and $f_k(t;\alpha)$ has the form as Lemma 45 for $k=1,2,3,\ldots$.

Proof. We start with the identity (2.3.14)

$$\operatorname{cosech}^k(a\sqrt{s/\nu}) = 2^k\sum_{m=0}^\infty \binom{m+k-1}{m} e^{-(2m+k)a\sqrt{s/\nu}},$$

to write

$$(\sqrt{s})^k\operatorname{cosech}^k(a\sqrt{s/\nu}) = 2^k s^{k/2}\sum_{m=0}^\infty \binom{m+k-1}{m} e^{-(2m+k)a\sqrt{s/\nu}}. \tag{II.1.9}$$

Now we need to use the inverse Laplace transform (see Oberhettinger [72])

$$\mathcal{L}^{-1}\left\{s^{n/2}e^{-\alpha\sqrt{s}}\right\} = \frac{2^{-\frac{n+1}{2}}}{\sqrt{\pi}}t^{-\frac{n}{2}-1}\exp(-\alpha^2/4t)\mathrm{He}_{n+1}\left(\frac{\alpha}{\sqrt{2t}}\right),$$

where $\mathrm{He}_n(x)$ is the Hermite polynomial of degree n. Taking the inverse Laplace transform term by term (which is again justified in virtue of Fubini-Tonelli's theorem) on the right hand side of (II.1.9), we obtain

$$\mathcal{L}^{-1}\left\{\sqrt{s^k}\,\operatorname{cosech}^k(a\sqrt{s/\nu})\right\} = \frac{2^{\frac{k-1}{2}}}{\sqrt{\pi t^{k+2}}}\cdot\sum_{m=0}^\infty \binom{m+k-1}{m}\exp\left(-\frac{(2m+k)^2a^2}{4\nu t}\right)\mathrm{He}_{k+1}\left(\frac{(2m+k)a}{\sqrt{2\nu t}}\right).$$

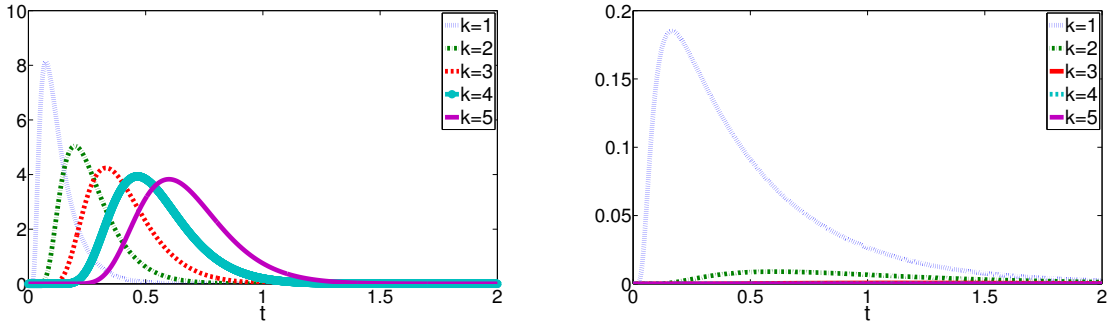


Figure II.1.3: For a particular set of choices $a = 0.9$, $b = 1$, $\nu = 1$, and as k varies, plot of the curves $\mathcal{L}^{-1}\left\{(\sqrt{s})^k \text{cosech}^k(a\sqrt{s/\nu})\right\}$ on the left, and $\mathcal{L}^{-1}\left\{\frac{1}{(\sqrt{s})^k}\sinh^k((b-a)\sqrt{s/\nu})\text{sech}^k(b\sqrt{s/\nu})\right\}$ on the right

For the second identity, in a similar way as (II.1.4), we have

$$\frac{1}{(\sqrt{s})^k}\frac{\sinh^k(\mu\sqrt{s/\nu})}{\cosh^k(b\sqrt{s/\nu})} = s^{-\frac{k}{2}}\sum_{m=0}^{\infty}\sum_{l=0}^{k}(-1)^{m+l}\binom{m+k-1}{m}\binom{k}{l}e^{-(k(b-\mu)+2l\mu+2mb)\sqrt{s/\nu}}.$$

