UNIVERSITÉ PARIS XIII - SORBONNE PARIS NORD

École doctorale Galilée

Algorithmes de parallélisation en temps pour des problèmes de contrôle optimal de l'équation des ondes

Parallel in time algorithms for internal wave controlled problem

Thèse de doctorat

présentée par

THANH VUONG DANG

pour l'obtention du grade de

DOCTEUR EN MATHÉMATIQUES APPLIQUÉES

soutenue le 20 décembre 2023 devant le jury d'examen constitué de

DARBAS Marion, USPN, Présidente du jury
DELOURME Bérangère , USPN, co-encadrante de thèse
HALPERN Laurence, USPN, directrice de thèse,
KWOK Felix, rapporteur
MANDAL Bankim, rapporteur

Abstract

Résumé Cette thèse de doctorat porte sur des problèmes de contrôle dans lesquels les systèmes sous-jacents sont gouvernés par des équations aux dérivées partielles (EDP) hyperboliques qui modélisent des phénomènes de type onde. De tels problèmes proviennent souvent de problèmes inverses et d'assimilation de données. D'un point de vue numérique, ces problèmes sont difficiles à résoudre. L'objectif du projet ANR AlloWAPP est de développer et d'analyser des méthodes de décomposition de domaines espace/temps pour les résoudre.

Dans ce travail de thèse nous considérons le contrôle optimal en volume de l'équation d'onde, et nous avons conçu deux algorithmes de décomposition de domaine en temps : le premier est appelé *inherited* dans le sens où il utilise des conditions de transmission héritées de la fonction de coût. Le second est une nouvelle version de l'algorithme de Dirichlet-Neumann, conçu pour les équations elliptiques par Bjorstad et Widlund en 1984. Pour les deux, nous prouvons que l'algorithme est bien défini, et étudions la convergence. Nous comparons les résultats théoriques aux résultats numériques par l'application d'un schéma aux volumes finis de Gander/Halpern/Nataf en dimension 1.

Abstract This PhD thesis deals with control problems in which the underlying systems are governed by hyperbolic partial differential equations (PDEs) that model wave-type phenomena. Such problems often arise from inverse problems and data assimilation. From a numerical point of view, those problems are challenging to solve. The objective of the ANR project AlloWAPP is to develop and analyse domain decomposition methods in space and time to solve them.

In the PhD work we consider the optimal control of the wave equation, and we designed two algorithms for domain decomposition in time: the first one is called *inherited* in the sense that it uses transmission conditions inherited from the cost function. The second one is a version of the Dirichlet-Neumann algorithm designed for elliptic equations by Bjorstad and Widlund in 1986. For both we prove well-posedness and we study the convergence. We compare the theoretical results to numerical experiments, with a finite volume scheme developed in Gander/Halpern/Nataf.

Contents

Chapter 1

Introduction

This PhD thesis deal with control problems in which the underlying systems are governed by hyperbolic partial differential equations (PDEs) that model wave-type phenomena. Such problems often arise from inverse problems and data assimilation (in that field, a physical system is modeled approximately with unknown parameters and initial conditions, the objective is to use observations taken over the course of time to adjust or refine the model, so that model predictions match better with actual observations). One classic example of an inverse problem is seismic inversion [13, 77, 78] where the observed data are the seismic waves reflected by the different subsurface rock formations, and the goal is to deduce and refine the rock parameters that make the observed data possible. Another concrete example is the optimal placement of hydraulic generators [3], where a continuous stream of wave height observations are available at several near-shore locations, and the goal is to refine the wave model along the shoreline, which in turn allows different placements of hydraulic generators to be simulated in order to maximize energy production.

From a numerical point of view, those problems are challenging to solve: indeed, first, a large amount of data must be processed. Then, intensive computations are required to simulate hyperbolic phenomena to a high resolution. Finally, the optimization procedure (for instance by a gradient method) requires either to solve at each iteration in turn forward and backward equations or to solve a huge direct non triangular linear system. In order to tackle problems with such extreme computational requirements, it is essential to design scalable, highly efficient methods that can perform as many of the subtasks as possible in parallel using the large number of processors available on modern computing clusters. The objective of this work is to develop and analyse domain decomposition methods in time to solve them.

Parallel in space for PDE Let us start by giving a few references on the parallelization by domain decomposition methods in space for PDE. The most famous one might be the Schwarz method, whose were first invented by Hermann Amandus Schwarz in 1869 as an analytical tool to rigorously prove results obtained by Riemann through a minimization principle (see [88] and [33]). The pioneer work about the use of Schwarz algorithms as a numerical tool is presented in series of papers [68, 70, 71, 69] by Pierre Louis Lions. In particular, in the elliptic case, a convergent algorithm based on non overlapping subdomains is proposed: its consists in exchanging Robin traces (of the type $\partial_n u + \alpha u$) instead of the classical Dirichlet ones. Because there are infinitely many choices of α providing a convergent the algorithm, a natural question is to develop so-called optimized Schwarz methods, where the 'best' choice of transmission data is made in order to optimize the convergence rate of the method (see e.g. [54] for the first numerical experiments, [79, 61] for an extended analysis of the optimal transmission conditions, and [61, 30, 8, 55] for their extensions to different PDE and more sophisticated transmission conditions. Besides, discrete versions of the methods has been developed such as additive Schwarz (AS), multiplicative Schwarz (MS) and restrictive additive Schwarz (RAS)(see [26, 84, 23]). The method appears also powerful to construct preconditioners [10]: to be more specific, iterative Schwarz methods corresponds to solve an interface problem (instead of a volumic problems) by a Jacobi algorithm. It turns out that the interface problem usually have a better conditioning than its volumic counterpart. Then, the current procedure consists in solving the interface problems with fast iterative solvers such that conjugate gradient or GMRES. The strategy can also be applied to non linear equations, see [22]. For multi-subdomains, the standard algorithms are generally not as performant as in the two domains case, since several iterations are required to 'transport' the information between distant subdomains. The problems are known to be non-scalable ([12]). To overcome this difficulty coarse corrections are proposed: see [38, 27, 25, 24, 12]. For the time dependent problems (and, in particular wave equation), Schwarz waveform relaxations algorithms has been developed in [36], where a space-time interface condition is used (see also [10, 63]). Besides Schwarz method, the Schur complement methods (or substructured solvers) [23] such as Dirichlet-Neumann and Neumann/Neumann algorithms have been widely used. In that case Dirichlet or Neumann data are in turn exchanged through the interfaces of the subdomains (see [89]). They required a relaxation parameter to converge. Such method can be used also for time dependent problems [40, 76, 83, 87] or for fourth order equations [47, 48]. We point out that wave-form algorithms can be additionally parallelized by using the pipeline-strategy introduced in [63, 82]. The main idea is to compute simultaneously Schwarz waveform iterations, introducing then a natural time parallelization. It also has been applied to Neumann-Neumann algorithm, for parabolic and hyperbolic equations [41, 83].

Parallel in space methods for optimal control problems Naturally, the previous domain decomposition methods has been applied to the solve optimal control problems. In a nutshell, the goal of such control problems is to minimize a cost functional $J(u, y(u))$ (often quadratic in u and y) where u is a an admissible control which acts on $y(u)$ the solution to a partial differential equation (elliptic, parabolic or hyperbolic). in [6], [5], a Schwarz algorithm for the optimal control of an elliptic PDE is proposed. The work is extended to parabolic and hyperbolic equations in [7]. In those references, the Schwarz method is applied to the optimality system (Euler equations associated with the optimization problem). This problem is made of a system equations. transmission conditions exchanged between the subdomains coupled primal and dual variable. Convergence property of the algorithm is proven in general by energy estimation. In the elliptic case, Neumann-Neumann algorithms and Dirichlet Neumann algorithms are proposed (and analysed) in [58, 42] while an analysis of optimized Schwarz is given in [20]. In [57], a non overlapping Schwarz algorithm is proposed for the optimal control of parabolic problems. Each iteration requires to solve optimal controls problems in each subdomains. A proof of convergence for the parabolic case (boundary control) may be seen in [72] and in [14] for the periodic (in time) case. As for the optimal control of the dissipative wave equation, a non overlapping Schwarz method is given and analysed in [64][Chapter 6]. Recently, non linear elliptic control problems are tackled in [15] and 'economic' control problem (with mixed constraint) may be found in [16].

Parallel in time methods for time dependent PDE With the continuing increase of core counts in modern clusters, we have reached the spatial saturation point, where the subproblems in space become so small that communication and iteration costs start to dominate, and no more speedup can be obtained. Thus, the sequentiality of the simulation in time becomes the bottleneck, and it has become more and more urgent to develop and analyze methods that also allow parallelization in the time direction, particularly for computationally intensive applications such as data assimilation. Parallel-in-time methods have received much interest in the past decade in the high performance computing community, see the workshop series on parallel-in-time methods on parallel-in-time.org. We point out that the sequential nature of time dependent problems make their parallelization less natural. In particular the performance of the method are often sensitive to the nature the equation (parabolic or hyperbolic) An overview of such methods can be found in [32]. Let us mention a few of them. A first common technique is based on shooting methods: the time interval of simulation $[0, T]$ is split into N smaller intervals $[T_i, T_{i+1}], i \in [\![1, N]\!]$, and the continuity continuity conditions of the solution of the PDE at each interface T_i are iteratively enforced. For instance, the famous *parareal* algorithm [75] solves the interface equation (ensuring the continuity at each T_i) using a 'approximate' Newton method (see [35]). This algorithm is known to be very powerful for parabolic PDE but leads to slow convergence for hyperbolic equations ([31, 86], see also [18] and [81] for possible cures). Then, multigrid methods have also been widely studied [53, 74, 29, 50, 59]. Broadly speaking, those methods use several grids of different size, the algorithms then alternating between 'smoothing' steps (made on a fine grids) and a residual correction (on a coarser grid). We note that the parareal algorithm can also be interpreted as a multigrid method in time, with aggressive coarsening and slightly special smoother [51, 29, 45]. In that class, we also mention the algorithm *PFASST* (parallel full approximation scheme in space and time), introduced by [28], which is based on a deferred correction method performed on different grids. Beside, a collection of methods appears specially interesting for the parallelization of hyperbolic equations: *Paraexp* [34] (Exact computation of matrix exponential), diagonalization techniques [46, 19, 37]. A recent unified analysis of the different strategies to parallelize in time might be found in [49].

Parallel in time methods for optimal control problems We end this bibliographical part by given a few references on parallel in time algorithms for control problems. In the well-known paper of J.E.Lagnese and G. Leugering in 2002 [65], the authors consider an optimal control problem of the wave equation. The solution w of the homogeneous wave equation is driven by a control v imposed through the dissipative boundary condition

$$\partial_n w + \alpha \partial_t w = v.$$

The authors minimize the quadratic cost function

$$J(v) = \frac{1}{2} \int_{\Sigma^N} |v|^2 d\Sigma + \frac{k}{2} \left\| (w(T), \frac{\partial}{\partial t} w(T)) - (z_0, z_1) \right\|_{\mathcal{H}}^2$$

As in [6], the exchange data between the subdomains in times are linear combinations of the primal and the dual variables. They prove the convergence of the algorithm for a relaxed algorithm. In 2003, Matthias Heinkenschloss [56] proposes to solve in parallel (using Gauss-Seidel and Jacobi methods) the interface in time problem obtained by decomposing the Euler system associated with an abstract internal optimal control problem. In [62], F. Kwok and M. Gander introduce an overlapping Schwarz in time algorithm to solve an optimal control governed by a system of O.D.E. They provide explicit expression of the corresponding convergence factor. The extension to non-overlapping optimized Schwarz methods can be found in [39], wherein a convergence proof is given (under appropriate assumptions) as well as an optimization of the convergence factor. This approach has also been studied for the parabolic control problem discretized in space by Galerkin method [92]. Besides Schwarz-like in time algorithms, other parallelization approaches have been investigated. Without being exhaustive, let us mention a few of them. In [90], S.Ulbrich proposes a new SQP (sequential quadratic programming) convergent approach for solving the interface in time problem. Its method uses parareal algorithms to solve backward and forward equation at each iteration. This method is extended to the construction of preconditioners in [91]. In [21], a new parallel in time gradient method is introduced: during the optimisation process, the computation of the gradient of the functional, which requires to solve a forward and a backward equation, is made in parallel. Space-and time parallel gradient method, based on the use of PFASST is given in [52]. Then, [4] introduce an additive Schwarz preconditioner for solving the Euler system. Indirect shooting methods applied to non linear control of parabolic equation can be found in [11]. Finally, an extension of the parareal algorithm to optimal control problems, named ParaOpt is proposed in [44]. The convergence is proved for parabolic problems.

Summary of the PhD thesis The aim of this thesis is to study domain decomposition in time algorithm to solve a particular optimal control problem. Let us first consider the functional

$$J(v,y) = \frac{1}{2}\int_{\Omega\times[0,T]} |y(x,t)-\widehat{y}(x,t)|^2 dxdt + \frac{\gamma}{2}\int_{\Omega} |y(x,T)-\widehat{z}(x)|^2 dx + \frac{\alpha}{2}\int_{\Omega\times[0,T]} |v(x,t)|^2 dxdt, \quad (1.0.1)$$

where $\gamma \geq 0$ and $\alpha > 0$ are parameters and, $\widehat{y} \in L^2(\Omega\times(0,T))$ and $\widehat{z} \in L^2(\Omega)$ are target functions.

We shall minimize the functional

$$\widehat{J}(v) = J(v, y(v)),$$

over all the controls $v \in L^2(\Omega \times (0,T))$. The function $y(v)$ is the solution of the following wave problem:

$$\begin{cases} \partial_{tt} y - \Delta y = v & (x,t) \in \Omega \times [0,T], \\ y(x,t) = 0, & x \in \partial\Omega, t \in [0,T], \\ y(x,0) = y^{(0)}(x), y_t(x,0) = y^1(x), & x \in \Omega, \end{cases}$$

and $(y^{(0)}(x), y^{(1)}(x)) \in H_0^1(\Omega) \times L^2(\Omega)$.

Remark 1. *We emphasize that in this PhD thesis, we only consider the optimal control of the wave equation. This choice is motivated by the fact that the exact control problem of the wave equation (wherein we impose $y(T,x) = \hat{z}$) is numerically difficult to solve [93]: because the high frequencies are not uniformly controllable with respect to the discretization parameter, the discrete control diverges and blows up when refining the mesh. Fortunately, there are some cure methods to eliminate the oscillation of discrete control such as Tychonoff regularization, mixed finite elements, two-grid scheme in time or Fourier filtering. The research about those methods has became active, see e.g. [2, 60, 66, 80, 73, 1, 17].*

Let us now summarize the content of the manuscript. In Chapter 2, we prove that the minimization problem is well-posed (using standard tools of convex optimization) and we write its associated Euler system. Then, we explain how to discretize it using a finite volume method. In particular, we show that the discretized problem is also well-posed.

Chapter 3 is dedicated to the construction and analysis of the so-called 'Inherited algorithm', a time domain domain decomposition method based on an optimized Schwarz algorithm. Using Fourier series in space and trigonometric matrices, we write the corresponding iteration matrix. We analyse it first in the overlapping case with infinite subdomains in time. Then on finite time domains, in the non overlapping case, choosing $\gamma = 0$ and $\alpha = 1$. We prove that the inherited algorithm converges if T is sufficiently small and diverged for large T. We conclude this part by presenting numerical results that illustrate our theoretical study.

Finally, Chapter 4 presents two alternative algorithms for the time parallelization. The first one is the relaxed variant of the inherited algorithm. We prove that there always exists a relaxation parameter for which the relaxed algorithm converges, regardless of the value of T. Then, we optimized the relaxation parameter in order to minimize the convergence factor. The second algorithm is a Dirichlet Neumann in time algorithm. Here again, we construct the corresponding iteration matrix. In particular, we prove that in the case $T = +\infty$, the algorithm converges into two iterations.

Chapter 2

Presentation of problem, minimisation, existence and uniqueness

2.1 Definition and well-posedness

Let Ω be a bounded domain in $\mathbb{R}^d$, $T > 0$. The cost function is defined by

$$J(v, y) = \frac{1}{2}\int_{\Omega\times(0,T)} |y(x,t) - \widehat{y}(x,t)|^2 dxdt + \frac{\gamma}{2}\int_{\Omega} |y(x,T) - \widehat{z}(x)|^2 dx + \frac{\alpha}{2}\int_{\Omega\times(0,T)} |v(x,t)|^2 dxdt, \tag{2.1.1}$$

$\widehat{y}$, $\widehat{z}$ are target, α and γ are the coefficients. Define $y = \mathcal{Y}(v; y^{(0)}, y^{(1)})$ the linear function of v and the initial data, given by

$$\begin{cases} \partial_{tt} y - \Delta y = v & \text{in } \Omega \times (0,T), \\ y = 0 & \text{in } \partial\Omega \times (0,T), \\ y(\cdot,0) = y^{(0)},\ \partial_t y(\cdot,0) = y^{(1)}, & \text{in } \Omega. \end{cases} \tag{2.1.2}$$

The initial data are $y^{(0)} \in H_0^1(\Omega)$ and $y^{(1)} \in L^2(\Omega)$. From [67] follows that for any $v \in L^2(0,T,L^2(\Omega))$, the problem (2.1.2) has a unique solution. The mapping

$$(v; y^{(0)}, y^{(1)}) \to (y, \frac{dy}{dt})$$

is a linear continuous map of $L^2(0,T,L^2(\Omega)) \times H_0^1(\Omega) \times L^2(\Omega)$ to $L^2(0,T,H_0^1(\Omega)) \times L^2(0,T,L^2(\Omega))$.

The minimization problem is to find a global minimum in $L^2(\Omega \times (0,T))$ of the function

$$\widehat{J}(v) = J(v, \mathcal{Y}(v; y^{(0)}, y^{(1)})). \tag{2.1.3}$$

Theorem 2.1.1. *The function $\widehat{J}$ has a unique global minimum $u \in L^2(0,T,L^2(\Omega))$, characterized by the Euler equation $\widehat{J}'(u) = 0$.*

Proof. The function $\widehat{J}$ is a quadratic function of v, twice differentiable, its first derivative is calculated easily by

$$\widehat{J}'(v) \cdot w = \lim_{h\to 0} \frac{\widehat{J}(v + hw) - \widehat{J}(v)}{h} = \int_{\Omega\times(0,T)} (y(x,t) - \widehat{y}(x,t))\tilde{y}(x,t) dxdt$$
$$+ \gamma \int_{\Omega} (y(x,T) - \widehat{z}(x))\tilde{y}(x,T) dx + \alpha \int_{\Omega\times(0,T)} v(x,t) w(x,t) dxdt, \tag{2.1.4}$$

where $\tilde{y} = \mathcal{Y}(w; 0, 0)$ is the solution of

$$\begin{cases} \partial_{tt}\tilde{y} - \Delta\tilde{y} = w, & \text{in } \Omega \times (0,T), \\ \tilde{y} = 0, & \text{in } \partial\Omega \times (0,T), \\ \tilde{y}(\cdot,0) = \partial_t\tilde{y}(\cdot,0) = 0, & \text{in } \Omega. \end{cases} \tag{2.1.5}$$

The second derivative is given by

$$\widehat{J}''(v) \cdot w_1 \cdot w_2 = \lim_{h\to 0} \frac{\widehat{J}'(v + hw_2) \cdot w_1 - \widehat{J}'(v) \cdot w_1}{h}$$
$$= \int_{\Omega\times(0,T)} \tilde{y}_1(x,t)\tilde{y}_2(x,t)dxdt + \gamma \int_\Omega \tilde{y}_1(x,T)\tilde{y}_2(x,T)dx + \alpha \int_{\Omega\times(0,T)} w_1(x,t)w_2(x,t)dxdt, \tag{2.1.6}$$

where $\tilde{y}_j = \mathcal{Y}(w_j; 0, 0)$ is the solution of

$$\begin{cases} \partial_{tt}\tilde{y}_j - \Delta\tilde{y}_j = w_j, & \text{in } \Omega \times (0,T), \\ \tilde{y}_j = 0, & \text{in } \partial\Omega \times (0,T), \\ \tilde{y}_j(\cdot,0) = \partial_t\tilde{y}_j(\cdot,0) = 0, & \text{in } \Omega. \end{cases} \tag{2.1.7}$$

From (2.1.6), we infer that for any $(v,\, w)$, $\widehat{J}''(v)\cdot w\cdot w \geq \alpha\|w\|^2_{L^2(\Omega\times(0,T))}$, and therefore $\widehat{J}$ is $\alpha-$convex. Standard theorems in convex analysis in [85] apply to conclude that there exists a unique minimum point, u, characterized by the Euler equation $\widehat{J}'(u) = 0$. □

The next theorem identifies $\widehat{J}'(u)$ and sets a system, called the optimality system, defining u.

Theorem 2.1.2. *The global minimizer u of $\widehat{J}$ satisfies the following system of partial differential equations, which is therefore well-posed.*

$$\widehat{J}'(u) := \alpha u - \lambda = 0 \ \textit{in } \Omega \times (0,T), \ \textit{with} \tag{2.1.8a}$$

$$\begin{cases} \partial_{tt}y - \Delta y = u & \textit{in } \Omega \times (0,T), \\ y = 0 & \textit{on } \partial\Omega \times (0,T), \\ y(\cdot,0) = y^{(0)},\ \partial_t y(\cdot,0) = y^{(1)} & \textit{in } \Omega. \end{cases} \tag{2.1.8b}$$

$$\begin{cases} \partial_{tt}\lambda - \Delta\lambda + y - \widehat{y} = 0 & \textit{in } \Omega \times (0,T), \\ \lambda = 0 & \textit{on } \partial\Omega \times (0,T), \\ \lambda(.,T) = 0,\ \partial_t\lambda(.,T) - \gamma(y(.,T) - \widehat{z}) = 0 & \textit{in } \Omega. \end{cases} \tag{2.1.8c}$$

Remark 2. *The direct problem (2.1.8b) for the state y has initial data $(y^{(0)}, y^{(1)})$ which are given by the problem, and a source term u which is the goal of the problem. The backward problem (2.1.8c) for the adjoint λ has a source term and final data (which are of mixed type) involving the state y. It is well-posed for similar reasons as the direct problem. The relation (2.1.8a) is the Euler equation of the minimization of $\widehat{J}$.*

Proof. In order to identify $\widehat{J}'(v)$, we use the expression (2.1.4), rewritten in a more compact way

$$\widehat{J}'(v)\cdot w = (y-\widehat{y}, \tilde{y})_{L^2(\Omega\times(0,T))} + \gamma(y(\cdot,T)-\widehat{z}, \tilde{y}(\cdot,T))_{L^2(\Omega)} + \alpha(v,w)_{L^2(\Omega\times(0,T))}. \tag{2.1.9}$$

where y is the solution of (2.1.8b) with u replaced by v, and $\tilde{y}$ is the solution of (2.1.5). Choose now λ solution of (2.1.8c). Multiply the volume equation by $\tilde{y}$, and integrate twice by parts in time and space. Compute the terms separately,

$$\int_0^T \int_\Omega (\Delta\tilde{y}, \lambda)\, dtdx = \int_0^T \int_\Omega (\Delta\lambda, \tilde{y})\, dtdx,$$

since both $\tilde{y}$ and λ vanish on $\partial\Omega$. As for the time derivative,

$$\int_0^T (\partial_{tt}\tilde{y}, \lambda)_{L^2(\Omega)}\, dt = \int_0^T (\partial_{tt}\lambda, \tilde{y})_{L^2(\Omega)}\, dt + (\partial_t\tilde{y}(\cdot,T), \lambda(\cdot,T))_{L^2(\Omega)} - (\tilde{y}(\cdot,T), \partial_t\lambda(\cdot,T))_{L^2(\Omega)},$$

since $\tilde{y}$ and $\partial_t\tilde{y}$ vanish for $t=0$. Grouping, we obtain

$$(\partial_{tt}\lambda - \Delta\lambda, \tilde{y}) = (\partial_{tt}\tilde{y} - \Delta\tilde{y}, \lambda) - (\partial_t\tilde{y}(\cdot,T), \lambda(\cdot,T))_{L^2(\Omega)} + (\tilde{y}(\cdot,T), \partial_t\lambda(\cdot,T))_{L^2(\Omega)}.$$

Replace in the equation to obtain

$$(\hat{y} - y, \tilde{y}) = (w, \lambda) - (\partial_t\tilde{y}(\cdot,T), \lambda(\cdot,T))_{L^2(\Omega)} + (\tilde{y}(\cdot,T), \partial_t\lambda(\cdot,T))_{L^2(\Omega)}.$$

Plug it into (2.1.9) to obtain

$$\begin{aligned}
\widehat{J}'(v)\cdot w &= -(w,\lambda)_{L^2(\Omega\times(0,T))} + (\partial_t\tilde{y}(\cdot,T), \lambda(\cdot,T))_{L^2(\Omega)} - (\tilde{y}(\cdot,T), \partial_t\lambda(\cdot,T))_{L^2(\Omega)} \\
&\quad + \gamma(y(\cdot,T)-\widehat{z}, \tilde{y}(\cdot,T))_{L^2(\Omega)} + \alpha(v,w)_{L^2(\Omega\times(0,T))} \\
&= (\alpha v - \lambda, w)_{L^2(\Omega\times(0,T))} + (\gamma(y(\cdot,T)-\widehat{z}) - \partial_t\lambda(\cdot,T), \tilde{y}(\cdot,T))_{L^2(\Omega)} \\
&\quad + (\partial_t\tilde{y}(\cdot,T), \lambda(\cdot,T))_{L^2(\Omega)} \\
&= (\alpha v - \lambda, w)_{L^2(\Omega\times(0,T))}, \quad \text{due to the conditions on } \lambda \text{ at } t=T,
\end{aligned}$$

which shows that $\widehat{J}'(v) = \alpha v - \lambda$. Now the unique minimum point of the quadratic function $\widehat{J}$ is characterized by the Euler equation $\widehat{J}'(u) = 0$, which shows that (2.1.8) is well-posed and defines u. □

2.2 Discretization of the optimality system in one dimension

Here $\Omega = (a, b)$. We replace u by $\frac{1}{\alpha}\lambda$ in the equation for y. We use the finite volumes in time and space developed for the direct problem in [36]. We recall here the definitions. There are $J+2$ points in space numbered from 0 up to $J+1$ and $\Delta x = (b-a)/(J+1)$ and $K+1$ grid points in time with $\Delta t = T/K$ numbered from 0 up to K. We denote the numerical approximation to $y(a+j\Delta x, k\Delta t)$ on Ω at iteration step k by $Y(j,k)$. Similarly the numerical approximation to $\lambda(a+j\Delta x, k\Delta t)$ on Ω at iteration step k is denoted by $\Lambda(j,k)$. Most of this section is integrally taken in [36] for clarity. Adaptation to the optimality problem (discretization on the final line) is original.

2.2.1 Discretization of the Problem from [36]

Interior Points

Denoting by D the volume around a grid point ($x = a + j\Delta x, t = k\Delta t$) in the interior of subdomain $\Omega \times (0, T)$, we obtain the finite volume scheme by integration of the equation over the volume D and application of the divergence theorem, see Figure 2.1, on the left

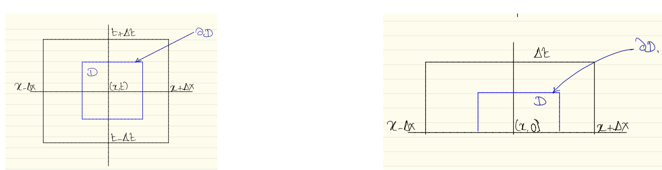


Figure 2.1: Control volume of an interior grid point and on the initial line

$$
\begin{aligned}
0 &= \int_{x-\Delta x/2}^{x+\Delta x/2} \int_{t-\Delta t/2}^{t+\Delta t/2} \left[\frac{\partial^2 y}{\partial t^2}(\xi,\tau) - \frac{\partial^2 y}{\partial x^2}(\xi,\tau) \right] d\tau d\xi \\
&= \int_{x-\Delta x/2}^{x+\Delta x/2} \frac{\partial y}{\partial t}(\xi, t+\Delta t/2) d\xi - \int_{x-\Delta x/2}^{x+\Delta x/2} \frac{\partial y}{\partial t}(\xi, t-\Delta t/2) d\xi \\
&\quad - \int_{t-\Delta t/2}^{t+\Delta t/2} \frac{\partial y}{\partial x}(x+\Delta x/2, \tau) d\tau + \int_{t-\Delta t/2}^{t+\Delta t/2} \frac{\partial y}{\partial x}(x-\Delta x/2, \tau) d\tau.
\end{aligned}
\tag{2.2.1}
$$

Now we approximate the remaining derivatives by finite differences on the grid,

$$\begin{array}{llll} D_t^+(Y)(j,k) := \frac{Y(j,k+1)-Y(j,k)}{\Delta t} & \approx \frac{\partial y}{\partial t}(\xi, t+\Delta t/2), & & \\ D_t^-(Y)(j,k) := \frac{Y(j,k)-Y(j,k-1)}{\Delta t} & \approx \frac{\partial y}{\partial t}(\xi, t-\Delta t/2), & x-\frac{\Delta x}{2} \le \xi \le x+\frac{\Delta x}{2}, & \\ D_x^+(Y)(j,k) := \frac{Y(j+1,k)-Y(j,k)}{\Delta x} & \approx \frac{\partial y}{\partial x}(x+\Delta x/2, \tau), & & \\ D_x^-(Y)(j,k) := \frac{Y(j,k)-Y(j-1,k)}{\Delta x} & \approx \frac{\partial y}{\partial x}(x-\Delta x/2, \tau), & t-\frac{\Delta t}{2} \le \tau \le t+\frac{\Delta t}{2}. & \end{array} \tag{2.2.2}$$

We thus obtain from (2.2.1) the discrete quantity

$$\Delta x(D_t^+ - D_t^-)(Y)(j,k) - \Delta t(D_x^+ - D_x^-)(Y)(j,k) = \Delta t \Delta x U(j,k).$$

which yields on using the identities $\Delta t D_t^+ D_t^- = D_t^+ - D_t^-$ and $\Delta x D_x^+ D_x^- = D_x^+ - D_x^-$ the well known finite difference scheme, for $1 \le j \le J$, $1 \le k \le K$.

$$\left(D_t^+ D_t^- - D_x^+ D_x^-\right)(Y)(j,k) = U(j,k) = \frac{1}{\alpha}\Lambda(j,k), \quad 1 \le j \le J, \tag{2.2.3}$$

for points in the interior of the domain .Similarly

$$\left(D_t^+ D_t^- - D_x^+ D_x^-\right)(\Lambda)(j,k) + Y(j,k) - \hat{Y}(j,k) = 0. \tag{2.2.4}$$

Points on the Initial Line

Remember that $t = 0$ has initial data for y, while it is the final time for λ. Suppose $(x = a + j\Delta x, 0)$ is a grid point on the interior of the initial line of $\Omega \times (0,T)$. We have half a volume D to integrate over, see Figure 2.1, on the right. Integrating as before for y we obtain

$$\begin{aligned} \int_D u(\xi,\tau)\, d\xi\, d\tau \quad = \quad & \int_{x-\Delta x/2}^{x+\Delta x/2} \frac{\partial y}{\partial t}(\xi, \Delta t/2) d\xi - \int_{x-\Delta x/2}^{x+\Delta x/2} \frac{\partial y}{\partial t}(\xi, 0) d\xi \\ & - \int_0^{\Delta t/2} \frac{\partial y}{\partial x}(x+\Delta x/2, \tau) d\tau + \int_0^{\Delta t/2} \frac{\partial y}{\partial x}(x-\Delta x/2, \tau) d\tau. \end{aligned}$$

Now the remaining derivatives on the right hand side can be approximated by finite differences (2.2.2), except $\frac{\partial y}{\partial t}(\xi, 0)$. But this derivative is given explicitly by the initial condition, and approximated on one grid cell by $\Delta x Y^{(1)}(j) := \Delta x y^{(1)}(a + j\Delta x)$. We thus obtain

$$\Delta x(D_t^+(Y)(j,0) - Y^{(1)}(j)) - \frac{\Delta t}{2}(D_x^+ - D_x^-)(Y)(j,0) = \frac{\Delta t \Delta x}{2} U(j,0).$$

and the scheme

$$\left(D_t^+ - \frac{\Delta t}{2} D_x^+ D_x^-\right)(Y)(j,0) - Y^{(1)}(j) = \frac{\Delta t}{2} U(j,0). \tag{2.2.5}$$

Points on the Final Line

Suppose $(x = a + j\Delta x, 0)$ is a grid point on the interior of the final line of $\Omega \times (0,T)$. We have half a volume D to integrate over, see Figure 2.2.

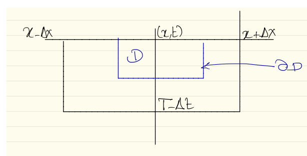


Figure 2.2: Control volume of a grid point on the final line

Integrating as before for λ we obtain

$$\begin{aligned}
-\int_D (y-\hat{y})(\xi,\tau)\, d\xi\, d\tau \quad = \quad & \int_{x-\Delta x/2}^{x+\Delta x/2} \frac{\partial \lambda}{\partial t}(\xi,T) d\xi - \int_{x-\Delta x/2}^{x+\Delta x/2} \frac{\partial \lambda}{\partial t}(\xi, T-\Delta t/2) d\xi \\
& - \int_{T-\Delta t/2}^{T} \frac{\partial \lambda}{\partial x}(x+\Delta x/2, \tau) d\tau + \int_{T-\Delta t/2}^{T} \frac{\partial \lambda}{\partial x}(x-\Delta x/2,\tau) d\tau.
\end{aligned}$$

Now the remaining derivatives on the right hand side can be approximated by finite differences (2.2.2), except $\frac{\partial \lambda}{\partial t}(\xi,T)$. But this derivative is given explicitly by the final condition and can be replaced by

$$\int_{x-\Delta x/2}^{x+\Delta x/2} \frac{\partial \lambda}{\partial t}(\xi,T) d\xi = \gamma \int_{x-\Delta x/2}^{x+\Delta x/2} (y(\xi,T) - \hat{z}(\xi)) d\xi.$$

Which gives the scheme

$$\gamma \Delta x (Y(j,K) - \hat{Z}(j)) - \Delta x D_t^-(\Lambda)(j,K) - \frac{\Delta t}{2}(D_x^+ - D_x^-)(\Lambda)(j,K) + \frac{\Delta t \Delta x}{2}(Y - \hat{Y})(j,K) = 0,$$

defining $\Delta x \hat{Z}(j) = \int_{x-\Delta x/2}^{x+\Delta x/2} \hat{z}(\xi) d\xi$. Dividing by Δx, we obtain

$$\gamma(Y(j,K) - \hat{Z}(j)) - D_t^-(\Lambda)(j,K) - \frac{\Delta t}{2}(D_x^+ D_x^-)(\Lambda)(j,K) + \frac{\Delta t}{2}(Y - \hat{Y})(j,K) = 0. \tag{2.2.6}$$

So the discretized optimality condition are, with $Y^{(0)}(j) := y^0(x_j)$ and $Y^{(1)}(j) := y^{(1)}(x_j)$

$$U(j,k) = \frac{1}{\alpha}\Lambda(j,k), 0 \le j \le J+1,\ 0 \le k \le K. \tag{2.2.7a}$$

$$\begin{cases} \left(D_t^+ D_t^- - D_x^+ D_x^-\right)(Y)(j,k) = U(j,k) & 1 \le j \le J, 1 \le k \le K. \\ Y(0,k) = Y(J+1,k) = 0 & 0 \le k \le K, \\ Y(j,0) = Y^{(0)}(j) & 1 \le j \le J \\ \left(D_t^+ - \frac{\Delta t}{2} D_x^+ D_x^-\right)(Y)(j,0) - Y^{(1)}(j) = \frac{\Delta t}{2} U(j,0). & 1 \le j \le J \end{cases} \tag{2.2.7b}$$

$$\begin{cases} \left(D_t^+ D_t^- - D_x^+ D_x^-\right)(\Lambda)(j,k) + Y(j,k) - \hat{Y}(j,k) = 0. & 1 \le j \le J, 1 \le k \le K. \\ \Lambda(0,k) = \Lambda(J+1,k) = 0 & 0 \le k \le K, \\ \Lambda(j,K) = 0 & 1 \le j \le J \\ (D_t^- + \frac{\Delta t}{2} D_x^+ D_x^-)(\Lambda)(j,K) - \gamma(Y(j,K) - \hat{Z}(j)) - \frac{\Delta t}{2}(Y - \hat{Y})(j,K) = 0. & 1 \le j \le J \end{cases} \tag{2.2.7c}$$

Equations (2.2.7b,2.2.7c) are separately well defined by the analysis in [36]. Note that the initial condition of the backward problem for Λ is a bit different, but the analysis by energy estimates extends.

Though discretization of the optimality conditions, there is no proof so far that this system is well-posed. The unknowns of the system are the $(Y(j,k), \Lambda(j,k))$ for $1 \le j \le J$, $0 \le k \le K$, which are $2(J \times K + 1)$ unknowns. The system is squared. Introduce the discrete functional obtained by the trapezoidal rule from the cost function J. Define the discrete scalar product on $\mathbb{R}^{K+1}$ by

$$(V,W)_K = \Delta t \left(\frac{1}{2} V(0)W(0) + \sum_{k=1}^{K-1} V(k)W(k) + \frac{1}{2} V(K)W(K) \right).$$

Since all functions vanish at endpoints in x, the discrete scalar product in x is

$$(V,W)_J = \Delta x \sum_{j=1}^{J} V(j)W(j).$$

The norms are defined accordingly. Then the discrete cost function is easily defined by

$$J_d(V,Y) = \frac{1}{2}\|Y - \widehat{Y}\|^2_{J,K} + \frac{\gamma}{2}\|Y(\cdot,K) - \widehat{Z}\|^2_J + \frac{\alpha}{2}\|U\|^2_{J,K}. \tag{2.2.8}$$

and the discrete targets are obtained by $\widehat{Y}(j,k) = \widehat{y}(x_j,t_k)$ and $\widehat{Z}(j) = \widehat{z}(x_j)$. Define now $\mathcal{Y}(V,Y^{(0)},Y^{(1)})$ to be the unique solution of (2.2.7b), see [36], and

$$\hat{J}_d(V) = J_d(V,\mathcal{Y}(V,Y^{(0)},Y^{(1)})) \tag{2.2.9}$$

Theorem 2.2.1. *The discrete functional $\hat{J}_d$ is twice differentiable, strictly convex and therefore has a unique global minimum, defined by the Euler equation $\hat{J}_d'(U) = 0$. Its gradient is defined by*

$$\hat{J}_d'(V) = \alpha V - \Lambda, \ \textit{with } Y = \mathcal{Y}(V,Y^{(0)},Y^{(1)}) \textit{ and } \Lambda \textit{ defined by} \tag{2.2.10a}$$

$$\begin{cases} \left(D_t^+D_t^- - D_x^+D_x^-\right)(\Lambda)(j,k) + Y(j,k) - \hat{Y}(j,k) = 0. & 1 \le j \le J, 1 \le k \le K. \\ \Lambda(0,k) = \Lambda(J+1,k) = 0 & 0 \le k \le K, \\ \Lambda(j,K) = 0 & 1 \le j \le J \\ (D_t^- + \frac{\Delta t}{2}D_x^+D_x^-)(\Lambda)(j,K) - \gamma(Y(j,K) - \hat{Z}(j)) - \frac{\Delta t}{2}(Y-\hat{Y})(j,K) = 0. & 1 \le j \le J \end{cases} \tag{2.2.10b}$$

Thus $\hat{J}_d'(u) = 0$ is equivalent to the optimality conditions (2.2.7), which are therefore well-posed.