We then take the inverse Laplace transform on both sides of this expression and use Lemma 45 to obtain

$$\begin{aligned}\mathcal{L}^{-1}\left\{\frac{1}{(\sqrt{s})^k}\frac{\sinh^k(\mu\sqrt{s/\nu})}{\cosh^k(b\sqrt{s/\nu})}\right\} &= \sum_{m=0}^{\infty}\sum_{l=0}^{k}(-1)^{m+l}\binom{m+k-1}{m}\binom{k}{l}\mathcal{L}^{-1}\left\{s^{-\frac{k}{2}}e^{-\xi_{l,m}\sqrt{s/\nu}}\right\}\\ &= \sum_{m=0}^{\infty}\sum_{l=0}^{k}(-1)^{m+l}\binom{m+k-1}{m}\binom{k}{l}f_k(t;\xi_{l,m}/\sqrt{\nu}),\end{aligned}$$

where $\xi_{l,m} = k(b-\mu)+2l\mu+2mb$ for $k = 1, 2, 3, \ldots$ and $f_k(t;\alpha)$ has the form as in Lemma 45. Now we replace μ by $b-a$ in the above expression to get the required result. □

In a similar way as in Case-I we compute the series of (II.1.8), and plot them in Figure II.1.3 for different values of k. In the left of Figure II.1.4 we plot the kernel $\mathcal{L}^{-1}\left\{G^k(s)\right\}$ for $a < b$ by convolving the two functions of (II.1.8). Clearly like Case-I, these curves have the same property, that illustrates the reason of getting superlinear convergence in the DNWR algorithm for $\theta = 1/2$.

For the NNWR algorithm we can write

$$Y(s) = \frac{1}{4}\left(G(s;a,b) + G(s;b,a)\right), \tag{II.1.10}$$

where we call $G(s;a,b) := G(s)$. So if we assume $a > b$, then the first function $G(s;a,b)$ in (II.1.10) is to be calculated by Case-I, and $G(s;b,a)$ according to Case-II. On the right of Figure II.1.4 we plot the kernel $\mathcal{L}^{-1}\left\{Y^k(s)\right\}$ for $a = 1, b = 0.7$ and for different values of k. These curves also shift to the right and at the same time, the peak decreases as k increases. Therefore for small time windows, the peak will exit the time limit for sufficiently large k, again resulting in the superlinear convergence of NNWR.

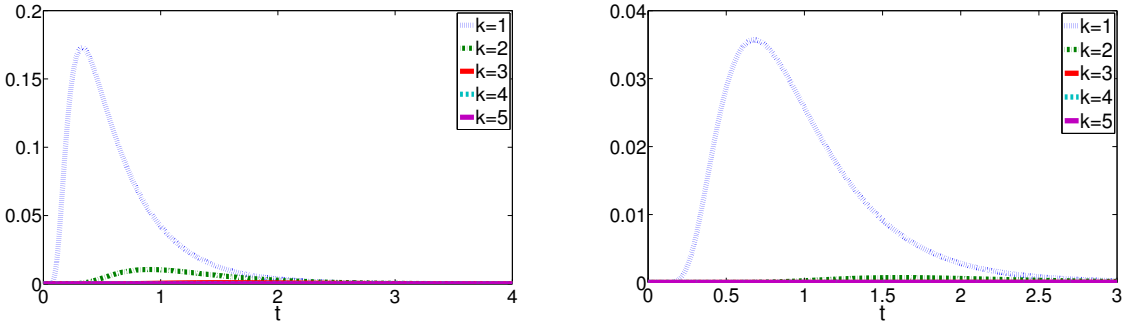


Figure II.1.4: As k varies, the curves $\mathcal{L}^{-1}\left\{G^k(s)\right\}$ for $a = 0.9$, $b = 1$, $\nu = 1$ on the left, and the curves $\mathcal{L}^{-1}\left\{Y^k(s)\right\}$ for $a = 1$, $b = 0.7$, $\nu = 1$ on the right