Remark 3. *In short, this proves that the optimality conditions of the discrete functional are the discretized optimality conditions.*

Proof. Parallel to the continuous case,

$$\widehat{J}_d'(V)\cdot W = (Y - \widehat{Y}, \tilde{Y})_{J,K} + \gamma(Y(\cdot,K) - \widehat{Z}, \tilde{Y}(\cdot,K))_J + \alpha(V,W)_{J,K}. \tag{2.2.11}$$

where $\tilde{Y} = \mathcal{Y}(W;0,0)$ is the solution of

$$\begin{cases} \left(D_t^+D_t^- - D_x^+D_x^-\right)(\tilde{Y})(j,k) = W(j,k) & 1 \le j \le J, \ 1 \le k \le K. \\ \tilde{Y}(0,k) = \tilde{Y}(J+1,k) = 0 & 0 \le k \le K, \\ \tilde{Y}(j,0) = 0 & 1 \le j \le J \\ D_t^+\tilde{Y}(j,0) = \frac{\Delta t}{2}W(j,0). & 1 \le j \le J. \end{cases} \tag{2.2.12}$$

The second derivative is

$$\widehat{J}_d''(V)\cdot W_1 \cdot W_2 = (\tilde{Y}_1, \tilde{Y}_2)_{J,K} + \gamma(\tilde{Y}_1(\cdot,K), \tilde{Y}_2(\cdot,K))_J + \alpha(W_1,W_2)_{J,K}. \tag{2.2.13}$$

and therefore

$$\widehat{J}_d''(V)\cdot W\cdot W \geq \alpha\|W\|_{J,K}^2$$

Therefore $\widehat{J}_d$ is α convex, has a unique global minimum, characterized by $\widehat{J}_d'(U)=0$. To further understand that, define Λ from Y by

$$\begin{cases}\left(D_t^+D_t^- - D_x^+D_x^-\right)(\Lambda)(j,k)+Y(j,k)-\hat{Y}(j,k)=0. & 1\leq j\leq J, 1\leq k\leq K.\\ \Lambda(0,k)=\Lambda(J+1,k)=0 & 0\leq k\leq K,\\ \Lambda(j,K)=0 & 1\leq j\leq J\end{cases} \tag{2.2.14}$$

Compute the first term in (2.2.11)

$$(Y-\widehat{Y},\tilde{Y})_{J,K}=\frac{\Delta t}{2}(Y(\cdot,K)-\widehat{Y}(\cdot,K),\tilde{Y}(\cdot,K))_J+\frac{\Delta t}{2}(Y(\cdot,0)-\widehat{Y}(\cdot,0),\tilde{Y}(\cdot,0))_J$$
$$+\sum_{k=1}^{K-1}(Y(\cdot,k)-\widehat{Y}(\cdot,k),\tilde{Y}(\cdot,k))_J.$$

Replace for $1\leq k\leq K-1$, $Y(j,k)-\widehat{Y}(j,k)$ using the first equation in (2.2.14).

$$(Y-\widehat{Y},\tilde{Y})_{J,K}=\frac{\Delta t}{2}(Y(\cdot,K)-\widehat{Y}(\cdot,K),\tilde{Y}(\cdot,K))_J+\frac{\Delta t}{2}(Y(\cdot,0)-\widehat{Y}(\cdot,0),\tilde{Y}(\cdot,0))_J$$
$$-\sum_{k=1}^{K-1}\Big((D_t^+D_t^- - D_x^+D_x^-)(\Lambda)(\cdot,k),\tilde{Y}(\cdot,k)\Big)_J.$$

Due to the boundary conditions in space, for each k,

$$\Big(D_x^+D_x^-(\Lambda)(\cdot,k),\tilde{Y}(\cdot,k)\Big)_J=\Big(D_x^+D_x^-(\tilde{Y})(\cdot,k),\Lambda(\cdot,k)\Big)_J$$

Consider now the derivatives in time, and write a discrete integration by parts formula. For sake of readability, since the space does not intervene here, the index j in space is omitted all through.

$$\begin{aligned}
\Delta t \sum_{k=1}^{K-1} D_t^+ D_t^-(\Lambda)(k)\, \tilde{Y}(k) &= \frac{1}{\Delta t}\sum_{k=1}^{K-1} (\Lambda(k+1) - 2\Lambda(k) + \Lambda(k-1))\tilde{Y}(k) \\
&= \frac{1}{\Delta t}\left(\sum_{k=1}^{K-1} (\Lambda(k+1)\tilde{Y}(k) - 2\sum_{k=1}^{K-1} \Lambda(k)\tilde{Y}(k) + \sum_{k=1}^{K-1} \Lambda(k-1)\tilde{Y}(k)\right) \\
&= \frac{1}{\Delta t}\left(\sum_{k=2}^{K} (\Lambda(k)\tilde{Y}(k-1) - 2\sum_{k=1}^{K-1} \Lambda(k)\tilde{Y}(k) + \sum_{k=0}^{K-2} \Lambda(k)\tilde{Y}(k+1)\right) \\
&= \frac{1}{\Delta t}\sum_{k=1}^{K-1} \Lambda(k)(\tilde{Y}(k-1) - 2\tilde{Y}(k) + \tilde{Y}(k+1)) \\
&\quad + \frac{1}{\Delta t}(\Lambda(K)\tilde{Y}(K-1) - \Lambda(K-1)\tilde{Y}(K)) + \frac{1}{\Delta t}(\Lambda(0)\tilde{Y}(1) - \Lambda(1)\tilde{Y}(0)).
\end{aligned}$$

Consider the boundary terms in the previous sum. First use

$$\tilde{Y}(0) = 0, \quad \frac{1}{\Delta t}\tilde{Y}(1) = D_t^+ \tilde{Y}(0) = \frac{\Delta t}{2} W(0).$$

On the other endpoint, $\Lambda(K) = 0$, and

$$-\Lambda(K-1)\tilde{Y}(K) = (\Lambda(K) - \Lambda(K-1))\tilde{Y}(K) = \Delta t\, D_t^- \Lambda(K)\tilde{Y}(K).$$

Grouping all together gives the partial integration by parts formula

$$\Delta t \sum_{k=1}^{K-1} D_t^+ D_t^-(\Lambda)(k)\, \tilde{Y}(k) = \Delta t \sum_{k=1}^{K-1} D_t^+ D_t^-(\tilde{Y})(k)\, \Lambda(k) + D_t^- \Lambda(K)\tilde{Y}(K) + \frac{\Delta t}{2}\Lambda(0) W(0).$$

Plugging into the formula for $(Y - \widehat{Y}, \tilde{Y})_{J,K}$, gives

$$\begin{aligned}
(Y - \widehat{Y}, \tilde{Y})_{J,K} &= -\Delta t \sum_{k=1}^{K-1} \Big((D_t^+ D_t^- - D_x^+ D_x^-)(\tilde{Y})(\cdot, k), \Lambda(\cdot, k)\Big)_J \\
&\quad + \frac{\Delta t}{2}(Y(\cdot, K) - \widehat{Y}(\cdot, K), \tilde{Y}(\cdot, K))_J - (D_t^- \Lambda(\cdot, K), \tilde{Y}(\cdot, K))_J - \frac{\Delta t}{2}\Lambda(\cdot, 0) W(\cdot, 0). \\
&= -\Delta t \sum_{k=1}^{K-1} (W(\cdot, k), \Lambda(\cdot, k))_J + \left(\frac{\Delta t}{2}(Y(\cdot, K) - \widehat{Y}(\cdot, K)) - D_t^- \Lambda(\cdot, K), \tilde{Y}(\cdot, K)\right)_J \\
&\quad - \frac{\Delta t}{2}\Lambda(\cdot, 0) W(\cdot, 0).
\end{aligned}$$

Add now

$$(Y-\widehat{Y},\tilde{Y})_{J,K}+\gamma(Y(\cdot,K)-\widehat{Z},\tilde{Y}(\cdot,K))_J =$$
$$\left(\frac{\Delta t}{2}(Y(\cdot,K)-\widehat{Y}(\cdot,K))+\gamma(Y(\cdot,K)-\widehat{Z})-D_t^-\Lambda(\cdot,K),\tilde{Y}(\cdot,K)\right)_J$$
$$-\Delta t\sum_{k=1}^{K-1}(W(\cdot,k),\Lambda(\cdot,k))_J-\frac{\Delta t}{2}\Lambda(\cdot,0)W(\cdot,0).$$

Add now

$$\alpha(V,W)_{J,K}=\frac{\alpha\Delta t}{2}(V(\cdot,0),W(\cdot,0))_J+\alpha\Delta t\sum_{k=1}^{K-1}(V(\cdot,k),W(\cdot,k))_J+\frac{\alpha\Delta t}{2}(V(\cdot,K),W(\cdot,K))_J$$

to obtain

$$\widehat{J}_d'(V)\cdot W=\left(\tfrac{\Delta t}{2}(Y(\cdot,K)-\widehat{Y}(\cdot,K))+\gamma(Y(\cdot,K)-\widehat{Z})-D_t^-\Lambda(\cdot,K),\tilde{Y}(\cdot,K)\right)_J$$
$$+\frac{\alpha\Delta t}{2}(V(\cdot,K),W(\cdot,K))_J+\Delta t\sum_{k=1}^{K-1}(W(\cdot,k),\alpha V(\cdot,k)-\Lambda(\cdot,k))_J+\frac{\Delta t}{2}(\alpha V(\cdot,0)\Lambda(\cdot,0),W(\cdot,0))_J$$

Add the final condition on Λ,

$$Y(\cdot,K)-\widehat{Y}(\cdot,K)+\gamma(Y(\cdot,K)-\widehat{Z})-D_t^-\Lambda(\cdot,K)=0.$$

Then the derivative is

$$\widehat{J}_d'(V)\cdot W=\frac{\alpha\Delta t}{2}(V(\cdot,K),W(\cdot,K))_J+\Delta t\sum_{k=1}^{K-1}(W(\cdot,k),\alpha V(\cdot,k)-\Lambda(\cdot,k))_J$$
$$+\frac{\Delta t}{2}(\alpha V(\cdot,0)-\Lambda(\cdot,0),W(\cdot,0))_J$$

And since $\Lambda(\cdot,K)=0$,

$$\begin{aligned}\widehat{J}_d'(V)\cdot W &= \frac{\Delta t}{2}(\alpha V(\cdot,K)-\Lambda(\cdot,K),W(\cdot,K))_J\\ &\quad+\Delta t\sum_{k=1}^{K-1}(W(\cdot,k),\alpha V(\cdot,k)-\Lambda(\cdot,k))_J+\frac{\Delta t}{2}(\alpha V(\cdot,0)-\Lambda(\cdot,0),W(\cdot,0))_J\\ &= (\alpha V-\Lambda,W)_{J,K}.\end{aligned}$$

This proves the form of the derivative. The end of the proof is similar as in the continuous case. □

2.2.2 Discretization of the optimality system

In order to use them with time domain decomposition, we write a monolithic code. Define the vector

$$\underline{Y}(k) = (Y(1, k-1), \ldots, Y(J, k-1)),$$

and similarly for Λ. From (2.2.7), we have the vector recursion

$$\begin{cases} \underline{Y}(k+1) = \dfrac{(\Delta t)^2}{\alpha}\underline{\Lambda}(k) + A\underline{Y}(k) - \underline{Y}(k-1), \ 2 \leq k \leq K \\ \underline{\Lambda}(k-1) = (\Delta t)^2(\widehat{\underline{Y}}(k) - \underline{Y}(k)) + A\underline{\Lambda}(k) - \underline{\Lambda}(k+1), \ 2 \leq k \leq K \\ \underline{Y}(2) = \left(\dfrac{\Delta t}{\Delta x}C + I\right)\underline{Y}(1) + (\Delta t)Y^{(1)} + \dfrac{2}{\alpha}\dfrac{(\Delta t)^2}{2}\underline{\Lambda}(1), \\ \underline{\Lambda}(K) = \left(\dfrac{\Delta t}{\Delta x}C + I\right)\underline{\Lambda}(K+1) + \dfrac{(\Delta t)^2}{2}(\widehat{\underline{Y}}(K+1) - \underline{Y}(K+1)) - \gamma(\Delta t)(\underline{Y}(K+1) - \widehat{\underline{Z}}), \end{cases} \tag{2.2.15}$$

with the initial and final conditions

$$\underline{Y}(1) = \underline{Y}^{(0)}, \ \Lambda(K+1) = 0.$$

where the matrix A and C have the form ($r = \frac{\Delta t}{\Delta x}$)

$$\begin{cases} A = \begin{pmatrix} 2(1-r^2) & r^2 & 0 & 0 & \ldots & 0 \\ r^2 & 2(1-r^2) & r^2 & 0 & \ldots & 0 \\ 0 & r^2 & 2(1-r^2) & r^2 & 0 & 0 \\ 0 & \ldots & r^2 & 2(1-r^2) & r^2 & 0 \\ 0 & \ldots & 0 & r^2 & 2(1-r^2) & r^2 \\ 0 & 0 & \ldots & 0 & r^2 & 2(1-r^2) \end{pmatrix} \\ C = \begin{pmatrix} -r & \frac{r}{2} & 0 & 0 & \ldots & 0 \\ \frac{r}{2} & -r & \frac{r}{2} & 0 & \ldots & 0 \\ 0 & \frac{r}{2} & -r & \frac{r}{2} & 0 & 0 \\ 0 & \ldots & \frac{r}{2} & -r & \frac{r}{2} & 0 \\ 0 & \ldots & 0 & \frac{r}{2} & -r & \frac{r}{2} \\ 0 & 0 & \ldots & 0 & \frac{r}{2} & -r \end{pmatrix} \end{cases} \tag{2.2.16}$$

We implement the scheme on MatLab. In the case $\alpha = 1$,$\gamma = 0$, $a = 0$, $b = 1$ and using the manufactured exact solution

$$y(x,t) = \sin(\pi x)(t - T) \quad \lambda(x,t) = \pi^2 \sin(\pi x)(t - T),$$

we display on Figure 2.3 the associated approximate solution in the case $\Delta t = \Delta x = \frac{1}{N-1}$.

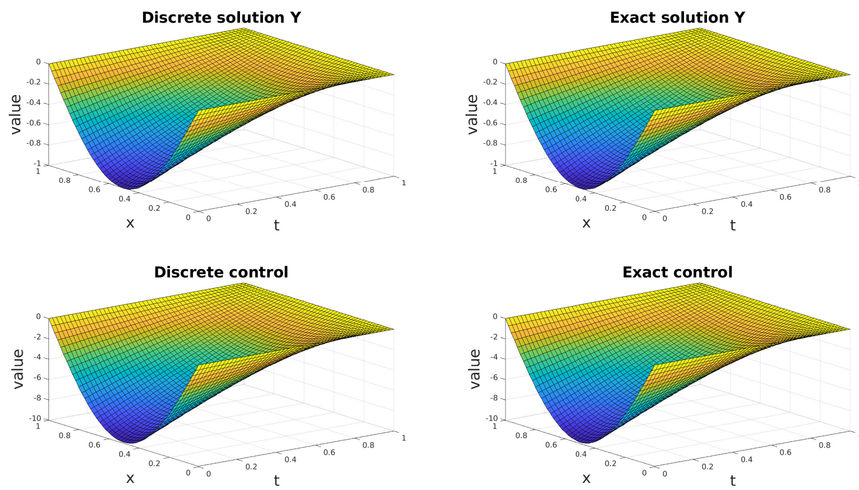


Figure 2.3: Illustration of discrete and exact solution

Then, we make varying N (the number of mesh point in space). We compute the error between the discrete approximate solution and the exact solution for y and λ. We use the weighted discrete l^2 norm defined, for any $v \in \mathbb{R}^{J\times(K+2)}$ by

$$\|v\|_{l^2}^2 = \sum_{j=1}^{J} \sum_{k=0}^{K+1} \Delta x \Delta t \, (v_{j,k}^2),$$

and the l_∞ norm. We observe that the errors are of order 2 (as expected but not proved theoretically), i.e. denoting $y_{\Delta x}$ and $\lambda_{\Delta x}$ the approximate solution, we obtain

$$\begin{cases} ||y_{\Delta x} - y||_{l^2} \leq C\Delta x^2, \\ ||y_{\Delta x} - y||_{l^\infty} \leq C\Delta x^2, \\ ||\lambda_{\Delta x} - \lambda||_{l^2} \leq C\Delta x^2, \end{cases}$$

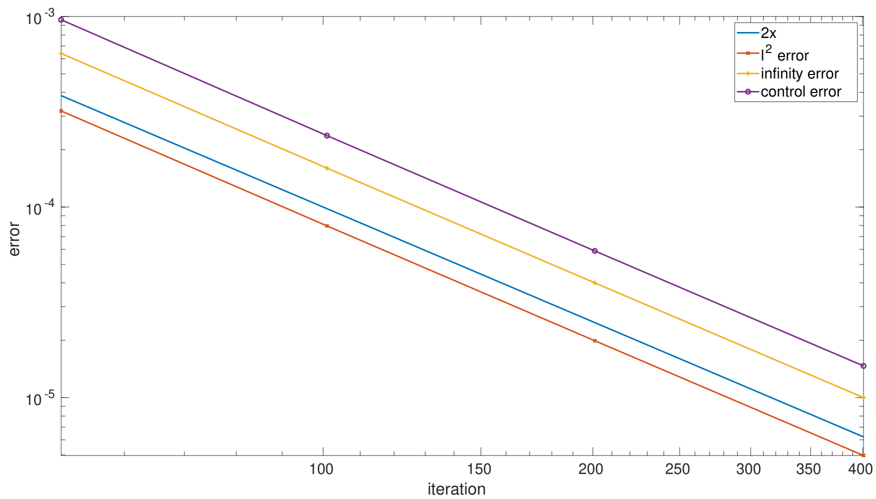


Figure 2.4: Convergence order of coupled wave scheme with respect to $\Delta x = \Delta t$

Chapter 3

Inherited parallel in time algorithm

3.1 Definition of the inherited algorithm

Let $\delta \geq 0$, and let us divide $(0,T)$ into two possibly overlapping intervals $O_1 = (0, T_1 + \delta)$ and $O_2 = (T_1, T)$. Imitating a Schwarz algorithm, but with a decomposition in time, our iterative algorithm solves the optimality conditions (2.1.8b-2.1.8c) in turn in $\Omega_1 = \Omega \times O_1$ and $\Omega_2 = \Omega \times O_2$, see Figure 3.1.

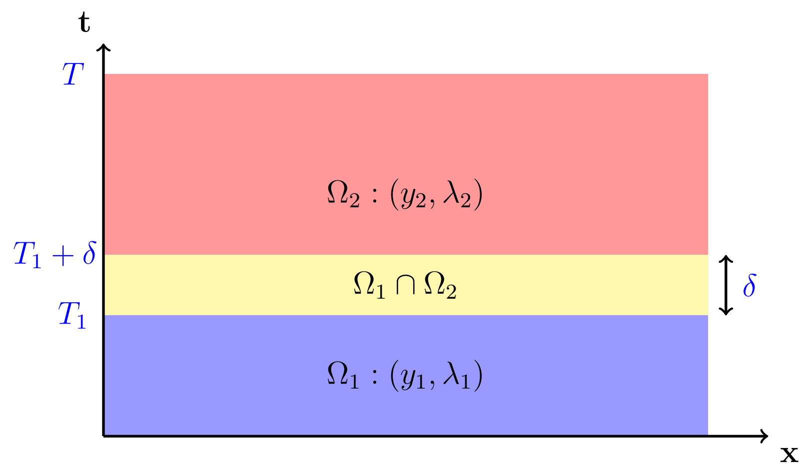


Figure 3.1: Schematic representation of the domains Ω_1 and Ω_2

3.1.1 Statement of the algorithm

The iteration is as follows: we first solve a system of coupled wave equations in domain 1 before solving a similar problem on domain 2 but with initial data coming from domain 1. The solution in Ω_j at iteration n is denoted by (y_j^n, λ_j^n).

Problem for Subdomain O_1

State equation

$$\partial_{tt} y_1^{n+1} - \Delta y_1^{n+1} - \frac{1}{\alpha} \lambda_1^{n+1} = 0, \qquad \text{in } \Omega \times O_1, \tag{3.1.1a}$$

with boundary condition

$$y_1^{n+1} = 0, \qquad \text{in } \partial\Omega \times O_1, \tag{3.1.1b}$$

with initial data

$$\begin{aligned} y_1^{n+1}(\cdot, 0) &= y^{(0)}, & , \\ \partial_t y_1^{n+1}(\cdot, 0) &= y^{(1)}, & \text{in } \Omega, \end{aligned} \tag{3.1.1c}$$

Adjoint equation

$$\partial_{tt}\lambda_1^{n+1} - \Delta\lambda_1^{n+1} + y_1^{n+1} = \widehat{y}, \qquad \text{in } \Omega \times O_1, \tag{3.1.1d}$$

with boundary condition

$$\lambda_1^{n+1} = 0, \qquad \text{in } \partial\Omega \times O_1, \tag{3.1.1e}$$

with the transmission final conditions

$$\lambda_1^{n+1}(\cdot, T_1 + \delta) = \lambda_2^n(\cdot, T_1 + \delta), \tag{3.1.1f}$$

$$\partial_t\lambda_1^{n+1}(\cdot, T_1 + \delta) - \gamma y_1^{n+1}(\cdot, T_1 + \delta) = \partial_t\lambda_2^n(\cdot, T_1 + \delta) - \gamma y_2^n(\cdot, T_1 + \delta), \qquad \text{in } \Omega. \tag{3.1.1g}$$

Problem for subdomain O_2,

State equation

$$\partial_{tt}y_2^{n+1} - \Delta y_2^{n+1} - \frac{1}{\alpha}\lambda_2^{n+1} = 0, \qquad \text{in } \Omega \times O_2, \tag{3.1.2a}$$

with boundary condition

$$y_2^{n+1} = 0, \qquad \text{in } \partial\Omega \times O_2, \tag{3.1.2b}$$

with initial transmission conditions

$$\begin{aligned} y_2^{n+1}(\cdot, T_1) &= y_1^{n+1}(\cdot, T_1), & \text{in } \Omega, \\ \partial_t y_2^{n+1}(\cdot, T_1) &= \partial_t y_1^{n+1}(\cdot, T_1), & \text{in } \Omega. \end{aligned} \tag{3.1.2c}$$

ADJOINT EQUATION

$$\partial_{tt}\lambda_2^{n+1} - \Delta\lambda_2^{n+1} + y_2^{n+1} = \widehat{y}, \qquad \text{in } \Omega \times O_2, \tag{3.1.2d}$$

with boundary condition

$$\lambda_2^{n+1} = 0, \qquad \text{in } \partial\Omega \times O_2. \tag{3.1.2e}$$

with the final conditions

$$\begin{aligned} &\lambda_2^{n+1}(\cdot, T) = 0, \\ &\partial_t\lambda_2^{n+1}(\cdot, T) = \gamma(y_2^{n+1}(\cdot, T) - \widehat{z}), \qquad \text{in } \Omega \end{aligned} \tag{3.1.2f}$$

For $n = 0$, the algorithm is initialized by the transmission final conditions

$$\lambda_1^1(\cdot, T_1 + \delta) = g,\ \partial_t\lambda_1^1(\cdot, T_1 + \delta) - \gamma y_1^1(\cdot, T_1 + \delta) = h.$$

3.1.2 Important properties

On the domain Ω_1, we use the natural initial data of y to initialize y_1 at $T = 0$ (see (3.1.1c)), and we impose final condition on λ_1 at the final $t = T_1 + \delta$ (see (3.1.1g)). We point out that those conditions are similar the final condition (2.1.8c) imposed in λ for the full problem. In the same way, on the domain Ω_2, we know the final data of λ_2. Hence, we impose the transmission condition y_2 at the point $t = T_1$ with the same form of initial data of y at the point $t = 0$. In that sense, the algorithm is **inherited**, because the nature of the subproblems in domain 1 and 2 is the same as the initial one. This property will be very helpful to prove the well-posedness of our algorithm.
Our proposed inherited algorithm differs from a classical Schwarz algorithm. Indeed, if the sequence $(y_i^n, \lambda_i^n)_{i\in\{1,2\}}$ converges toward $(y_i^*, \lambda_i^*)_{i\in\{1,2\}}$, then it satisfies the following four different transmission conditions:

$$y_1^*(\cdot, T_1) = y_2^*(\cdot, T_1), \quad \partial_t y_1^*(\cdot, T_1) = \partial_t y_2^*(\cdot, T_1),$$
$$\lambda_1^*(\cdot, T_1 + \delta) = \lambda_2^*(\cdot, T_1 + \delta),$$

$$\partial_t\lambda_1^*(\cdot, T_1 + \delta) - \gamma y_1^*(\cdot, T_1 + \delta) = \partial_t\lambda_2^*(\cdot, T_1 + \delta) - \gamma y_2^*(\cdot, T_1 + \delta).$$

As a result, for any $\delta \geq 0$, $(y_1^*, \lambda_1^*) = (y_{|\Omega_1}, \lambda_{|\Omega_1})$ and $(y_2^*, \lambda_2^*) = (y_{|\Omega_2}, \lambda_{|\Omega_2})$, namely, if the algorithm converges, it converges toward the solution of the initial problem, which is not the case for a classical non overlapping Schwarz algorithm. In that sense the previous algorithm resembles an optimized Schwarz algorithm.