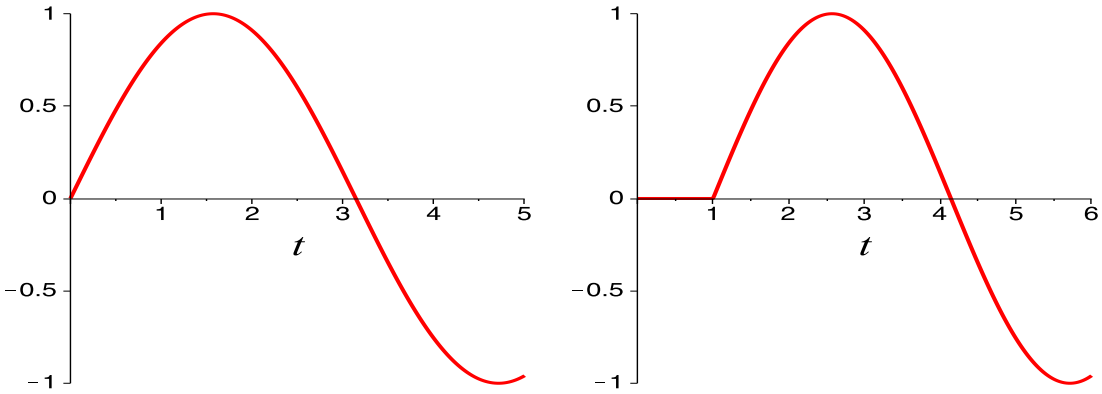


Figure II.1.5: Example of time-shifting for the function $F(t) = \sin(t)$: $\mathcal{L}^{-1}\left\{\hat{F}(s)\right\}$ on the left, and $\mathcal{L}^{-1}\left\{e^{-s}\hat{F}(s)\right\}$ on the right

II.2 Kernel for the Wave equation

By analyzing the expression of the kernel for the wave equation, we try to understand the reason behind the finite step convergence to the exact solution for a particular value of the parameter θ. Recall that in case of the DNWR algorithm, the updating term (2.4.6) has the form

$$\hat{h}^k(s) = \left(-\frac{1}{2}\right)^k \left\{G_b^a(s)\right\}^k \hat{h}^0(s),\ k = 1, 2, \ldots$$

for $\theta = 1/2$. The error $h^k(t)$ at the kth iteration is therefore obtained by convolving the initial error $h^0(t)$ with $\mathcal{L}^{-1}\left\{\left(G_b^a(s)\right)^k(s)\right\}$, where $G_b^a(s)$ is of the form (by Lemma 19 of Subsection 2.4.1.2)

$$\begin{aligned} G_b^a(s) &:= \coth(as/c)\tanh(bs/c) - 1 \\ &= 2\sum_{m=1}^{\infty} e^{-2ams/c} - 2\sum_{n=1}^{\infty}(-1)^{n-1}e^{-2bns/c} - 4\sum_{n=1}^{\infty}\sum_{m=1}^{\infty}(-1)^{n-1}e^{-2(bn+am)s/c}. \end{aligned}$$

Now it is interesting to note that by the time-shifting property L2* of Laplace transforms, we can write

$$\mathcal{L}^{-1}\left\{e^{-\beta s}\hat{h}^0(s)\right\} = h^0(t-\beta)H(t-\beta),$$

with $H(t) = \begin{cases} 1, & t > 0, \\ 0, & t \leq 0. \end{cases}$ being the Heaviside step function. The right hand side of the above expression becomes identically zero for $t \leq \beta$, so that for sufficiently small time window length T (e.g., $T \leq \beta$) the error becomes zero and results to convergence in the next iteration. In Figure II.1.5 we plot $\mathcal{L}^{-1}\left\{\hat{F}(s)\right\}$ with $F(t) = \sin(t)$ on the left, and show the effect of time-shifting on the right. A similar phenomenon describes the convergence for the NNWR algorithm.