3.1.3 Well-posedness of the algorithm

We shall see that the subproblems in Ω_1 and Ω_2 correspond to Euler equations associated with adapted quadratic cost functions. As a result, they are well-posed.

Define for given (β_0, β_1) in $L^2(\Omega)$,

$$\begin{aligned} J_1(v_1, y_1) &= \frac{1}{2}\int_{\Omega\times(0,T_1+\delta)} |y_1(x,t) - \widehat{y}(x,t)|^2 dxdt \\ &+ \frac{\gamma}{2}\int_{\Omega} |y_1(x, T_1+\delta)|^2 dx + \frac{\alpha}{2}\int_{\Omega\times(0,T_1+\delta)} |v_1(x,t)|^2 dxdt \\ &- \int_{\Omega} \beta_0(x)\partial_t y_1(x, T_1+\delta)dx + \int_{\Omega} \beta_1(x) y_1(x, T_1+\delta)dx, \end{aligned} \tag{3.1.3}$$

The first two lines contain the quadratic part, the third line is the affine part. Define

$$\widehat{J_1}(v_1) = J_1(v_1, y_1(v_1)), \tag{3.1.4}$$

where $v_1 \in L^1(\Omega_1)$ is the control and y_1 is the solution of system

$$\begin{cases} \partial_{tt} y_1 - \Delta y_1 = v_1 & \text{in } \Omega \times O_1, \\ y_1 = 0 & \text{in } \partial\Omega \times O_1, \\ y_1(\cdot, 0) = y^{(0)}, \quad \partial_t y_1(\cdot, 0) = y^{(1)} & \text{in } \Omega. \end{cases} \tag{3.1.5}$$

Lemma 1. *For any β_0, β_1 in $L^2(\Omega)$, $\widehat{J_1}$ has a unique global minimum. The minimum point is the only solution of the Euler equation $\widehat{J_1'}(u_1) = 0$. The derivative is equal to $\widehat{J_1'}(v_1) = \alpha v_1 - \lambda$, where λ is defined by*

$$\begin{cases} \partial_{tt}\lambda_1 - \Delta\lambda_1 + y_1 = \widehat{y} & \textit{in } \Omega \times O_1, \\ \lambda_1 = 0 & \textit{in } \partial\Omega \times O_1, \\ \lambda_1(\cdot, T_1+\delta) = \beta_0 & \textit{in } \Omega, \\ \partial_t\lambda_1(., T_1+\delta) - \gamma y_1(., T_1+\delta) = \beta_1 & \textit{in } \Omega. \end{cases}.$$

Proof. Similarly to the first chapter, rewrite

$$J_1(v_1,y_1) = \frac{1}{2}\|y_1-\widehat{y}\|^2_{L^2(\Omega\times(0,T_1+\delta))}+\frac{\gamma}{2}\|y_1(\cdot,T_1+\delta)\|^2_{L^2(\Omega)}+\frac{\alpha}{2}\|v_1\|^2_{L^2(\Omega\times(0,T_1+\delta))} - (\beta_0,\partial_t y_1(\cdot,T_1+\delta))_{L^2(\Omega)} + (\beta_1, y_1(\cdot,T_1+\delta))_{L^2(\Omega)}. \tag{3.1.6}$$

Compute the derivative of $\widehat{J}_1$.

$$\widehat{J}_1'(v_1)\cdot w_1 = (y_1-\widehat{y}_1,\tilde{y}_1)_{L^2(\Omega\times(0,T_1+\delta))} + \gamma(y_1(\cdot,T_1+\delta),\tilde{y}_1(\cdot,T_1+\delta))_{L^2(\Omega)} + \alpha(v_1,w_1)_{L^2(\Omega\times(0,T_1+\delta))} - (\beta_0,\partial_t\tilde{y}_1(\cdot,T_1+\delta))_{L^2(\Omega)} + (\beta_1,\tilde{y}_1(\cdot,T_1+\delta))_{L^2(\Omega)}. \tag{3.1.7}$$

where $\tilde{y}_1$ is solution of

$$\begin{cases} \partial_{tt}\tilde{y}_1 - \Delta\tilde{y}_1 = w_1 & \text{in } \Omega\times O_1,\\ \tilde{y}_1 = 0 & \text{in } \partial\Omega\times O_1,\\ \tilde{y}_1(\cdot,0)=0, \quad \partial_t\tilde{y}_1(\cdot,0)=0, & \text{in } \Omega. \end{cases}$$

For the second derivative,

$$\widehat{J}_1''(v_1)(w_1,w_1) = \|\tilde{y}_1\|^2_{L^2(\Omega\times(0,T_1+\delta))}+\gamma\|\tilde{y}_1(\cdot,T_1+\delta)\|^2_{L^2(\Omega)}+\alpha\|w_1\|^2_{L^2(\Omega\times(0,T_1+\delta))} \geq \alpha\|w_1\|^2_{L^2(\Omega\times(0,T_1+\delta))}. \tag{3.1.8}$$

Hence $\widehat{J}_1$ is $\alpha-$convex and therefore admits a unique minimizer u_1, characterized by the Euler equation $\widehat{J}_1'(u_1)$. Choose now λ_1 solution of

$$\begin{array}{ll} \partial_{tt}\lambda_1 - \Delta\lambda_1 + y_1 = \widehat{y} & \text{in } \Omega\times O_1,\\ \lambda_1 = 0 & \text{in } \partial\Omega\times O_1,\\ \lambda_1(\cdot,T_1+\delta) = 0 & \text{in } \Omega. \end{array}$$

Then, as in chapter 2,

$$\begin{aligned} \widehat{J}_1'(v_1)\cdot w_1 \;=\; & (\alpha v_1 - \lambda_1, w_1)_{L^2(\Omega\times(0,T_1+\delta))} + (\gamma y_1(\cdot,T_1+\delta) - \partial_t\lambda_1(\cdot,T_1+\delta), \tilde{y}(\cdot,T_1+\delta))_{L^2(\Omega)}\\ & -(\beta_0,\partial_t\tilde{y}_1(\cdot,T_1+\delta))_{L^2(\Omega)} + (\beta_1,\tilde{y}_1(\cdot,T_1+\delta))_{L^2(\Omega)}\\ & +(\partial_t\tilde{y}_1(\cdot,T_1+\delta),\lambda_1(\cdot,T_1+\delta))_{L^2(\Omega)}. \end{aligned}$$

Write the terms at $t=T_1+\delta$ as

$$(\gamma y_1 - \partial_t\lambda_1 + \beta_1, \tilde{y}_2) + (\lambda_1 - \beta_0, \partial_t\tilde{y}_1)$$

to obtain that if

$$\partial_t \lambda_1 - \gamma y_1 = \beta_1, \quad \lambda_1 = \beta_0,$$

then $\widehat{J}_1'(v_1) = \alpha v_1 - \lambda_1$. □

In the domain Ω_2, we consider the cost function

$$J_2(v_2, y_2) = \frac{1}{2}\|y_2 - \widehat{y}\|^2_{L^2(\Omega\times(T_1,T))} + \frac{\gamma}{2}\|y_2(\cdot,T) - \widehat{z}\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\|v_2\|^2_{L^2(\Omega\times(T_1,T))}, \tag{3.1.9}$$

$$\widehat{J}_2(v_2) = J_2(v_2, y_2(v_2)), \tag{3.1.10}$$

where $v_2 \in L^2(\Omega_2)$ is the control and y_2 is the solution of system

$$\begin{cases} \partial_{tt} y_2 - \Delta y_2 = v_2 & \text{in } \Omega \times O_2, \\ y_2 = 0 & \text{in } \partial\Omega \times O_2, \\ y_2(\cdot, T_1) = \alpha_0, \ \partial_t y_2(\cdot, T_1) = \alpha_1 & \text{in } \Omega. \end{cases} \tag{3.1.11}$$

Lemma 2. *For any α_0, α_1 in $L^2(\Omega)$, $\widehat{J}_2$ has a unique global minimum. The minimum point is the only solution of the Euler equation $\widehat{J}_2'(u_2) = 0$. The derivative is equal to $\widehat{J}_2'(v_2) = \alpha v_2 - \lambda_2$, where λ_2 is defined by*

$$\begin{cases} \partial_{tt} \lambda_2 - \Delta \lambda_2 + y_2 = \widehat{y} & \text{in } \Omega \times O_2, \\ \lambda_2 = 0 & \text{in } \partial\Omega \times O_2, \\ \lambda_2(\cdot, T) = 0 & \text{in } \Omega, \\ \partial_t \lambda_2(.,T) - \gamma(y_2(.,T) - \widehat{z}) = 0 & \text{in } \Omega. \end{cases} \tag{3.1.12}$$

Proof. Compute the derivative of $\widehat{J}_2$.

$$\widehat{J}_2'(v_2) \cdot w_2 = (y_2 - \widehat{y}_2, \tilde{y}_2)_{L^2(\Omega\times O_2)} + \gamma(y_2(\cdot,T) - \widehat{z}, \tilde{y}_2(\cdot,T))_{L^2(\Omega)} + \alpha(v_2, w_2)_{L^2(\Omega\times O_2)}. \tag{3.1.13}$$

where $\tilde{y}_2$ is solution of

$$\begin{cases} \partial \tilde{y}_2 - \Delta \tilde{y}_2 = w_2, & \text{in } \Omega \times O_2, \\ \tilde{y}_2 = 0, & \text{in } \partial\Omega \times O_2, \\ \tilde{y}_2(\cdot, T_1) = 0, \ \partial_t \tilde{y}_2(\cdot, T_1) = 0, & \text{in } \Omega. \end{cases}$$

For the second derivative,

$$\widehat{J}_2''(v_2)(w_2,w_2) = \|\tilde{y}_2\|^2_{L^2(\Omega\times O_2)} + \gamma\|\tilde{y}_2(\cdot,T)\|^2_{L^2(\Omega)} + \alpha\|w_2\|^2_{L^2(\Omega\times O_2)} \geq \alpha\|w_2\|^2_{L^2(\Omega\times O_2)}. \tag{3.1.14}$$

Hence $\widehat{J}_2(v_2)$ is alpha-convex and therefore admits a unique minimizer u_2, characterized by the Euler equation $\widehat{J}_2'(u_2)$. Choose now λ_2 solution of (3.1.12). Then, as in chapter 2,

$$\begin{aligned}
\widehat{J}_2'(v_2)\cdot w_2 &= (\alpha v_2 - \lambda_2, w_2)_{L^2(\Omega\times O_2)} + (\gamma(y_2(\cdot,T) - \widehat{z}) - \partial_t\lambda_2(\cdot,T), \tilde{y}_2(\cdot,T))_{L^2(\Omega)} \\
&\quad -(\tilde{\beta}_0, \partial_t\tilde{y}_2(\cdot,T))_{L^2(\Omega)} + (\tilde{\beta}_1, \tilde{y}_2(\cdot,T))_{L^2(\Omega)} + (\partial_t\tilde{y}_2(\cdot,T), \lambda_2(\cdot,T))_{L^2(\Omega)}, \\
&= (\alpha v_2 - \lambda_2, w_2)_{L^2(\Omega\times O_2)},
\end{aligned}$$

thanks to the final conditions. □

Theorem 3.1.1. *For any initial guess* (λ_1^0, y_1^0) *, the sequence of iterates defined in (3.1.1,3.1.2) is well-defined.*

Proof. It relies on the two previous lemma, with

$$\beta_0 = \lambda_2^n(\cdot,T_1+\delta), \quad \beta_1 = \partial_t\lambda_2^n(\cdot,T_1+\delta) - \gamma y_2^n(\cdot,T_1+\delta),$$

and

$$\alpha_0 = y_1^{n+1}(\cdot,T_1), \quad \alpha_1 = \partial_t y_1^{n+1}(\cdot,T_1).$$

□

3.2 Convergence analysis: computation of the iteration matrix

In order to investigate the convergence of the algorithm, we study the error $(y_j^n - y, \lambda_j^n - \lambda)$, which is solution of the homogeneous equations associated to (3.1.1 ,3.1.2),i.e. (3.1.1 ,3.1.2) with $y^{(0)} = y^{(1)} = 0$, $\hat{y} = 0$, $\hat{z} = 0$. For the sake on simplicity, we translate the time interval into $(-T_1, T_2)$, where T_1 and T_2 are positive and $T_1 + T_2 = T$. The (classical) idea is to use separation of variables and to expand the functions of x and t over the eigenmodes of the Laplacian in space, and to work on the so-obtained differential equations in time.

We start with the system of wave equations

$$\partial_{tt}y - \Delta y - \frac{1}{\alpha}\lambda = 0, \quad \partial_{tt}\lambda - \Delta\lambda + y = 0, \tag{3.2.1}$$

that we rewrite as a system by introducing $Y = (y, \lambda)$

$$\partial_{tt}Y - \Delta Y + \begin{pmatrix} 0 & -\frac{1}{\alpha} \\ 1 & 0 \end{pmatrix} Y = 0. \tag{3.2.2}$$

Then, we expand (y, λ) on the countable eigenmodes of $-\Delta$ in Ω. In one dimension, $\Omega = (a, b)$, and the eigenvalues and eigenmodes of the Laplacian are given by

$$\xi(k) = \frac{k\pi}{b-a}, \quad \Phi_k = \sin(\xi(k)x), \; k \geq 1.$$

Writing

$$Y(x,t) = \sum_{k\geq 1} \hat{Y}(k,t)\Phi_k(t),$$

we obtain the matrix wave equation

$$\partial_{tt}\hat{Y}(k,t) + M(k)\hat{Y}(k,t) = 0, \; M(k) = \begin{pmatrix} \xi^2(k) & -\frac{1}{\alpha} \\ 1 & \xi^2(k) \end{pmatrix}. \tag{3.2.3}$$

3.2.1 Some algebra on matrices

The second order vectorial differential equation (3.2.3) can be solved explicitly, using trigonometric functions, as in the scalar wave equation. A general formula is a variation of

$$\hat{Y}(k,t) = \cos(N(k)t)Y_0(k) + N(k)^{-1}\sin(N(k)t)Y_0'(k), \tag{3.2.4}$$

where $N = \sqrt{M}$.

Let us indicate now how to define properly N, $\cos(N(k)t)$ and $\sin(N(k)t)$, and let us show how to compute them in a rigorous way. We define the matrix

$$J = \begin{pmatrix} 0 & -\frac{1}{\sqrt{\alpha}} \\ \sqrt{\alpha} & 0 \end{pmatrix},$$

which is such that $J^2 = -I$. J plays the role of $i = \sqrt{-1}$ and will be very useful in the work, to develop an algebra parallel to that in the complex numbers. We will see that all the matrices encountered in the analysis below are combinations of the identity matrix I and J, starting from

$$M(k) = \xi^2(k)I + \frac{1}{\sqrt{\alpha}}J.$$

For each k, the matrix $M(k)$ is diagonalizable, with conjugate eigenvalues $\mu(k)$ and $\bar{\mu}(k)$ where

$$\mu(k) = \xi^2(k) + i\sqrt{\frac{1}{\alpha}}.$$

The corresponding eigenmatrix is given by

$$P = \begin{pmatrix} \frac{i}{\sqrt{\alpha}} & \frac{-i}{\sqrt{\alpha}} \\ 1 & 1 \end{pmatrix}$$

and is independent of k. Therefore, the square root of M is defined by

$$N = \sqrt{M} = P\sqrt{D}P^{-1} \quad \text{with } D = \begin{pmatrix} \sqrt{\mu(k)} & 0 \\ 0 & \sqrt{\bar{\mu}(k)} \end{pmatrix}.$$

Let

$$\nu = \sqrt{\mu} = \nu_1 + i\nu_2,$$

where ν_1 and ν_2 real positive. Let us collect useful notations for the sequel

$$\begin{array}{l} \mu = \xi^2 + \frac{i}{\sqrt{\alpha}}, \quad \nu = \sqrt{\mu} = \nu_1 + i\nu_2 \\ \nu_1^2 - \nu_2^2 = \xi^2, \quad 2\nu_1\nu_2 = \frac{1}{\sqrt{\alpha}} \\ \nu_1 = \frac{1}{\sqrt{2}}\sqrt{\xi^2 + \sqrt{\xi^4 + \frac{1}{\alpha}}}, \quad \nu_2 = \frac{1}{\sqrt{2}}\sqrt{-\xi^2 + \sqrt{\xi^4 + \frac{1}{\alpha}}}. \end{array} \tag{3.2.5}$$

All these quantities are functions of k, or similarly of ξ. A direct computation shows that N

$$N = \nu_1 I + \nu_2 J, \quad N^{-1} = \frac{\nu_1 I - \nu_2 J}{|\nu|^2} = \operatorname{Re}\nu^{-1} I + \operatorname{Im}\nu^{-1} J. \tag{3.2.6}$$

Indeed,

$$(\nu_1 I + \nu_2 J)^2 = (\nu_1^2 - \nu_2^2)I + 2\nu_1\nu_2 J = \operatorname{Re}\mu I + \operatorname{Im}\mu J = M,$$

$$(\nu_1 I + \nu_2 J)(\nu_1 I - \nu_2 J) = |\nu|^2 I.$$

Trigonometric matrices Let us start with generalities. The cosine and sine of a matrix A must be understood as the sum of the Taylor series. If A and B commute, then the usual trigonometric properties on sine and cosine of $A+B$ are valid. In our case, since one of the matrices is a multiple of identity, it is always true. In particular the addition formulas are valid, *i.e.*

$$\cos(A+B) = \cos A \cos B - \sin A \sin B, \ \ etc.$$

Furthermore $\cos(\nu I) = \cos(\nu) I$ and $\sin(\nu I) = \sin(\nu) I$. Compute now

$$\cos(\nu J) = \sum_{p=0}^{\infty} (-1)^p \frac{(\nu J)^{2p}}{(2p)!} = \sum_{p=0}^{\infty} \frac{\nu^{2p}}{(2p)!} I = \cosh(\nu) I.$$

$$\sin(\nu J) = \sum_{p=0}^{\infty} (-1)^p \frac{(\nu J)^{2p+1}}{(2p+1)!} = \sum_{p=0}^{\infty} \frac{\nu^{2p+1}}{(2p+1)!} J = \sinh(\nu) J.$$

From this, by the addition formulas, we deduce the elegant formulas

$$\begin{aligned} C(t) &:= \cos(Nt) = \cos(\nu_1 t I + \nu_2 t J) = \cos \nu_1 t \cosh \nu_2 t I - \sin \nu_1 t \sinh \nu_2 t J, \\ S(t) &:= \sin(Nt) = \sin(\nu_1 t I + \nu_2 t J) = \sin \nu_1 t \cosh \nu_2 t I + \cos \nu_1 t \sinh \nu_2 t J, \end{aligned} \tag{3.2.7}$$

which are parallel to the formulas

$$\begin{aligned} c(t) &:= \cos \nu t = \cos \nu_1 t \cosh \nu_2 t - i \sin \nu_1 t \sinh \nu_2 t, \\ s(t) &:= \sin \nu t = \sin \nu_1 t \cosh \nu_2 t + i \cos \nu_1 t \sinh \nu_2 t, \end{aligned} \tag{3.2.8}$$

We will use, according to our need, the formulas

$$\begin{aligned} \cos(Nt) &= \operatorname{Re} \cos \nu t I + \operatorname{Im} \cos \nu t J, \\ \sin(Nt) &= \operatorname{Re} \sin \nu t I + \operatorname{Im} \sin \nu t J. \end{aligned} \tag{3.2.9}$$

$$\begin{aligned} C(t) &= \operatorname{Re} c(t) I + \operatorname{Im} c(t) J, \\ S(t) &= \operatorname{Re} s(t) I + \operatorname{Im} s(t) J, \end{aligned} \tag{3.2.10}$$

and also

$$C(t) = \begin{pmatrix} \operatorname{Re} c(t) & -\operatorname{Im} c(t)/\sqrt{\alpha} \\ \sqrt{\alpha} \operatorname{Im} c(t) & \operatorname{Re} c(t) \end{pmatrix}, \tag{3.2.11}$$

and similarly for $S(t)$ and so on. Using again Taylor series, it can be shown that the derivatives of sine and cosine are calculated as in the scalar case

$$\begin{aligned} &\tfrac{d}{dt} \cos(Nt) = -N \sin(Nt), \ \tfrac{d}{dt} \sin(Nt) = N \cos(Nt), \\ &C'(t) = -NS(t), \ S'(t) = NC(t). \end{aligned} \tag{3.2.12}$$

This validates formula (3.2.4).

Furthermore, we compute $N\sin(Nt)$ and $N^{-1}\sin(Nt)$ using the algebra defined above. For example

$$
\begin{aligned}
N\sin(Nt) &= (\nu_1 I + \nu_2 J)(\operatorname{Re}\sin\nu t I + \operatorname{Im}\sin\nu t J),\\
&= (\nu_1 \operatorname{Re}\sin\nu t - \nu_2 \operatorname{Im}\sin\nu t)I + (\nu_1 \operatorname{Im}\sin\nu t + \nu_2 \operatorname{Re}\sin\nu t)J,\\
&= \operatorname{Re}\nu\sin\nu t I + \operatorname{Im}\nu\sin\nu t J.
\end{aligned}
$$

So we have the useful formulas

$$
\begin{cases}
\sin(Nt) = \operatorname{Re}\sin\nu t I + \operatorname{Im}\sin\nu t J,\\
\cos(Nt) = \operatorname{Re}\cos\nu t I + \operatorname{Im}\cos\nu t J,\\
N\sin(Nt) = \operatorname{Re}\nu\sin\nu t I + \operatorname{Im}\nu\sin\nu t J,\\
N\cos(Nt) = \operatorname{Re}\nu\cos\nu t I + \operatorname{Im}\nu\cos\nu t J,\\
N^{-1}\sin(Nt) = \operatorname{Re}\nu^{-1}\sin\nu t I + \operatorname{Im}\nu^{-1}\sin\nu t J,\\
N^{-1}\cos(Nt) = \operatorname{Re}\nu^{-1}\cos\nu t I + \operatorname{Im}\nu^{-1}\cos\nu t J.
\end{cases}
\tag{3.2.13}
$$

Collect all necessary formulas for the functions of t

$$
\begin{cases}
C = \operatorname{Re} c I + \operatorname{Im} c J,\\
S = \operatorname{Re} s I + \operatorname{Im} s J,\\
NS = \operatorname{Re}(\nu s) I + \operatorname{Im}(\nu s) J,\\
NC = \operatorname{Re}(\nu c) I + \operatorname{Im}(\nu c) J,\\
N^{-1}S = \operatorname{Re}(\nu^{-1} s) I + \operatorname{Im}(\nu^{-1} s) J,\\
N^{-1}C = \operatorname{Re}(\nu^{-1} c) I + \operatorname{Im}(\nu^{-1} c) J.
\end{cases}
\tag{3.2.14}
$$

3.2.2 The infinite domain case

We consider here $T_1 = T_2 = +\infty$.

Computation of the convergence matrix

Start with the problem in subdomain 1, whose solution we call $Y_1 = (y_1, \lambda_1)$. By separation of variables, we write it as

$$Y_1(x,t) = \sum_{k\geq 1} \hat{Y}_1(k,t)\Phi_k(x),$$

and implement first the boundary condition at infinity,

$$\lim_{t\to-\infty} Y_1(\cdot,t) = 0.$$

We use a simpler variation of (3.2.4).

$$\hat{Y}_1(k,t) = \cos(Nt)\hat{G}(k) + \sin(Nt)\tilde{G}(k). \tag{3.2.15}$$

We write first the behavior at infinity of the real hyperbolic sine and cosine functions:

$$\cosh\nu_2 t \sim \frac{1}{2}e^{-\nu_2 t} := X, \quad \sinh\nu_2 t \sim -\frac{1}{2}e^{-\nu_2 t} = -X.$$

X is the large parameter. This implies by (3.2.8)

$$\operatorname{Re}\cos\nu t \sim X\cos\nu_1 t, \quad \operatorname{Im}\cos\nu t \sim X\sin\nu_1 t,$$

$$\Longrightarrow \quad \cos(Nt) \sim X(\cos\nu_1 tI + \sin\nu_1 tJ).$$

$$\operatorname{Re}\sin\nu t \sim X\sin\nu_1 t, \quad \operatorname{Im}\sin\nu t \sim -X\cos\nu_1 t,$$

$$\Longrightarrow \quad \sin(Nt) \sim X(\sin\nu_1 tI - \cos\nu_1 tJ).$$

Insert into (3.2.15),

$$\hat{Y}_1 \sim X\left[(\cos\nu_1 tI + \sin\nu_1 tJ)\hat{G} + (\sin\nu_1 tI - \cos\nu_1 tJ)\tilde{G}\right].$$

Reorder in cosine and sine

$$\hat{Y}_1 \sim X\left[\cos\nu_1 t(\hat{G} - J\tilde{G}) + \sin\nu_1 t(J\hat{G} + \tilde{G})\right].$$

Use that $J^2 = -I$,

$$\hat{Y}_1 \sim X\left[\cos\nu_1 t(\hat{G} - J\tilde{G}) + \sin\nu_1 tJ(\hat{G} - J\tilde{G})\right] \sim X(\cos\nu_1 tI + \sin\nu_1 tJ)(\hat{G} - J\tilde{G}).$$

The matrix $\cos\nu_1 tI + \sin\nu_1 tJ$ is a rotation matrix, has no limit at infinity. If Y has to tend to 0 at $-\infty$, we must have $\hat{G} - J\tilde{G} = 0$. Therefore $\tilde{G} = -J\hat{G}$ and since J and N commute, we have

$$\hat{Y}_1 = (\cos(Nt) - J\sin(Nt))\hat{G}. \tag{3.2.16}$$

Write the same formula in $(0,+\infty)$,

$$\hat{Y}_2 = \cos(Nt)\tilde{H} + \sin(Nt)\hat{H}. \tag{3.2.17}$$

Enforce the condition at infinity $Y(\cdot,+\infty)=0$.

$$\cosh \nu_2 t \sim \frac{1}{2}e^{\nu_2 t} := X, \quad \sinh \nu_2 t \sim \frac{1}{2}e^{\nu_2 t} = X.$$

X is the new large parameter. First by (3.2.7),

$$\cos(Nt) \sim X(\cos\nu_1 tI - \sin\nu_1 tJ),\ \sin(Nt) \sim X(\sin\nu_1 tI + \cos\nu_1 tJ).$$

Insert into (3.2.17),

$$\begin{aligned}
\hat{Y}_2 &\sim X\left[(\cos\nu_1 tI - \sin\nu_1 tJ)\tilde{H} + (\sin\nu_1 tI + \cos\nu_1 tJ)\hat{H}\right],\\
&\sim X\left[(\cos\nu_1 tI - \sin\nu_1 tJ)\tilde{H} + (-\sin\nu_1 tJ + \cos\nu_1 tI)J\hat{H}\right],\\
&\sim X(\cos\nu_1 tI - \sin\nu_1 tJ)\left[\tilde{H} + J\hat{H}\right].
\end{aligned}$$

Consequently

$$\tilde{H} + J\hat{H} = 0, \text{ or equivalently } \hat{H} = J\tilde{H}.$$

Hence

$$\hat{Y}_2(k,t) = (\cos(Nt)I + \sin(Nt)J)\hat{H}, \tag{3.2.18}$$

renaming $\tilde{H}$ by $\hat{H}$. We will need to express separately y and λ in these formulas, in order to be able to enforce the transmission conditions. For this we use again formulas (3.2.9) or in the concise form (3.2.10).

$$\begin{aligned}
\cos(Nt)I - \sin(Nt)J &= \operatorname{Re} c(t)I + \operatorname{Im} c(t)J - (\operatorname{Re} s(t)I + \operatorname{Im} s(t)J)J,\\
&= (\operatorname{Re} c(t) + \operatorname{Im} s(t))I + (\operatorname{Im} c(t) - \operatorname{Re} s(t))J,\\
\hat{Y}_1(k,t) &= (\operatorname{Re} c(t) + \operatorname{Im} s(t))\hat{G} + (\operatorname{Im} c(t) - \operatorname{Re} s(t))J\hat{G}.
\end{aligned}$$

Use now

$$J\begin{pmatrix} G_1 \\ G_2 \end{pmatrix} = \begin{pmatrix} -\frac{1}{\sqrt{\alpha}}G_2 \\ \sqrt{\alpha}G_1 \end{pmatrix},$$

to write y and λ separately:

$$\left\{\begin{array}{lcl}
\hat{y}_1 &=& (\operatorname{Re} c(t) + \operatorname{Im} s(t))\hat{G}_1 - \frac{1}{\sqrt{\alpha}}(\operatorname{Im} c(t) - \operatorname{Re} s(t))\hat{G}_2,\\
\hat{\lambda}_1 &=& (\operatorname{Re} c(t) + \operatorname{Im} s(t))\hat{G}_2 + \sqrt{\alpha}(\operatorname{Im} c(t) - \operatorname{Re} s(t))\hat{G}_1.
\end{array}\right. \tag{3.2.19}$$

It is very easy to differentiate in time, from (3.2.12),

$$\frac{d}{dt}\hat{Y}_1(k,t) = -N\sin(Nt)\hat{G}(k) - JN\cos(Nt)\hat{G}(k), \tag{3.2.20}$$

and by the formulas (3.2.14), it suffices to replace in (3.2.19) $c(t)$ by $-\nu s(t)$ and $s(t)$ by $\nu c(t)$, to obtain

$$\left\{\begin{array}{lcl} \hat{y}_1' &=& (-\operatorname{Re}\nu s(t) + \operatorname{Im}\nu c(t))\hat{G}_1 - \frac{1}{\sqrt{\alpha}}(-\operatorname{Im}\nu s(t) - \operatorname{Re}\nu c(t))\hat{G}_2, \\ \hat{\lambda}_1' &=& (-\operatorname{Re}\nu s(t) + \operatorname{Im}\nu c(t))\hat{G}_2 + \sqrt{\alpha}(-\operatorname{Im}\nu s(t) - \operatorname{Re}\nu c(t))\hat{G}_1. \end{array}\right. \tag{3.2.21}$$

For Y_2 just change t in $-t$, that is s into $-s$:

$$\left\{\begin{array}{lcl} \hat{y}_2 &=& (\operatorname{Re} c(t) - \operatorname{Im} s(t))\hat{H}_1 - \frac{1}{\sqrt{\alpha}}(\operatorname{Im} c(t) + \operatorname{Re} s(t))\hat{H}_2, \\ \hat{\lambda}_2 &=& (\operatorname{Re} c(t) - \operatorname{Im} s(t))\hat{H}_2 + \sqrt{\alpha}(\operatorname{Im} c(t) + \operatorname{Re} s(t))\hat{H}_1. \end{array}\right. \tag{3.2.22}$$

$$\frac{d}{dt}\hat{Y}_2(k,t) = -N\sin(Nt)\hat{H}(k) + N\cos(Nt)\tilde{H}(k), \tag{3.2.23}$$

$$\left\{\begin{array}{lcl} \hat{y}_2' &=& (-\operatorname{Re}\nu s(t) - \operatorname{Im}\nu c(t))\hat{H}_1 - \frac{1}{\sqrt{\alpha}}(-\operatorname{Im}\nu s(t) + \operatorname{Re}\nu c(t))\hat{H}_2, \\ \hat{\lambda}_2' &=& (-\operatorname{Re}\nu s(t) - \operatorname{Im}\nu c(t))\hat{H}_2 + \sqrt{\alpha}(-\operatorname{Im}\nu s(t) + \operatorname{Re}\nu c(t))\hat{H}_1. \end{array}\right. \tag{3.2.24}$$

It is easy now to address the transmission conditions

$$\begin{aligned} &\hat{\lambda}_1^{n+1}(\cdot,\delta) = \hat{\lambda}_2^{n}(\cdot,\delta) \iff \\ &(\operatorname{Re} c(\delta) + \operatorname{Im} s(\delta))\hat{G}_2^{n+1} + \sqrt{\alpha}(\operatorname{Im} c(\delta) - \operatorname{Re} s(\delta))\hat{G}_1^{n+1} = \\ &\qquad (\operatorname{Re} c(\delta) - \operatorname{Im} s(\delta))\hat{H}_2^{n} + \sqrt{\alpha}(\operatorname{Im} c(\delta) + \operatorname{Re} s(\delta))\hat{H}_1^{n}, \end{aligned}$$

and

$$\begin{aligned} &(\partial_t\hat{\lambda}_1^{n+1} - \gamma\hat{y}_1^{n+1})(\cdot,\delta) = (\partial_t\hat{\lambda}_2^{n} - \gamma\hat{y}_2^{n})(\cdot,\delta) \iff \\ &(-\operatorname{Re}\nu s(\delta) + \operatorname{Im}\nu c(\delta))\hat{G}_2^{n+1} + \sqrt{\alpha}(-\operatorname{Im}\nu s(\delta) - \operatorname{Re}\nu c(\delta))\hat{G}_1^{n+1} \\ &- \gamma((\operatorname{Re} c(\delta) + \operatorname{Im} s(\delta))\hat{G}_1^{n+1} - \frac{1}{\sqrt{\alpha}}(\operatorname{Im} c(\delta) - \operatorname{Re} s(\delta))\hat{G}_2^{n+1}) = \\ &- (\operatorname{Re}\nu s(\delta) + \operatorname{Im}\nu c(\delta))\hat{H}_2^{n} - \sqrt{\alpha}(\operatorname{Im}\nu s(\delta) - \operatorname{Re}\nu c(\delta))\hat{H}_1^{n} \\ &- \gamma((\operatorname{Re} c(\delta) - \operatorname{Im} s(\delta))\hat{H}_1^{n} - \frac{1}{\sqrt{\alpha}}(\operatorname{Im} c(\delta) + \operatorname{Re} s(\delta))\hat{H}_2^{n}). \end{aligned}$$

We rewrite this as a system, defining two matrices,

$$\tilde{S}_1(t) = \begin{pmatrix} \sqrt{\alpha}(\operatorname{Im} c(t) - \operatorname{Re} s(t)) & \operatorname{Re} c(t) + \operatorname{Im} s(t) \\ \big[-\sqrt{\alpha}(\operatorname{Im} \nu s(t) + \operatorname{Re} \nu c(t)) & \big[\operatorname{Im} \nu c(t) - \operatorname{Re} \nu s(t) \\ -\gamma(\operatorname{Re} c(t) + \operatorname{Im} s(t))\big] & +\frac{\gamma}{\sqrt{\alpha}}(\operatorname{Im} c(t) - \operatorname{Re} s(t))\big] \end{pmatrix} \tag{3.2.25}$$

$$S_1(t) = \begin{pmatrix} \sqrt{\alpha}(\operatorname{Im} c(t) + \operatorname{Re} s(t)) & \operatorname{Re} c(t) - \operatorname{Im} s(t) \\ \big[\sqrt{\alpha}(-\operatorname{Im} \nu s(t) + \operatorname{Re} \nu c(t)) & \big[-(\operatorname{Im} \nu c(t) + \operatorname{Re} \nu s(t)) \\ -\gamma(\operatorname{Re} c(t) - \operatorname{Im} s(t))\big] & +\frac{\gamma}{\sqrt{\alpha}}(\operatorname{Im} c(t) + \operatorname{Re} s(t))\big] \end{pmatrix} \tag{3.2.26}$$

and

$$\tilde{S}_1(\delta)\hat{G}^{n+1} = S_1(\delta)\hat{H}^n, \text{ or}$$

$$\hat{G}^{n+1} = M_1\hat{H}^n, M_1 = \tilde{S}_1^{-1}(\delta)S_1(\delta).$$

Rewrite $\tilde{S}_1$ more explicitly.

$$\operatorname{Re} s - \operatorname{Im} c = \sin\nu_1 t(\sinh\nu_2 t + \cosh\nu_2 t) = \sin\nu_1 t e^{\nu_2 t},$$

$$\operatorname{Re} c + \operatorname{Im} s = \cos\nu_1 t(\sinh\nu_2 t + \cosh\nu_2 t) = \cos\nu_1 t e^{\nu_2 t}.$$

We need now to express $\operatorname{Im} \nu s(t) + \operatorname{Re} \nu c(t)$ and $\operatorname{Im} \nu c(t) - \operatorname{Re} \nu s(t)$

$$\begin{aligned} \operatorname{Re} \nu c + \operatorname{Im} \nu s &= \nu_1 \operatorname{Re} c - \nu_2 \operatorname{Im} c + \nu_1 \operatorname{Im} s + \nu_2 \operatorname{Re} s, \\ &= \nu_1(\operatorname{Re} c + \operatorname{Im} s) + \nu_2(\operatorname{Re} s - \operatorname{Im} c), \\ &= (\nu_1 \cos\nu_1 t + \nu_2 \sin\nu_1 t)e^{\nu_2 t}. \end{aligned}$$

Similarly,

$$\begin{aligned} \operatorname{Im} \nu c(t) - \operatorname{Re} \nu s(t) &= \nu_1 \operatorname{Im} c + \nu_2 \operatorname{Re} c - (\nu_1 \operatorname{Re} s - \nu_2 \operatorname{Im} s), \\ &= -\nu_1(\operatorname{Re} s - \operatorname{Im} c) + \nu_2(\operatorname{Re} c + \operatorname{Im} s), \\ &= (-\nu_1 \sin\nu_1 t + \nu_2 \cos\nu_1 t)e^{\nu_2 t}. \end{aligned}$$

We obtain

$$\tilde{S}_1(t) = e^{\nu_2 t}\begin{pmatrix} -\sqrt{\alpha}\sin\nu_1 t & \cos\nu_1 t \\ -(\sqrt{\alpha}\nu_1 + \gamma)\cos\nu_1 t - \sqrt{\alpha}\nu_2 \sin\nu_1 t & -(\nu_1 + \frac{\gamma}{\sqrt{\alpha}})\sin\nu_1 t + \nu_2 \cos\nu_1 t \end{pmatrix} \tag{3.2.27}$$

We easily compute the determinant of $\tilde{S}_1(t)$:

$$\det\tilde{S}_1(t) = (\sqrt{\alpha}\nu_1 + \gamma)e^{2\nu_2 t} > 0.$$

Do the same on S_1:

$$\operatorname{Re} s + \operatorname{Im} c = \sin\nu_1 t e^{-\nu_2 t},$$

$$\operatorname{Re} c - \operatorname{Im} s = \cos\nu_1 t e^{-\nu_2 t},$$

$$\begin{aligned}
\operatorname{Re}\nu c - \operatorname{Im}\nu s &= \nu_1 \operatorname{Re} c - \nu_2 \operatorname{Im} c - (\nu_1 \operatorname{Im} s + \nu_2 \operatorname{Re} s), \\
&= \nu_1(\operatorname{Re} c - \operatorname{Im} s) - \nu_2(\operatorname{Re} s + \operatorname{Im} c), \\
&= (\nu_1 \cos\nu_1 t - \nu_2 \sin\nu_1 t)e^{-\nu_2 t}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\operatorname{Im}\nu c(t) + \operatorname{Re}\nu s(t) &= \nu_1 \operatorname{Im} c + \nu_2 \operatorname{Re} c + \nu_1 \operatorname{Re} s - \nu_2 \operatorname{Im} s, \\
&= \nu_1(\operatorname{Re} s + \operatorname{Im} c) + \nu_2(\operatorname{Re} c - \operatorname{Im} s), \\
&= (\nu_1 \sin\nu_1 t + \nu_2 \cos\nu_1 t)e^{-\nu_2 t}.
\end{aligned}$$

$$S_1(t) = e^{-\nu_2 t}\begin{pmatrix} \sqrt{\alpha}\sin\nu_1 t & \cos\nu_1 t \\ (\sqrt{\alpha}\nu_1 - \gamma)\cos\nu_1 t - \sqrt{\alpha}\nu_2 \sin\nu_1 t & (-\nu_1 + \frac{\gamma}{\sqrt{\alpha}})\sin\nu_1 t - \nu_2\cos\nu_1 t \end{pmatrix}.$$