*see Appendix A

Bibliography

[1] V. Akçelik, G. Biros, O. Ghattas, K. R. Long, and B. van Bloemen Waanders, *A variational finite element method for source inversion for convective-diffusive transport*, Finite Elem. Anal. Des. **39** (2003), no. 8, 683–705.

[2] A. Bayliss and E. Turkel, *Radiation boundary conditions for wave-like equations*, Comm. Pure Appl. Math. **33** (1980), no. 6, 707–725.

[3] A. Bellen and M. Zennaro, *The use of Runge-Kutta formulae in waveform relaxation methods*, Appl. Numer. Math. **11** (1993), 95–114.

[4] J.-D. Benamou, *Décomposition de domaine pour le contrôle optimal de systémes gouvernés par des équations aux dérivées partielles elliptiques*, C.R. Acad. Sci. Paris Sér. I Math. **317** (1993), no. 2, 205–209.

[5] ———, *A domain decomposition method with coupled transmission conditions for the optimal control of systems governed by elliptic partial differential equations*, SIAM J. Numer. Anal. **33** (1996), no. 6, 2401–2416.

[6] ———, *A domain decomposition method for control problems*, Domain Decomposition in Science and Engineering IX (Norway) (P. Bjørstad, M. Espedal, and D. Keyes, eds.), 1998, pp. 266–273.

[7] ———, *Domain decomposition, optimal control of systems governed by partial differential equations, and synthesis of feedback laws*, J. Optim. Theory Appl. **102** (1999), no. 1, 15–36.

[8] D. Bennequin, M. J. Gander, and L. Halpern, *A Homographic Best Approximation Problem with Application to Optimized Schwarz Waveform Relaxation*, Math. Comp. **78** (2009), no. 265, 185–223.

[9] M. Bjørhus, *A note on the convergence of discretized dynamic iteration*, BIT (1995), 291–296.

[10] ———, *On Domain Decomposition, Subdomain Iteration and Waveform Relaxation*, Ph.D. thesis, University of Trondheim, Norway, 1995.

[11] P. E. Bjørstad and O. B. Widlund, *Iterative Methods for the Solution of Elliptic Problems on Regions Partitioned into Substructures*, SIAM J. Numer. Anal. **23** (1986), no. 6.

[12] A. Bounaim, *On the optimal control problem of the heat equation: new formulation of the problem using a non-overlapping domain decomposition technique*, Tech. report, University of Oslo, Scientific Computing Group, Department of Informatics, 2002.

[13] J. F. Bourgat, R. Glowinski, P. L. Tallec, and M. Vidrascu, *Variational Formulation and Algorithm for Trace Operator in Domain Decomposition Calculations*, Domain Decomposition Methods (T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds.), SIAM, 1989, pp. 3–16.

[14] J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *An Iterative Method for Elliptic Problems on Regions Partitioned into Substructures*, Math. Comp. **46** (1986), no. 174, 361–369.

[15] X.-C. Cai, *Additive Schwarz algorithms for parabolic convection-diffusion equations*, Numer. Math. **60** (1991), 41–61.

[16] ———, *Multiplicative Schwarz Methods for Parabolic Problems*, SIAM J. Sci. Comput. **15** (1994), no. 3, 587–603.

[17] J. R. Cannon, *The One-Dimensional Heat Equation*, Addison-Wesley Publishing Company, 1984.

[18] R. V. Churchill, *Operational Mathematics*, 2nd ed., McGraw-Hill, 1958.

[19] M. Dryja, *An additive Schwarz algorithm for two- and three-dimensional finite element elliptic problems*, Domain Decomposition Methods (Philadelphia) (T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds.), SIAM, 1989.