Compute now the iteration matrix $M_1 = \tilde{S}_1^{-1}(t)S_1(t)$. First

$$\tilde{S}_1^{-1}(t) = \frac{e^{-\nu_2 t}}{\sqrt{\alpha}\nu_1 + \gamma}\begin{pmatrix} -(\nu_1 + \frac{\gamma}{\sqrt{\alpha}})\sin\nu_1 t + \nu_2\cos\nu_1 t & -\cos\nu_1 t \\ (\sqrt{\alpha}\nu_1 + \gamma)\cos\nu_1 t + \sqrt{\alpha}\nu_2\sin\nu_1 t & -\sqrt{\alpha}\sin\nu_1 t \end{pmatrix}$$

A long calculation shows

$$M_1 = \frac{e^{-2\nu_2 t}}{\sqrt{\alpha}\nu_1 + \gamma}\begin{pmatrix} -\sqrt{\alpha}\nu_1 + \gamma\cos(2\nu_1 t) + \nu_2\sqrt{\alpha}\sin(2\nu_1 t) & 2\cos\nu_1 t(-\frac{\gamma}{\sqrt{\alpha}}\sin\nu_1 t + \nu_2\cos\nu_1 t) \\ 2\sqrt{\alpha}\sin\nu_1 t(\gamma\cos\nu_1 t + \sqrt{\alpha}\nu_2\sin\nu_1 t) & \sqrt{\alpha}\nu_1 + \gamma\cos(2\nu_1 t) + \nu_2\sqrt{\alpha}\sin(2\nu_1 t) \end{pmatrix},$$

Now the transmission conditions

$$\hat{y}_2^{n+1}(0) = \hat{y}_1^{n+1}(0), \quad \partial_t\hat{y}_2^{n+1}(0) = \partial_t\hat{y}_1^{n+1}(0),$$

are easy to treat. They give

$$\begin{aligned}
&\hat{H}_1^{n+1} = \hat{G}_1^{n+1}, \\
&-\nu_2\hat{H}_1^{n+1} - \tfrac{\nu_1}{\sqrt{\alpha}}\hat{H}_2^{n+1} = \nu_2\hat{G}_1^{n+1} + \tfrac{\nu_1}{\sqrt{\alpha}}\hat{G}_2^{n+1}.
\end{aligned}$$

In matrix form

$$\tilde{S}_2\hat{H}^{n+1} = S_2\hat{G}^{n+1},$$

$$\tilde{S}_2 = \begin{pmatrix} 1 & 0 \\ -\nu_2 & -\frac{\nu_1}{\sqrt{\alpha}} \end{pmatrix}, \quad S_2 = \begin{pmatrix} 1 & 0 \\ \nu_2 & \frac{\nu_1}{\sqrt{\alpha}} \end{pmatrix}.$$

Consequently,

$$\hat{H}^{n+1} = M_2\hat{G}^{n+1}, \quad M_2 := \tilde{S}_2^{-1}S_2 = -\frac{\sqrt{\alpha}}{\nu_1}\begin{pmatrix} -\frac{\nu_1}{\sqrt{\alpha}} & 0 \\ 2\nu_2 & \frac{\nu_1}{\sqrt{\alpha}} \end{pmatrix}.$$

We couple now

$$\hat{G}^{n+1} = M_1 \hat{H}^n, \quad M_1 := \tilde{S}_1^{-1}(\delta) S_1(\delta), \quad \hat{H}^n = M_2 \hat{G}^n, \quad M_2 := \tilde{S}_2^{-1} S_2,$$

to obtain the iterate

$$\hat{G}^{n+1} = M \hat{G}^n, \quad M = M_1 M_2.$$

Compute now the matrix M, or rather what we need from matrix M: the determinant and the trace. It is easy to find that

$$\det M = \frac{\det S_1(\delta)}{\det \tilde{S}_1(\delta)} \frac{\det S_2}{\det \tilde{S}_2} = -\frac{\det S_1(\delta)}{\det \tilde{S}_1(\delta)} = e^{-4\nu_2\delta} \frac{\gamma - \sqrt{\alpha}\nu_1}{\gamma + \sqrt{\alpha}\nu_1}.$$

Now

$$\begin{aligned}
\operatorname{Tr} M &= (M_1)_{11}(M_2)_{11} + (M_1)_{12}(M_2)_{21} + (M_1)_{21}(M_2)_{12} + (M_1)_{22}(M_2)_{22}, \\
&= -\tfrac{\sqrt{\alpha}}{\nu_1}\left(-\tfrac{\nu_1}{\sqrt{\alpha}}(M_1)_{11} + 2\nu_2 (M_1)_{12} + \tfrac{\nu_1}{\sqrt{\alpha}}(M_1)_{22}\right), \\
&= -\tfrac{\sqrt{\alpha}}{\nu_1}\left(\tfrac{\nu_1}{\sqrt{\alpha}}((M_1)_{22} - (M_1)_{11}) + 2\nu_2 (M_1)_{12}\right), \\
&= -\tfrac{\sqrt{\alpha}}{\nu_1}\tfrac{e^{-2\nu_2\delta}}{\sqrt{\alpha}\nu_1+\gamma}\left(\tfrac{\nu_1}{\sqrt{\alpha}}(2\nu_1\sqrt{\alpha}) + 4\nu_2 \cos\nu_1\delta(-\tfrac{\gamma}{\sqrt{\alpha}}\sin\nu_1\delta + \nu_2\cos\nu_1\delta)\right), \\
&= -2\tfrac{\sqrt{\alpha}}{\nu_1}\tfrac{e^{-2\nu_2\delta}}{\sqrt{\alpha}\nu_1+\gamma}\left(\nu_1^2 + 2\nu_2^2\cos^2\nu_1\delta - 2\nu_2\tfrac{\gamma}{\sqrt{\alpha}}\sin\nu_1\delta\cos\nu_1\delta\right), \\
&= -2\tfrac{\sqrt{\alpha}}{\nu_1}\tfrac{e^{-2\nu_2\delta}}{\sqrt{\alpha}\nu_1+\gamma}\left(|\nu|^2 + \nu_2^2\cos 2\nu_1\delta - \nu_2\tfrac{\gamma}{\sqrt{\alpha}}\sin 2\nu_1\delta\right).
\end{aligned}$$

using that

$$2\cos^2\theta = 1 + \cos 2\theta, \; 2\sin\theta\cos\theta = \sin 2\theta.$$

The characteristic polynomial is $\lambda^2 - \operatorname{Tr} M\lambda + \det M$. The reduced discriminant is

$$\Delta' = \frac{\alpha}{\nu_1^2}\frac{e^{-4\nu_2\delta}}{(\sqrt{\alpha}\nu_1+\gamma)^2}\left(|\nu|^2 + \nu_2^2\cos 2\nu_1\delta - \nu_2\frac{\gamma}{\sqrt{\alpha}}\sin 2\nu_1\delta\right)^2 - e^{-4\nu_2\delta}\frac{\gamma - \sqrt{\alpha}\nu_1}{\gamma + \sqrt{\alpha}\nu_1}.$$

Analysis of the case $\gamma = 0$

The general analysis of the convergence matrix is not easy. However, in the case $\gamma = 0$, we can show a divergence result:

Theorem 3.2.1. *Assume that $\gamma = 0$. For any $\alpha > 0$, there exists, $\delta_0 > 0$, such that, for any $\delta < \delta_0$,the algorithm is divergent.*

Proof. In that case, we see that the discriminant is positive, the trace is negative:

$$\Delta' = \frac{e^{-4\nu_2\delta}}{\nu_1^4}\left((|\nu|^2 + \nu_2^2\cos 2\nu_1\delta)^2 + \nu_1^4\right).$$

$$\operatorname{Tr} M = -2\frac{e^{-2\nu_2\delta}}{\nu_1^2}\left(|\nu|^2 + \nu_2^2\cos 2\nu_1\delta\right),$$

and the determinant, equal to $\det M = -e^{-4\nu_2\delta}$, is negative.
Therefore there are two opposite sign eigenvalues and the spectral radius of the matrix M is equal to

$$\rho(\xi) = \frac{e^{-2\nu_2\delta}}{\nu_1^2}\left(|\nu|^2 + \nu_2^2\cos 2\nu_1\delta + \sqrt{(|\nu|^2 + \nu_2^2\cos 2\nu_1\delta)^2 + \nu_1^4}\right).$$

For large ξ, $\nu_1(\xi) \sim \xi$ is large and $\nu_2(\xi) \sim 1/(2\xi\sqrt{\alpha})$ is small. The spectral radius $\rho(\xi)$ is considered as

$$\rho(\xi) \sim e^{-\delta/(\xi\sqrt{\alpha})}(1+\sqrt{2}) > 1.$$

With this asymptotics computation, we can conclude that there exists large ξ where the convergence factor with respect to ξ is larger than 1 when δ is small and therefore the algorithm is divergent. □

So it seems that the algorithm is not suitable for infinite time domains. We now turn to finite time domains, and see if we can forge a converging algorithm.

3.2.3 The finite time domain case

The aim of this section is to prove the following result:

Theorem 3.2.2. *The convergence of the algorithm is defined by the iteration matrix $M = M_1M_2$, with $M_1 = \tilde{S}_1(\delta+T_1)^{-1}S_1(\delta-T_2)$ and $M_2 = \tilde{S}_2^{-1}(-T_2)S_2(T_1)$, with*

$$\tilde{S}_1(t) = \begin{pmatrix} \operatorname{Re} c(t) & \operatorname{Re}\nu^{-1}s(t) \\ -\operatorname{Re}\nu s(t) + \frac{\gamma}{\sqrt{\alpha}}\operatorname{Im} c(t) & \operatorname{Re} c(t) + \frac{\gamma}{\sqrt{\alpha}}\operatorname{Im}\nu^{-1}s(t) \end{pmatrix}, \quad (3.2.28)$$

$$S_1(t) = \begin{pmatrix} \sqrt{\alpha}\operatorname{Im} c(t) + \gamma\operatorname{Re}\nu^{-1}s(t) & \sqrt{\alpha}\operatorname{Im}\nu^{-1}s(t) \\ -\sqrt{\alpha}\operatorname{Im}\nu s(t) + \frac{\gamma^2}{\sqrt{\alpha}}\operatorname{Im}\nu^{-1}s(t) & \sqrt{\alpha}\operatorname{Im} c(t) - \gamma\operatorname{Re}\nu^{-1}s(t) \end{pmatrix}, \quad (3.2.29)$$

$$\tilde{S}_2(t) = \begin{pmatrix} \operatorname{Re} c(t) - \frac{\gamma}{\sqrt{\alpha}} \operatorname{Im} \nu^{-1} s(t) & \operatorname{Re} \nu^{-1} s(t) \\ -\operatorname{Re} \nu s(t) - \frac{\gamma}{\sqrt{\alpha}} \operatorname{Im} c(t) & \operatorname{Re} c(t) \end{pmatrix}, \tag{3.2.30}$$

$$S_2(t) = \frac{1}{\sqrt{\alpha}} \begin{pmatrix} -\operatorname{Im} c(t) & -\operatorname{Im} \nu^{-1} s(t) \\ \operatorname{Im} \nu s(t) & -\operatorname{Im} c(t) \end{pmatrix}. \tag{3.2.31}$$

In each subinterval in time, we have the matrix wave equations (3.2.2) and hence (3.2.3). The solutions are in vector form $\hat{Y}_1^n$ defined on $(-T_1, \delta)$ and $\hat{Y}_2^n$ defined on $(0, T_2)$. Rewrite the boundary conditions on the coefficients $\hat{Y}_j^n(k,t) = (\hat{y}_j^n(k,t), \hat{\lambda}_j^n(k,t))$:

$$\begin{cases} \hat{y}_1^n(k,-T_1) = \partial_t \hat{y}_1^n(k,-T_1) = 0, \\ \hat{\lambda}_1^{n+1}(k,\delta) = \hat{\lambda}_2^n(k,\delta), \\ \partial_t \hat{\lambda}_1^{n+1}(k,\delta) - \gamma \hat{y}_1^{n+1}(k,\delta) = \partial_t \hat{\lambda}_2^n(k,\delta) - \gamma \hat{y}_2^n(k,\delta), \end{cases}$$
$$\begin{cases} \hat{\lambda}_2^n(k,T_2) = \partial_t \hat{\lambda}_2^n(k,T_2) - \gamma \hat{y}_2^n(k,T_2) = 0, \\ \hat{y}_2^{n+1}(k,0) = \hat{y}_1^{n+1}(k,0), \partial_t \hat{y}_2^{n+1}(k,0) = \partial_t \hat{y}_1^{n+1}(k,0). \end{cases} \tag{3.2.32}$$

We now write the iterates in the convenient form

$$\begin{cases} \hat{Y}_1^n(k,t) = C(t+T_1)\tilde{Y}_1^n + N^{-1} S(t+T_1) \underline{Y}_1^n, \\ \hat{Y}_2^n(k,t) = C(t-T_2)\tilde{Y}_2^n + N^{-1} S(t-T_2) \underline{Y}_2^n. \end{cases} \tag{3.2.33}$$

and we will use formulas (3.2.12) for the derivatives:

$$\begin{cases} \partial_t \hat{Y}_1^n(k,t) = -NS(t+T_1)\tilde{Y}_1^n + C(t+T_1) \underline{Y}_1^n, \\ \partial_t \hat{Y}_2^n(k,t) = -NS(t-T_2)\tilde{Y}_2^n + C(t-T_2) \underline{Y}_2^n. \end{cases} \tag{3.2.34}$$

From the condition $\hat{y}_1^n(k,-T_1) = 0$ we get

$$(\tilde{Y}_1^n)_1 = 0.$$

From the condition $\partial_t \hat{y}_1^n(k,-T_1) = 0$ we get

$$(\underline{Y}_1^n)_1 = 0.$$

It remains only two parameters,

$$\tilde{Y}_1^n = \alpha_1^n e_2, \quad \underline{Y}_1^n = \beta_1^n e_2,$$

with $e_1 = (1,0)^T$, $e_2 = (0,1)^T$ and $\hat{Y}_1^n$ takes the form

$$\hat{Y}_1^n(k,t) = \alpha_1^n C(t+T_1)e_2 + \beta_1^n N^{-1} S(t+T_1)e_2. \tag{3.2.35}$$

We will also use

$$\partial_t \hat{Y}_1^n(k,t) = -\alpha_1^n N S(t+T_1)e_2 + \beta_1^n C(t+T_1)e_2. \tag{3.2.36}$$

For the domain $(0,T_2)$, from the condition $\hat{\lambda}_2^n(k,T_2) = 0$ we get

$$(\tilde{Y}_2^n)_2 = 0.$$

From the condition $\partial_t \hat{\lambda}_2^n(k,T_2) - \gamma \hat{y}_2^n(k,T_2) = 0$ we get

$$(\underline{Y}_2^n)_2 - \gamma(\tilde{Y}_2^n)_1 = 0.$$

Define

$$\alpha_2^n = (\tilde{Y}_2^n)_1, \quad \beta_2^n = (\underline{Y}_2^n)_1$$

Then $(\underline{Y}_2^n)_2 = \gamma\alpha_2^n$, and we can write

$$\tilde{Y}_2^n = \alpha_2^n e_1, \quad \underline{Y}_2^n = \beta_2^n e_1 + \gamma\alpha_2^n e_2.$$

$$\hat{Y}_2^n(k,t) = \alpha_2^n C(t-T_2)e_1 + \beta_2^n N^{-1} S(t-T_2)e_1 + \gamma\alpha_2^n N^{-1} S(t-T_2)e_2. \tag{3.2.37}$$

We will also use

$$\partial_t \hat{Y}_2^n(k,t) = -\alpha_2^n N S(t-T_2)e_1 + \beta_2^n C(t-T_2)e_1 + \gamma\alpha_2^n C(t-T_2)e_2. \tag{3.2.38}$$

From formulas (3.2.10),

$$\hat{Y}_1^n(k,t) = \alpha_1^n(\operatorname{Re} c(t+T_1)I + \operatorname{Im} c(t+T_1)J)e_2 + \beta_1^n(\operatorname{Re}\nu^{-1}s(t+T_1)I + \operatorname{Im}\nu^{-1}s(t+T_1)J)e_2.$$

Since $Je_2 = -\frac{1}{\sqrt{\alpha}}e_1$ and $Je_1 = \sqrt{\alpha}e_2$, we can rewrite

$$\hat{Y}_1^n(k,t) = \alpha_1^n(\operatorname{Re} c(t+T_1)e_2 - \frac{1}{\sqrt{\alpha}}\operatorname{Im} c(t+T_1)e_1) + \beta_1^n(\operatorname{Re}\nu^{-1}s(t+T_1)e_2 - \frac{1}{\sqrt{\alpha}}\operatorname{Im}\nu^{-1}s(t+T_1)e_1).$$

Separate now the components:

$$\begin{aligned} \hat{y}_1^n(k,t) &= -\alpha_1^n \tfrac{1}{\sqrt{\alpha}}\operatorname{Im} c(t+T_1) - \tfrac{1}{\sqrt{\alpha}}\beta_1^n \operatorname{Im}\nu^{-1}s(t+T_1), \\ \hat{\lambda}_1^n(k,t) &= \alpha_1^n \operatorname{Re} c(t+T_1) + \beta_1^n \operatorname{Re}\nu^{-1}s(t+T_1). \end{aligned} \tag{3.2.39}$$

For $\partial_t Y_1$ we replace c by $-\nu s$ and $\nu^{-1}s$ by c.

$$\left\{\begin{array}{l} \partial_t \hat{y}_1^n(k,t) = \alpha_1^n \frac{1}{\sqrt{\alpha}} \operatorname{Im} \nu s(t+T_1) - \frac{1}{\sqrt{\alpha}} \beta_1^n \operatorname{Im} c(t+T_1), \\ \partial_t \hat{\lambda}_1^n(k,t) = -\alpha_1^n \operatorname{Re} \nu s(t+T_1) + \beta_1^n \operatorname{Re} c(t+T_1). \end{array}\right. \tag{3.2.40}$$

In subdomain 2,

$$\hat{Y}_2^n(k,t) = \alpha_2^n((\operatorname{Re} c(t-T_2)I + \operatorname{Im} c(t-T_2)J)e_1 + \beta_2^n(\operatorname{Re} \nu^{-1}s(t-T_2)I + \operatorname{Im} \nu^{-1}s(t-T_2)J)e_1 \\ + \gamma\alpha_2^n(\operatorname{Re} \nu^{-1}s(t-T_2)I + \operatorname{Im} \nu^{-1}s(t-T_2)J)e_2).$$

which is rewritten as

$$\hat{Y}_2^n(k,t) = \alpha_2^n((\operatorname{Re} c(t-T_2)e_1 + \sqrt{\alpha} \operatorname{Im} c(t-T_2)e_2) + \beta_2^n(\operatorname{Re} \nu^{-1}s(t-T_2)e_1 + \sqrt{\alpha} \operatorname{Im} \nu^{-1}s(t-T_2)e_2) \\ + \gamma\alpha_2^n(\operatorname{Re} \nu^{-1}s(t-T_2)e_2 - \frac{1}{\sqrt{\alpha}} \operatorname{Im} \nu^{-1}s(t-T_2)e_1).$$

$$\left\{\begin{array}{l} \hat{y}_2^n(k,t) = \alpha_2^n(\operatorname{Re} c(t-T_2) - \frac{\gamma}{\sqrt{\alpha}} \operatorname{Im} \nu^{-1}s(t-T_2)) + \beta_2^n \operatorname{Re} \nu^{-1}s(t-T_2), \\ \hat{\lambda}_2^n(k,t) = \alpha_2^n(\sqrt{\alpha} \operatorname{Im} c(t-T_2) + \gamma \operatorname{Re} \nu^{-1}s(t-T_2)) + \beta_2^n \sqrt{\alpha} \operatorname{Im} \nu^{-1}s(t-T_2). \end{array}\right. \tag{3.2.41}$$

For the time derivative replace c by $-\nu s$, and $\nu^{-1}s$ by c.

$$\left\{\begin{array}{l} \partial_t \hat{y}_2^n(k,t) = \alpha_2^n(-\operatorname{Re} \nu s(t-T_2) - \frac{\gamma}{\sqrt{\alpha}} \operatorname{Im} c(t-T_2)) + \beta_2^n \operatorname{Re} c(t-T_2) \\ \partial_t \hat{\lambda}_2^n(k,t) = \alpha_2^n(-\sqrt{\alpha} \operatorname{Im} \nu s(t-T_2) + \gamma \operatorname{Re} c(t-T_2)) + \beta_2^n \sqrt{\alpha} \operatorname{Im} c(t-T_2) \end{array}\right. \tag{3.2.42}$$