[20] M. Dryja and O. B. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Tech. report, Courant Inst., Tech. Rep. 339, also Ultracomputer Note 131, Dept. of Comp. Sci., 1987.

[21] ———, *Some domain decomposition algorithms for elliptic problems*, Iterative Methods for Large Linear Systems (San Diego) (L. Hayes and D. Kincaid, eds.), Academic Press, 1989, pp. 273–291.

[22] B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp. **31** (1977), no. 139, 629–651.

[23] C. Farhat and F.-X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Internat. J. Numer. Methods Engrg. **32** (1991), 1205–1227.

[24] D. Funaro, A. Quarteroni, and P. Zanolli, *An Iterative Procedure with Interface Relaxation for Domain Decomposition Methods*, SIAM J. Numer. Anal. **25** (1988), no. 6, 1213–1236.

[25] M. J. Gander, *Overlapping Schwarz for linear and nonlinear parabolic problems*, Proceedings of the 9th International Conference on Domain Decomposition, 1996, pp. 97–104.

[26] ———, *Optimized Schwarz methods*, SIAM J. Numer. Anal. **44** (2006), no. 2, 699–732.

[27] M. J. Gander and L. Halpern, *Méthodes de relaxation d'ondes (SWR) pour l'équation de la chaleur en dimension 1*, C. R. Acad. Sci. Paris **336** (2003), no. 6, 519–524.

[28] ———, *Absorbing Boundary Conditions for the Wave Equation and Parallel Computing*, Math. Comp. **74** (2004), no. 249, 153–176.

[29] ———, *Optimized Schwarz Waveform Relaxation for Advection Reaction Diffusion Problems*, SIAM J. Num. Anal. **45** (2007), no. 2, 666–697.

[30] M. J. Gander, L. Halpern, and F. Nataf, *Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation*, 11th International Conference on Domain Decomposition in Science and Engineering (C.-H. Lai, P. E. Bjørstad, M. Cross, and O. B. Widlund, eds.), 1999, pp. 253–260.

[31] ———, *Domain decomposition methods for wave propagation, mathematical and numerical aspects of wave propagation*, SIAM, 2000, pp. 310–314.

[32] ———, *Optimal Schwarz Waveform Relaxation for the One Dimensional Wave Equation*, SIAM J. Num. Anal. **41** (2003), no. 5, 1643–1681.

[33] M. J. Gander and C. Japhet, *An Algorithm for Non-Matching Grid Projections with Linear Complexity*, Domain Decomposition in Science and Engineering XVIII (M. Bercovier, M. J. Gander, D. Keyes, and O. Widlund, eds.), 2008.

[34] ———, *Algorithm 932: PANG: Software for Non-Matching Grid Projections in 2d and 3d with Linear Complexity*, ACM Transactions on Mathematical Software (TOMS) **40** (2013), no. 1, 6:1–6:25.

[35] M. J. Gander, F. Kwok, and B. C. Mandal, *Dirichlet-Neumann and Neumann-Neumann Waveform Relaxation for the Wave Equation*, to appear in Domain Decomposition in Science and Engineering XXII, Springer-Verlag.

[36] ———, *Dirichlet-Neumann Waveform Relaxation Method for the 1D and 2D Heat and Wave Equations in Multiple subdomains*, in Preparation.

[37] ———, *Dirichlet-Neumann and Neumann-Neumann Waveform Relaxation Algorithms for Parabolic Problems*, submitted (arXiv:1311.2709).

[38] M. J. Gander, F. Kwok, and G. Wanner, *Constrained Optimization: from Lagrangian Mechanics to Optimal Control and PDE Constraints*, Optimization with PDE Constraints, LNCSE **101** (2014), 151–202.

[39] M. J. Gander and C. Rohde, *Overlapping Schwarz Waveform Relaxation for Convection-Dominated Nonlinear Conservation Laws*, SIAM J. Sci. Comput. **27** (2005), no. 2, 415–439.

[40] M. J. Gander and A. M. Stuart, *Space-time continuous analysis of waveform relaxation for the heat equation*, SIAM J. for Sci. Comput. **19** (1998), no. 6, 2014–2031.

[41] E. Giladi and H. Keller, *Space time domain decomposition for parabolic problems*, Tech. Report 97-4, Center for research on parallel computation CRPC, Caltech, 1997.

[42] L. Halpern, *Schwarz Waveform Relaxation Algorithms*, 17th International Conference on Domain Decomposition in Science and Engineering (U. Langer, M. Discacciati, D. E. Keyes, O. B. Widlund, and W. Zulehner, eds.), Springer, 2008, pp. 57–69.

[43] M. Heinkenschloss, *Time-domain decomposition iterative methods for the solution of distributed linear quadratic optimal control problems*, J. Comput. Appl. Math. **173** (2005), 169–198.

[44] M. Heinkenschloss and M. Herty, *A spatial domain decomposition method for parabolic optimal control problems*, J. Comput. Appl. Math. **201** (2007), no. 1, 88–111.

[45] M. Heinkenschloss and H. Nguyen, *Balancing neumann-neumann methods for elliptic optimal control problems*, Domain Decomposition Methods in Science and Engineering (Springer, Heidelberg) (R. Kornhuber, R.H.W Hoppe, J. Périaux, O. Pironneau, O.B. Widlund, and J. Xu, eds.), vol. 40, 2004, pp. 589–596.

[46] ———, *Neumann-Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems*, SIAM J. Sci. Comput. **28** (2006), no. 3, 1001–1028.

[47] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with Pde Constraints. Mathematical Modelling: Theory and Applications*, Springer **23** (2009).

[48] T. T. P. Hoang, *Space-time domain decomposition methods for mixed formulations of flow and transport problems in porous media*, Ph.D. thesis, University Paris 6, France, 2013.

[49] J. Janssen and S.: Vandewalle, *Multigrid Waveform Relaxation on Spatial Finite Element Meshes: The Continuous-time Case*, SIAM J. Numer. Anal. **33** (1996), no. 2, 456–474.

[50] R. Jeltsch and B. Pohl, *Waveform relaxation with overlapping splittings*, SIAM J. Sci. Comput. **16** (1995), no. 1, 40–49.

[51] F. Kwok, *Neumann-Neumann Waveform Relaxation for the Time-Dependent Heat Equation*, Domain Decomposition in Science and Engineering XXI (J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi, and O. B. Widlund, eds.), vol. 98, Springer-Verlag, 2014, pp. 189–198.

[52] J. E. Lagnese and G. Leugering, *Domain decomposition methods in optimal control of partial differential equations*, International Series of Numerical Mathematics (Basel), vol. 148, 2004.

[53] I. Lasiecka and R. Triggiani, *Control theory for partial differential equations, Continuous and Approximation Theories*, vol. 1, Cambridge University Press, 1999.

[54] E. Lelarasmee, A. Ruehli, and A. Sangiovanni-Vincentelli, *The waveform relaxation method for time-domain analysis of large scale integrated circuits*, IEEE Trans. Compt.-Aided Design Integr. Circuits Syst. **1** (1982), no. 3, 131–145.

[55] E. Lindelöf, *Sur l'application des méthodes d'approximations successives à l'étude des intégrales réelles des équations différentielles ordinaires*, Journal de Mathématiques Pures et Appliquées (1894).

[56] J.-L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer, Berlin, 1971.

[57] P.-L. Lions, *On the Schwarz alternating method I*, First International Symposium on Domain Decomposition Methods for PDEs (Philadelphia), 1988, pp. 1–42.

[58] ———, *On the Schwarz alternating method II*, Stochastic interpretation and order properties, in Domain decomposition methods (Philadelphia), 1989, pp. 47–70.

[59] ———, *On the Schwarz alternating method III. A variant for nonoverlapping subdomains*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia), 1990, pp. 202–223.