Now impose the conditions at $t = \delta$,

$$\left\{\begin{array}{l} \hat{\lambda}_1^{n+1}(k,\delta) = \hat{\lambda}_2^n(k,\delta), \\ \partial_t \hat{\lambda}_1^{n+1}(k,\delta) - \gamma \hat{y}_1^{n+1}(k,\delta) = \partial_t \hat{\lambda}_2^n(k,\delta) - \gamma \hat{y}_2^n(k,\delta), \end{array}\right.$$

Write

$$\partial_t \hat{\lambda}_1^n(k,t) - \gamma \hat{y}_1^n(k,t) = \alpha_1^n(-\operatorname{Re} \nu s(t+T_1) + \gamma \operatorname{Im} c(t+T_1)/\sqrt{\alpha}) \\ + \beta_1^n(\operatorname{Re} c(t+T_1) + \gamma \operatorname{Im} \nu^{-1}s(t+T_1)/\sqrt{\alpha}).$$

$$\partial_t \hat{\lambda}_2^n(k,t) - \gamma \hat{y}_2^n(k,t) = \alpha_2^n(-\sqrt{\alpha} \operatorname{Im} \nu s(t-T_2) + \gamma^2 \operatorname{Im} \nu^{-1}s(t-T_2)/\sqrt{\alpha}) \\ + \beta_2^n(\sqrt{\alpha} \operatorname{Im} c(t-T_2) - \gamma \operatorname{Re} \nu^{-1}s(t-T_2)).$$

So the transmission conditions at $t=\delta$ give

$$\begin{aligned}
&\alpha_1^{n+1}\operatorname{Re} c(\delta+T_1)+\beta_1^{n+1}\operatorname{Re}\nu^{-1}s(\delta+T_1)\\
&\quad=\alpha_2^n(\sqrt{\alpha}\operatorname{Im} c(\delta-T_2)+\gamma\operatorname{Re}\nu^{-1}s(\delta-T_2))+\sqrt{\alpha}\beta_2^n\operatorname{Im}\nu^{-1}s(\delta-T_2),\\
&\alpha_1^{n+1}(-\operatorname{Re}\nu s(\delta+T_1)+\gamma\operatorname{Im} c(\delta+T_1)/\sqrt{\alpha})+\beta_1^{n+1}(\operatorname{Re} c(\delta+T_1)+\gamma\operatorname{Im}\nu^{-1}s(\delta+T_1)/\sqrt{\alpha})\\
&=\alpha_2^n(-\sqrt{\alpha}\operatorname{Im}\nu s(\delta-T_2)+\gamma^2/\sqrt{\alpha}\operatorname{Im}\nu^{-1}s(\delta-T_2))+\beta_2^n(\sqrt{\alpha}\operatorname{Im} c(\delta-T_2)-\gamma\operatorname{Re}\nu^{-1}s(\delta-T_2)).
\end{aligned}\tag{3.2.43}$$

Then we have the recursion

$$\tilde{S}_1(\delta+T_1)\begin{pmatrix}\alpha_1^{n+1}\\ \beta_1^{n+1}\end{pmatrix}=S_1(\delta-T_2)\begin{pmatrix}\alpha_2^{n}\\ \beta_2^{n}\end{pmatrix}. \tag{3.2.44}$$

Lemma 3. *For any $t\neq 0$, the determinant of $\tilde{S}_1(t)$ is equal to*

$$\det\tilde{S}_1(t)=\frac{\nu_1^2\cosh^2(\nu_2 t)+\nu_2^2\cos^2(\nu_1 t)}{|\nu|^2}-\frac{\gamma}{\sqrt{\alpha}}\operatorname{Im}\nu^{-1}s(t)\overline{c(t)}. \tag{3.2.45}$$

If it is different from 0 for $t=T_1+\delta$, the system (3.2.44) can be solved into

$$\begin{pmatrix}\alpha_1^{n+1}\\ \beta_1^{n+1}\end{pmatrix}=\tilde{S}_1^{-1}(\delta+T_1)S_1(\delta-T_2)\begin{pmatrix}\alpha_2^{n}\\ \beta_2^{n}\end{pmatrix} \tag{3.2.46}$$

Proof.

$$\det\tilde{S}_1(t)=(\operatorname{Re} c(t))^2+\operatorname{Re}\nu s(t)\operatorname{Re}\nu^{-1}s(t)+\frac{\gamma}{\sqrt{\alpha}}(\operatorname{Re} c(t)\operatorname{Im}\nu^{-1}s(t)-\operatorname{Im} c(t)\operatorname{Re}\nu^{-1}s(t)).$$

Use

$$\operatorname{Re}\nu s(t)=\nu_1\operatorname{Re} s(t)-\nu_2\operatorname{Im} s(t),\quad \operatorname{Re}\nu^{-1}s(t)=\frac{\nu_1\operatorname{Re} s(t)+\nu_2\operatorname{Im} s(t)}{|\nu|^2}.$$

to obtain

$$\det\tilde{S}_1(t)=(\operatorname{Re} c(t))^2+\frac{\nu_1^2(\operatorname{Re} s(t))^2-\nu_2^2(\operatorname{Im} s(t))^2}{|\nu|^2}-\frac{\gamma}{\sqrt{\alpha}}\operatorname{Im}\nu^{-1}s(t)\overline{c(t)}.$$

Expand the first term into

$$\frac{\nu_1^2}{|\nu|^2}((\operatorname{Re} c(t))^2+(\operatorname{Re} s(t))^2)+\frac{\nu_2^2}{|\nu|^2}((\operatorname{Re} c(t))^2-(\operatorname{Im} s(t))^2),$$

and from (3.2.8),

$$(\operatorname{Re} c(t))^2+(\operatorname{Re} s(t))^2=\cosh^2(\nu_2 t),\quad (\operatorname{Re} c(t))^2-(\operatorname{Re} s(t))^2=\cos^2(\nu_1 t),$$

which gives formula (3.2.45). □

We turn now to the second transmission condition, which is

$$\hat{y}_2^{n+1}(k,0) = \hat{y}_1^{n+1}(k,0), \quad \partial_t \hat{y}_2^{n+1}(k,0) = \partial_t \hat{y}_1^{n+1}(k,0).$$

We write it

$$\begin{aligned}\alpha_2^{n+1} \operatorname{Re} c(-T_2) + \beta_2^{n+1} \operatorname{Re} \nu^{-1} s(-T_2) - \frac{\gamma}{\sqrt{\alpha}} \alpha_2^{n+1} \operatorname{Im} \nu^{-1} s(-T_2) \\ = -\alpha_1^{n+1} \operatorname{Im} c(T_1)/\sqrt{\alpha} - \beta_1^{n+1} \operatorname{Im} \nu^{-1} s(T_1)/\sqrt{\alpha}, \\ -\alpha_2^{n+1} \operatorname{Re} \nu s(-T_2) + \beta_2^{n+1} \operatorname{Re} c(-T_2) - \frac{\gamma}{\sqrt{\alpha}} \alpha_2^{n+1} \operatorname{Im} c(-T_2) \\ = \alpha_1^{n+1} \operatorname{Im} \nu s(T_1)/\sqrt{\alpha} - \beta_1^{n+1} \operatorname{Im} c(T_1)/\sqrt{\alpha}.\end{aligned}$$

Then we have the recursion

$$\tilde{S}_2(-T_2) \begin{pmatrix} \alpha_2^{n+1} \\ \beta_2^{n+1} \end{pmatrix} = S_2(T_1) \begin{pmatrix} \alpha_1^{n+1} \\ \beta_1^{n+1} \end{pmatrix}. \tag{3.2.47}$$

Lemma 4. *For any $t \neq 0$,*

$$\det \tilde{S}_2(t) = \frac{\nu_1^2 \cosh^2(\nu_2 t) + \nu_2^2 \cos^2(\nu_1 t)}{|\nu|^2} + \frac{\gamma}{\sqrt{\alpha}} \operatorname{Im} \nu^{-1} s(t) \overline{c(t)}. \tag{3.2.48}$$

For any T_2, α, δ and $\gamma = 0$, $\det \tilde{S}_2(-T_2)$ is different from 0 and the system (3.2.47) can be written as

$$\begin{pmatrix} \alpha_2^{n+1} \\ \beta_2^{n+1} \end{pmatrix} = \tilde{S}_2^{-1}(-T_2) S_2(T_1) \begin{pmatrix} \alpha_1^{n+1} \\ \beta_1^{n+1} \end{pmatrix} \tag{3.2.49}$$

Proof.

$$\det \tilde{S}_2(t) = (\operatorname{Re} c(t))^2 + \operatorname{Re} \nu s(t) \operatorname{Re} \nu^{-1} s(t) - \frac{\gamma}{\sqrt{\alpha}} (\operatorname{Re} c(t) \operatorname{Im} \nu^{-1} s(t) - \operatorname{Im} c(t) \operatorname{Re} \nu^{-1} s(t)).$$

We proceed in the same way as lemma (3) to obtain

$$\det \tilde{S}_2(t) = (\operatorname{Re} c(t))^2 + \frac{\nu_1^2 (\operatorname{Re} s(t))^2 - \nu_2^2 (\operatorname{Im} s(t))^2}{|\nu|^2} - \frac{\gamma}{\sqrt{\alpha}} \operatorname{Im} \nu^{-1} s(t) \overline{c(t)}.$$

From the proof of lemma (3), we get

$$\frac{\nu_1^2}{|\nu|^2} ((\operatorname{Re} c(t))^2 + (\operatorname{Re} s(t))^2) + \frac{\nu_2^2}{|\nu|^2} ((\operatorname{Re} c(t))^2 - (\operatorname{Im} s(t))^2),$$

and

$$(\operatorname{Re} c(t))^2 + (\operatorname{Re} s(t))^2 = \cosh^2(\nu_2 t), \quad (\operatorname{Re} c(t))^2 - (\operatorname{Re} s(t))^2 = \cos^2(\nu_1 t),$$

which gives formula (3.2.48). □

For the moment we have no hint as when the determinants are different from 0, except when $\gamma = 0$. We finally have the recursion equation.

$$\begin{cases} \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix}^{n+1} = M_1 \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix}^{n}, \\ \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix}^{n+1} = M_2 \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix}^{n+1}. \end{cases} \tag{3.2.50}$$

with $M_1 = \tilde{S}_1(\delta + T_1)^{-1} S_1(\delta - T_2)$ and $M_2 = \tilde{S}_2^{-1}(-T_2) S_2(T_1)$. Define

$$M = M_1 M_2, \tag{3.2.51}$$

then the recursion relation becomes

$$\begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix}^{n+1} = M \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix}^{n}.$$

If the matrices $\tilde{S}_1$ and $\tilde{S}_2$ are invertible, then we can iterate and the convergence depends of the spectral radius of M. Even though the general existence theorem for the subproblems implies that those matrices are invertible, we have no hint up to now on how to deal with γ. Therefore we study now the case $\gamma = 0$, where the matrices are much simpler. If furthermore $\delta = 0$: no overlap, and the two windows are equal, the matrices simplify greatly, depending only of one parameter $T_1 = T_2$. We call

$$A_1 = \tilde{S}_1(T_1),\ A_2 = S_1(-T_1),\ B_1 = \tilde{S}_2(-T_1),\ B_2 = S_2(T_1).$$

3.3 Study of the toy model

The toy model is as described above $\gamma = 0$, $\delta = 0$, $T_1 = T_2$, and also $\alpha = 1$. We are aware that this problem is only academic. Indeed, in practice the parameter α is often a regularization term, that could be very small, see Remark 1. In this case, the formulas (3.2.5) for μ and ν simplify into

$$\begin{aligned}
&k \geq 1, \quad \xi(k) = \frac{k\pi}{b-a}, \quad \mu(k) = \xi^2(k) + i, \quad \nu(k) = \sqrt{\mu(k)} = \nu_1 + i\nu_2,\\
&\nu_1^2 - \nu_2^2 = \xi^2(k), \quad 2\nu_1\nu_2 = 1,\\
&\nu_1 = \frac{1}{\sqrt{2}}\sqrt{\xi^2(k) + \sqrt{\xi^4(k)+1}}.
\end{aligned} \tag{3.3.1}$$

In particular ν_1 is an increasing function of k, and ν_2 a decreasing function of k. Furthermore the matrices defining convergence are now

$$A_1 = \begin{pmatrix} \operatorname{Re} c(T_1) & \operatorname{Re} \nu^{-1} s(T_1) \\ -\operatorname{Re} \nu s(T_1) & \operatorname{Re} c(T_1) \end{pmatrix}, \; A_2 = \begin{pmatrix} \operatorname{Im} c(T_1) & -\operatorname{Im} \nu^{-1} s(T_1) \\ \operatorname{Im} \nu s(T_1) & \operatorname{Im} c(T_1) \end{pmatrix}, \tag{3.3.2}$$

$$B_1 = \begin{pmatrix} \operatorname{Re} c(T_1) & -\operatorname{Re} \nu^{-1} s(T_1) \\ \operatorname{Re} \nu s(T_1) & \operatorname{Re} c(T_1) \end{pmatrix}, \; B_2 = \begin{pmatrix} -\operatorname{Im} c(T_1) & -\operatorname{Im} \nu^{-1} s(T_1) \\ \operatorname{Im} \nu s(T_1) & -\operatorname{Im} c(T_1) \end{pmatrix}. \tag{3.3.3}$$

We note $M_1 = A_1^{-1}A_2$ and $M_2 = B_1^{-1}B_2$. From lemmas 3 and 4, the matrices A_1 and B_1 have the same determinant equal to

$$d(T_1) = \frac{\nu_1^2 \cosh^2(\nu_2 T_1) + \nu_2^2 \cos^2(\nu_1 T_1)}{|\nu|^2}. \tag{3.3.4}$$

Note $A = d(T_1)M_1$ and $B = d(T_1)M_2$. Then

$$A = \begin{pmatrix} \operatorname{Re} c(T_1) & -\operatorname{Re} \nu^{-1} s(T_1) \\ \operatorname{Re} \nu s(T_1) & \operatorname{Re} c(T_1) \end{pmatrix} \times \begin{pmatrix} \operatorname{Im} c(T_1) & -\operatorname{Im} \nu^{-1} s(T_1) \\ \operatorname{Im} \nu s(T_1) & \operatorname{Im} c(T_1) \end{pmatrix},$$

which gives

$$A = \begin{pmatrix} \operatorname{Re} c(T_1) \operatorname{Im} c(T_1) - \operatorname{Re} \nu^{-1} s(T_1) \operatorname{Im} \nu s(T_1) & -\operatorname{Re} c(T_1) \operatorname{Im} \nu^{-1} s(T_1) - \operatorname{Im} c(T_1) \operatorname{Re} \nu^{-1} s(T_1) \\ \operatorname{Re} \nu s(T_1) \operatorname{Im} c(T_1) + \operatorname{Re} c(T_1) \operatorname{Im} \nu s(T_1) & -\operatorname{Re} \nu s(T_1) \operatorname{Im} \nu^{-1} s(T_1) + \operatorname{Re} c(T_1) \operatorname{Im} c(T_1) \end{pmatrix}.$$

$$B = \begin{pmatrix} \operatorname{Re} c(T_1) & \operatorname{Re} \nu^{-1} s(T_1) \\ -\operatorname{Re} \nu s(T_1) & \operatorname{Re} c(T_1) \end{pmatrix} \times \begin{pmatrix} ,-\operatorname{Im} c(T_1) & -\operatorname{Im} \nu^{-1} s(T_1) \\ \operatorname{Im} \nu s(T_1) & -\operatorname{Im} c(T_1) \end{pmatrix}$$

which gives

$$B = \begin{pmatrix} -\operatorname{Re} c(T_1) \operatorname{Im} c(T_1) + \operatorname{Re} \nu^{-1} s(T_1) \operatorname{Im} \nu s(T_1) & -\operatorname{Re} c(T_1) \operatorname{Im} \nu^{-1} s(T_1) - \operatorname{Im} c(T_1) \operatorname{Re} \nu^{-1} s(T_1) \\ \operatorname{Re} \nu s(T_1) \operatorname{Im} c(T_1) + \operatorname{Re} c(T_1) \operatorname{Im} \nu s(T_1) & \operatorname{Re} \nu s(T_1) \operatorname{Im} \nu^{-1} s(T_1) - \operatorname{Re} c(T_1) \operatorname{Im} c(T_1) \end{pmatrix}.$$

Finally, we have

$$M = \frac{1}{d(T_1)^2} AB \tag{3.3.5}$$

where, defining the coefficients of A to be a_{ij},

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad B = \begin{pmatrix} -a_{11} & a_{12} \\ a_{21} & -a_{22} \end{pmatrix},$$

therefore

$$R := AB = \begin{pmatrix} -a_{11}^2 + a_{12}a_{21} & a_{12}(a_{11} - a_{22}) \\ -a_{21}(a_{11} - a_{22}) & -a_{22}^2 + a_{12}a_{21} \end{pmatrix}. \tag{3.3.6}$$

The coefficients are

$$\begin{cases} a_{11} = -\dfrac{\sin(2\nu_1 T_1)\sinh(2\nu_2 T_1)}{2} + \dfrac{\nu_1\nu_2}{2|\nu|^2}(\cos(2\nu_1 T_1) - \cosh(2\nu_2 T_1)), \\ a_{12} = \dfrac{-\nu_1\cos(2\nu_1 T_1)\sinh(2\nu_2 T_1) + \nu_2\cosh(2\nu_2 T_1)\sin(2\nu_1 T_1)}{2|\nu|^2}, \\ a_{21} = \dfrac{\nu_1\cos(2\nu_1 T_1)\sinh(2\nu_2 T_1) + \nu_2\cosh(2\nu_2 T_1)\sin(2\nu_1 T_1)}{2}, \\ a_{22} = -\dfrac{\sin(2\nu_1 T_1)\sinh(2\nu_2 T_1)}{2} + \dfrac{\nu_1\nu_2}{|\nu|^2}(\cosh(2\nu_2 T_1) - \cos(2\nu_1 T_1)), \end{cases} \tag{3.3.7}$$

and since $2\nu_1\nu_2 = 1$,

$$\begin{cases} a_{11} = -\dfrac{\sin(2\nu_1 T_1)\sinh(2\nu_2 T_1)}{2} - \dfrac{1}{4|\nu|^2}(\cosh(2\nu_2 T_1) - \cos(2\nu_1 T_1)), \\ a_{12} = \dfrac{-\nu_1\cos(2\nu_1 T_1)\sinh(2\nu_2 T_1) + \nu_2\cosh(2\nu_2 T_1)\sin(2\nu_1 T_1)}{2|\nu|^2}, \\ a_{21} = \dfrac{\nu_1\cos(2\nu_1 T_1)\sinh(2\nu_2 T_1) + \nu_2\cosh(2\nu_2 T_1)\sin(2\nu_1 T_1)}{2}, \\ a_{22} = -\dfrac{\sin(2\nu_1 T_1)\sinh(2\nu_2 T_1)}{2} + \dfrac{1}{4|\nu|^2}(\cosh(2\nu_2 T_1) - \cos(2\nu_1 T_1)). \end{cases} \tag{3.3.8}$$

3.3.1 Analysis of the eigenvalues

Theorem 3.3.1. *The eigenvalues of M (see (3.2.51) and (3.3.5)) are real negative, given by*

$$\begin{aligned}
\lambda^{\pm} &= -\frac{1}{d^2}\big[\sqrt{\varphi} \pm \frac{1}{4|\nu|^2}(\cosh(2\nu_2 T_1) - \cos(2\nu_1 T_1))\big]^2, \\
\varphi &= \frac{1}{4|\nu|^2}\left[\nu_1^2 \sinh^2(2\nu_2 T_1) - \nu_2^2 \sin^2(2\nu_1 T_1)\right], \\
d &= \frac{\nu_1^2 \cosh^2(\nu_2 T_1) + \nu_2^2 \cos^2(\nu_1 T_1)}{|\nu|^2}.
\end{aligned} \tag{3.3.9}$$

Consequently, the spectral radius of the matrix M is

$$\rho(k, T_1) = \Big[\frac{\sqrt{\varphi} + \frac{1}{4|\nu|^2}(\cosh(2\nu_2 T_1) - \cos(2\nu_1 T_1))}{d}\Big]^2. \tag{3.3.10}$$

Proof. Since $M = \frac{1}{d^2}R$, we concentrate on R, using the coefficients in (3.3.6). Expand the characteristic polynomial of R

$$\begin{aligned}
&P(\lambda) = \lambda^2 - \Sigma\lambda + \Pi, \\
&\Sigma := \mathrm{Tr}\, R = 2a_{12}a_{21} - (a_{11}^2 + a_{22}^2), \\
&\Pi := \mathrm{Det}\, R = (a_{12}a_{21} - a_{11}a_{22})^2.
\end{aligned} \tag{3.3.11}$$