[60] C. Lubich and A. Ostermann, *Multigrid dynamic iteration for parabolic equations*, BIT **27** (1987), 216–234.

[61] B. C. Mandal, *Dirichlet-Neumann and Neumann-Neumann Waveform Relaxation Algorithms for the Wave Equation in Multiple subdomains*, in Preparation.

[62] ———, *A Time-Dependent Dirichlet-Neumann Method for the Heat Equation*, Domain Decomposition in Science and Engineering XXI (J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi, and O. B. Widlund, eds.), vol. 98, Springer-Verlag, 2014, pp. 467–475.

[63] J. Mandel, *Balancing Domain Decomposition*, Comm. Numer. Meth. Engrg. **9** (1993), 233–241.

[64] J. Mandel and M. Brezina, *Balancing domain decomposition for problems with large jumps in coefficients*, Math. Comp. **65** (1996), no. 216, 1387–1401.

[65] V. Martin, *An optimized Schwarz waveform relaxation method for unsteady convection diffusion equation*, Appl. Numer. Math. **52** (2005), no. 4, 401–428.

[66] L. Martini and A. Quarteroni, *An Iterative Procedure for Domain Decomposition Methods: a Finite Element Approach*, SIAM, in Domain Decomposition Methods for PDEs, I (Philadelphia) (R. Glowinski et al., ed.), 1988, pp. 129–143.

[67] ———, *A Relaxation Procedure for Domain Decomposition Method using Finite Elements*, Numer. Math. **55** (1989), 575–598.

[68] U. Miekkala and O. Nevanlinna, *Convergence of dynamic iteration methods for initial value problems*, SIAM J. Sci. Stat. Comput. **8** (1987), 459–482.

[69] F. Nataf and F. Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, Math. Models Methods Appl. Sci. **5** (1995), no. 1, 67–93.

[70] O. Nevanlinna, *Remarks on Picard-Lindelöf iterations part i*, BIT **29** (1989), 328–346.

[71] ———, *Remarks on Picard-Lindelöf iterations part ii*, BIT **29** (1989), 535–562.

[72] Fritz Oberhettinger and L. Badii, *Tables of laplace transforms*, Springer-Verlag, 1973.

[73] B. W. Ong, S. High, and F. Kwok, *Pipeline Schwarz Waveform Relaxation*, to appear in Domain Decomposition in Science and Engineering XXII, Springer-Verlag.

[74] E. Picard, *Sur l'application des méthodes d'approximations successives à l'étude de certaines équations différentielles ordinaires*, Journal de Mathématiques Pures et Appliquées (1893).

[75] H. A. Priestley, *Introduction to integration*, Oxford University Press, 1997.

[76] A. Quarteroni and A. Valli, *Domain decomposition mathods for partial differential equations*, Clarendon Press, 1999.

[77] J.-P. Raymond, *Optimal Control of Partial Differential Equations*, Lecture notes of Université Paul Sabatier and French Indian Cyber-University in Science, 2009.

[78] Y. D. Roeck and P. L. Tallec, *Analysis and Test of a local domain decomposition preconditioner*, Domain Decomposition Methods for PDEs, I (Philadelphia) (R. Glowinski et al., ed.), SIAM, 1991, pp. 112–128.

[79] J. L. Schiff, *The laplace transform*, Springer, 1991.

[80] H. A. Schwarz, *Über einen Grenzübergang durch alternierendes Verfahren*, Vierteljahrsschrift der Naturforschenden Gesellschaft Zürich **15** (1870), 272–286.

[81] P. L. Tallec, Y. D. Roeck, and M. Vidrascu, *Domain decomposition methods for large linearly elliptic three-dimensional problems*, J. of Comput. and App. Math. **34** (1991), 93–117.

[82] A. Toselli and O. B. Widlund, *Domain decomposition methods, algorithms and theory*, Springer, 2005.

[83] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Graduate Studies in Mathematics, vol. 112, Amer. Math. Soc., 2010.