We see that the extradiagonal terms intervene only by their product, we call $C = a_{12}a_{21}$.

$$P(\lambda) = (\lambda - (C - \frac{1}{2}(a_{11}^2 + a_{22}^2)))^2 - \Phi, \quad \Phi = (C - \frac{1}{2}(a_{11}^2 + a_{22}^2))^2 - (C - a_{11}a_{22})^2. \tag{3.3.12}$$

Lemma 5. *For all k and T_1, Φ is positive.*

Expand Φ using $a^2 - b^2 = (a - b)(a + b)$,

$$\begin{aligned}
\Phi &= \big[C - \tfrac{1}{2}(a_{11}^2 + a_{22}^2) + C - a_{11}a_{22}\big]\big[C - \tfrac{1}{2}(a_{11}^2 + a_{22}^2) - (C - a_{11}a_{22})\big], \\
&= \big[2C - \tfrac{1}{2}(a_{11}^2 + a_{22}^2) - a_{11}a_{22}\big]\big[-\tfrac{1}{2}(a_{11}^2 + a_{22}^2) + a_{11}a_{22}\big], \\
&= \big[2C - \tfrac{1}{2}(a_{11} + a_{22})^2\big]\big[-\tfrac{1}{2}(a_{11} - a_{22})^2\big].
\end{aligned}$$

$$\Phi = -\frac{1}{2}(a_{11} - a_{22})^2(2C - \frac{1}{2}(a_{11} + a_{22})^2).$$

$$\Phi = (a_{11} - a_{22})^2\varphi, \quad \varphi = \frac{1}{4}(a_{11} + a_{22})^2 - C. \tag{3.3.13}$$

Insert now the formulas for the a_{ij} into φ, using for shortening.

$$\zeta_1 := 2\nu_1 T_1, \quad \zeta_2 := 2\nu_2 T_1.$$

$$a_{11} + a_{22} = -\sin(\zeta_1)\sinh(\zeta_2),$$

$$C := a_{12}a_{21} = \frac{1}{4|\nu|^2}(\nu_2^2\cosh(\zeta_2)\sin(\zeta_1) - \nu_1^2\cos(\zeta_1)\sinh(\zeta_2))$$

$$\begin{aligned}
\varphi &= \frac{1}{4}\sin^2(\zeta_1)\sinh^2(\zeta_2) - \frac{1}{4|\nu|^2}(\nu_2^2\cosh^2(\zeta_2)\sin^2(\zeta_1) - \nu_1^2\cos^2(\zeta_1)\sinh^2(\zeta_2)) \\
&= \frac{1}{4|\nu|^2}\left[|\nu|^2\sin^2(\zeta_1)\sinh^2(\zeta_2) - (\nu_2^2\cosh^2(\zeta_2)\sin^2(\zeta_1) - \nu_1^2\cos^2(\zeta_1)\sinh^2(\zeta_2))\right] \\
&= \frac{1}{4|\nu|^2}\left[\nu_1^2\sinh^2(\zeta_2) - \nu_2^2\sin^2(\zeta_1)\right] \\
&= \frac{\nu_1^2\nu_2^2}{4|\nu|^2}\left[\frac{\sinh^2(\zeta_2)}{\nu_2^2} - \frac{\sin^2(\zeta_1)}{\nu_1^2}\right] \\
&= \frac{4T_1^2\nu_1^2\nu_2^2}{4|\nu|^2}\left[\frac{\sinh^2(\zeta_2)}{\zeta_2^2} - \frac{\sin^2(\zeta_1)}{\zeta_1^2}\right] \\
&= \frac{T_1^2}{4|\nu|^2}\left[\frac{\sinh^2(\zeta_2)}{\zeta_2^2} - \frac{\sin^2(\zeta_1)}{\zeta_1^2}\right] > 0.
\end{aligned} \tag{3.3.14}$$

The term in the bracket is positive since the first term is larger than 1 and the second is smaller than 1. It is zero only if T_1 or k is zero, which is not in the range of values. Therefore we have proved that the eigenvalues are always real. Their product is Π defined in 3.3.11 which is positive. Therefore they are either both positive, or both negative. Let's find the sign of their sum $\Sigma = 2C - (a_{11}^2 + a_{22}^2)$.

$$\begin{aligned}
\Sigma &= \tfrac{1}{2|\nu|^2}\left[\nu_2^2\cosh^2(\zeta_2)\sin^2(\zeta_1) - \nu_1^2\cos^2(\zeta_1)\sinh^2(\zeta_2)\right] \\
&\quad -\left[\frac{\sin^2(\zeta_1)\sinh^2(\zeta_2)}{2} + \frac{1}{8|\nu|^4}(\cosh(\zeta_2) - \cos(\zeta_1))^2\right], \\
&= \tfrac{1}{2|\nu|^2}\left[\nu_2^2\cosh^2(\zeta_2)\sin^2(\zeta_1) - \nu_1^2\cos^2(\zeta_1)\sinh^2(\zeta_2) - |\nu|^2\sin^2(\zeta_1)\sinh^2(\zeta_2)\right] \\
&\quad -\frac{1}{8|\nu|^4}(\cosh(\zeta_2) - \cos(\zeta_1))^2, \\
&= \tfrac{1}{2|\nu|^2}(\nu_2^2\sin^2(\zeta_1) - \nu_1^2\sinh^2(\zeta_2))) - \frac{1}{8|\nu|^4}(\cosh(\zeta_2) - \cos(\zeta_1))^2, \\
&= -2\varphi - \frac{1}{8|\nu|^4}(\cosh(\zeta_2) - \cos(\zeta_1))^2 < 0.
\end{aligned}$$

Therefore the sum of the eigenvalues is negative, and there are both negative, given by

$$\begin{aligned}\lambda^{\pm} &= -\varphi - \frac{1}{16|\nu|^4}(\cosh(\zeta_2) - \cos(\zeta_1))^2 \pm \sqrt{\Phi}, \\ &= -\varphi - \frac{1}{16|\nu|^4}(\cosh(\zeta_2) - \cos(\zeta_1))^2 \pm |a_{11} - a_{22}|\sqrt{\varphi}.\end{aligned}$$

But

$$|a_{11} - a_{22}| = \frac{1}{2|\nu|^2}(\cosh(\zeta_2) - \cos(\zeta_1)),$$

and

$$\lambda^{\pm} = -\varphi - \frac{1}{16|\nu|^4}(\cosh(\zeta_2) - \cos(\zeta_1))^2 \pm \frac{1}{2|\nu|^2}(\cosh(\zeta_2) - \cos(\zeta_1))\sqrt{\varphi}.$$

$$a = \sqrt{\varphi}, \quad b = \frac{1}{4|\nu|^2}(\cosh(\zeta_2) - \cos(\zeta_1)),$$

$$\lambda^{\pm} = -a^2 - b^2 \pm 2ab = -(a \mp b)^2.$$

a and b are positive, therefore

$$max(|\lambda^{\pm}|) = (a + b)^2.$$

□

3.3.2 Numerical illustrations of the theoretical convergence factor

On figure 3.2, we have plotted the convergence factor $\rho(M)$ as a functions of the two variables ξ and T_1. On Figure 3.3, we have plotted $\rho(M)$ as a function of ξ for different values of T_1 (the right part being a zoom the figure is a zoom of the left one).

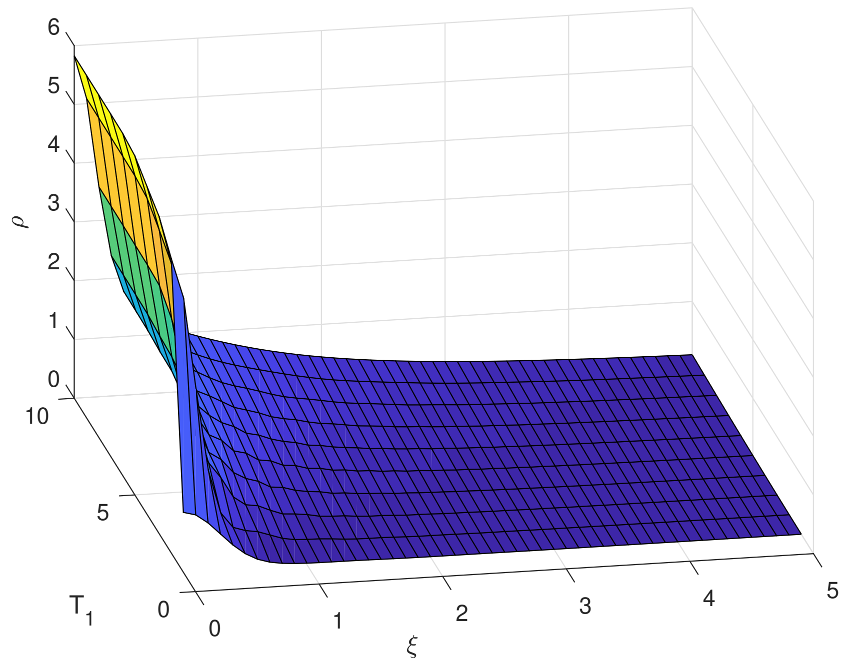


Figure 3.2: $\rho(M)$ as a function of ξ and T_1

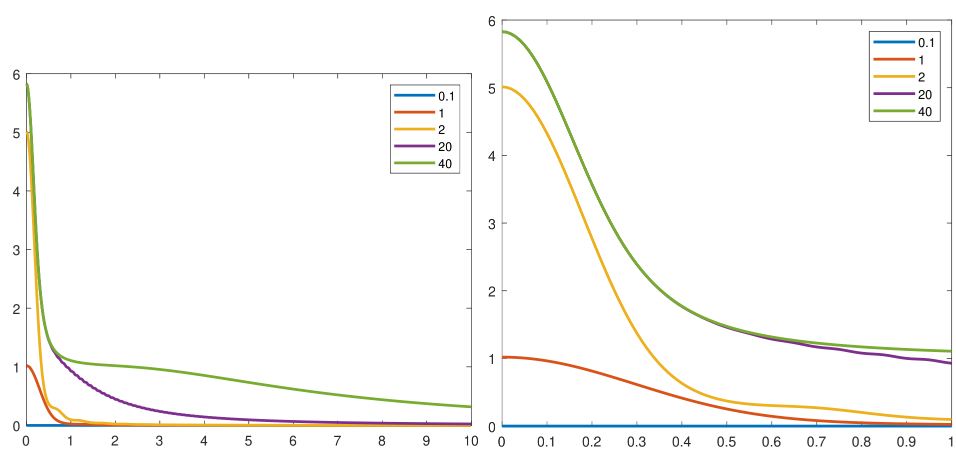


Figure 3.3: $\rho(M)$ as a function of ξ for different values of T_1: 0.1, 1, 2, 20, 40.

1. For a given T, the function $\xi \mapsto \rho(M)$ seems decreasing. As a result, the convergence factor corresponds to $\rho(M)$ computed at the smallest

value of ξ, namely $\xi_1 = \frac{\pi}{b-a}$ associated with $k = 1$. We have not been able to prove that. We use it as an empirical theorem. Because ξ_1 is proportional to the inverse of $(b-a)$, the smaller the space interval, the better the convergence is.

2. For a given ξ, the function $T \mapsto \rho(M)$ seems increasing but, here again, we have not been able to prove it. However, we prove in Theorem 3.3.2 and Theorem 3.3.3 that the convergence factor is smaller than one for small T and larger than one for T large.

3. For a given T, $\rho(M)$ tends to 0 as ξ goes to infinity. Indeed,

$$\lim_{\xi \to +\infty} \sqrt{\varphi} = \lim_{\xi \to +\infty} \frac{1}{4|\nu|^2}(\cosh(2\nu_2 T_1) - \cos(2\nu_1 T_1)) = 0$$

and

$$\lim_{\xi \to +\infty} d = 1.$$

As a result, the inherited algorithm acts as a smoother and is in theory very efficient for high frequencies. As a consequence, the convergence factor will a priori not deteriorate as the discretization step (in space and time) decreases. Indeed, a smaller space and time step leads to consider higher frequencies, that are efficiently treated by the algorithm. This result has to be temperated by numerical results (see Section 3.3.4) that show that in practice, the numerical convergence rate for high frequencies is different from the theoretical one (although the worst convergence rate is still obtained at the small frequency).

3.3.3 Theoretical convergence results

Theorem 3.3.2. *There exists $T_0 > 0$ such that, for any $T < T_0$, the algorithm converges.*

Proof. We first prove that $d(T_1) > 1$:

$$\begin{aligned}
d(T_1) &= \frac{\nu_1^2 \cosh^2(\nu_2 T_1) + \nu_2^2 \cos^2(\nu_1 T_1)}{\nu_1^2 + \nu_2^2} = \frac{\cosh^2(\nu_2 T_1) + \frac{\nu_2^2}{\nu_1^2} \cos^2(\nu_1 T_1)}{1 + \frac{\nu_2^2}{\nu_1^2}} \\
&= \frac{\cosh^2(\nu_2 T_1) + 4\nu_2^4 \cos^2(\nu_1 T_1)}{1 + 4\nu_2^4} = \frac{\cosh^2(\nu_2 T_1) + 4\nu_2^4 - 4\nu_2^4 \sin^2(\nu_1 T_1)}{1 + 4\nu_2^4} \\
&\geq \frac{\cosh^2(\nu_2 T_1) + 4\nu_2^4 - 4\nu_2^4 \nu_1^2 T_1^2}{1 + 4\nu_2^4} \geq \frac{\cosh^2(\nu_2 T_1) + 4\nu_2^4 - \nu_2^2 T_1^2}{1 + 4\nu_2^4} \\
&\geq \frac{1 + \nu_2^2 T_1^2 + \frac{\nu_2^4 T_1^4}{4} + 4\nu_2^4 - \nu_2^2 T_1^2}{1 + 4\nu_2^4} \\
&= \frac{1 + \frac{\nu_2^4 T_1^4}{4} + 4\nu_2^4}{1 + 4\nu_2^4} > 1.
\end{aligned}$$

Then, it is easily seen (see formula (3.3.9), reminding that φ is positive (3.3.14)) that

$$\varphi \leq \frac{1}{4} \sinh^2(2\nu_2 T_1),$$

and

$$\frac{1}{4|\nu|^2} \left(\cosh(2\nu_2 T_1) - \cos(2\nu_1 T_1)\right) \leq \frac{1}{4|\nu|^2} \left(\cosh(2\nu_2 T_1) + 1\right).$$

Therefore, we can somehow 'brutally' bound the convergence factor as follows:

$$\begin{aligned}
\sqrt{\rho(k, T_1)} &\leq \frac{1}{2} \sinh(2\nu_2 T_1) + \frac{1}{4|\nu|^2} \left(\cosh(2\nu_2 T_1) + 1\right) \\
&\leq \frac{1}{2} \sinh(2\nu_2(1) T_1) + \frac{1}{4|\nu(1)|^2} \left(\cosh(2\nu_2(1) T_1) + 1\right). \quad (3.3.15)
\end{aligned}$$

Here, we use the fact that $k \mapsto \nu_2(k)$ and $k \mapsto \frac{1}{|\nu(k)|^2}$ are both decaying. The function $f : \mathbb{R} \to \mathbb{R}$ defined by

$$f(T_1) = \frac{1}{2} \sinh(2\nu_2(1) T_1) + \frac{1}{4|\nu(1)|^2} \left(\cosh(2\nu_2(1) T_1) + 1\right),$$

is strictly increasing and continuous. Since $|\nu(1)| > 1$,

$$f(0) = \frac{1}{2|\nu(1)|^2} \leq \frac{1}{2} < 1.$$

Moreover

$$\lim_{T_1 \to +\infty} f(T_1) = +\infty,$$

and by the intermediate values Theorem, there exists $T_0 > 0$ such that, for any $T_1 < T_0$ ($T = 2T_1$), the algorithm converges. □

We notice that our upper-bound is not very sharp. On Figure 3.4, we plot $\sqrt{\rho}$ and its upper bound f with respect to T_1 in the case $a = 0$ and $b = 1$. We remark that for large T_1, f is much bigger than $\sqrt{\rho}$.

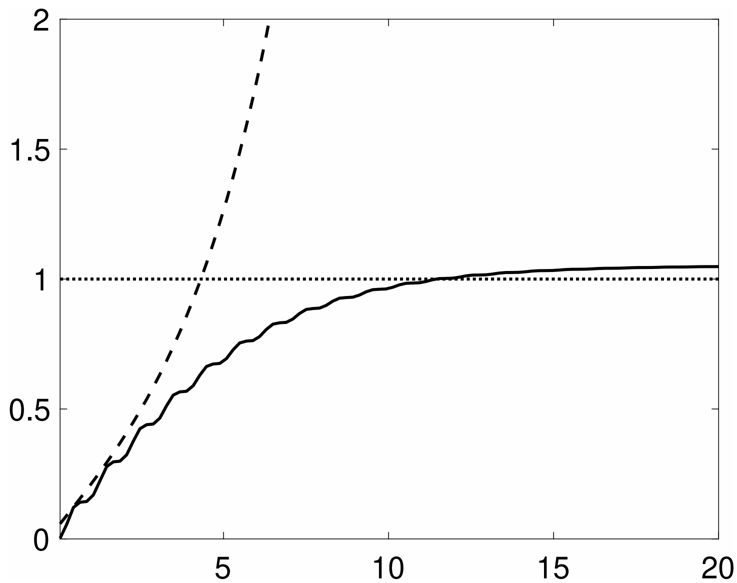


Figure 3.4: Evolution of $\sqrt{\rho}$ (plain) and f (dashed) with respect to T_1

Theorem 3.3.3. *For fixed k, as T_1 tends to infinity,*

$$\rho(M) \sim \Big(\frac{|\nu|}{\nu_1} + \frac{1}{2\nu_1^2}\Big)^2 > 1. \tag{3.3.16}$$

Therefore there exists a time $\bar{T}$ such that the algorithm diverges for $T > \bar{T}$.

Proof. If T_1 tends to infinity, $\cosh(\nu_2 T_1) \sim \sinh(\nu_2 T_1) \sim \frac{1}{2} e^{\nu_2 T_1}$. Similarly $\cosh(2\nu_2 T_1) \sim \frac{1}{2} e^{2\nu_2 T_1} \sim 2\cosh^2(\nu_2 T_1)$. Replace in (3.3.9):

$$
\begin{aligned}
&d \sim \frac{\nu_1^2 \cosh^2(\nu_2 T_1)}{|\nu|^2},\\
&\varphi \sim \frac{\nu_1^2}{4|\nu|^2} \sinh^2(2\nu_2 T_1),\\
&\lambda^{\pm} = -\frac{1}{d^2}\big[\sqrt{\varphi} \pm \frac{1}{4|\nu|^2}(\cosh(2\nu_2 T_1) - \cos(2\nu_1 T_1))\big]^2.
\end{aligned}
$$

Therefore

$$
\begin{aligned}
-\lambda^+ &\sim \frac{1}{d^2}\big[\frac{\nu_1}{2|\nu|}\sinh(2\nu_2 T_1) + \frac{1}{4|\nu|^2}\cosh(2\nu_2 T_1))\big]^2,\\
&\sim \left(\frac{\sinh(2\nu_2 T_1)}{d}\right)^2 \big[\frac{\nu_1}{2|\nu|} + \frac{1}{4|\nu|^2}\big]^2,\\
&\sim \left(\sinh(2\nu_2 T_1)\frac{|\nu|^2}{\nu_1^2\cosh^2(\nu_2 T_1)}\right)^2 \big[\frac{\nu_1}{2|\nu|} + \frac{1}{4|\nu|^2}\big]^2,\\
&\sim \left(2\frac{|\nu|^2}{\nu_1^2}\right)^2 \big[\frac{\nu_1}{2|\nu|} + \frac{1}{4|\nu|^2}\big]^2,\\
&= (\frac{|\nu|}{\nu_1} + \frac{1}{2\nu_1^2})^2 > 1.
\end{aligned}
$$

□

3.3.4 Numerical comparison

We finish this chapter by discussing the practical efficiency of the inherited algorithm. The discretization of the subproblems in domains 1 and 2 is based on the finite volume approach described in Section 2.2 (we remind that the subproblem have by construction the same structure as the initial one, see Section 3.1.2).

On Figure 3.5, we plot the theoretical and numerical convergence factor with respect to the frequency k. For the experiment, we choose $T = 1$, $b = 1$ and $a = 0$. We use $N = 51$ discretization points in space ($\Delta x = 1/50$) and $\Delta t = \Delta x$. For each frequency, the theoretical convergence factor is computed using Theorem 3.3.1. For the numerical one, we solve the homogeneous problem (without source, computing the zero solution) and we initialize the domain decomposition algorithm by the following mono-frequency initial guess:

$$
\lambda_2^0(x_j, T_1) = \sin(\frac{k\pi}{b-a}x_j), \quad \partial_t \lambda_2^0(x_j, T_1) = 0 \quad x_j = j\Delta x.
$$

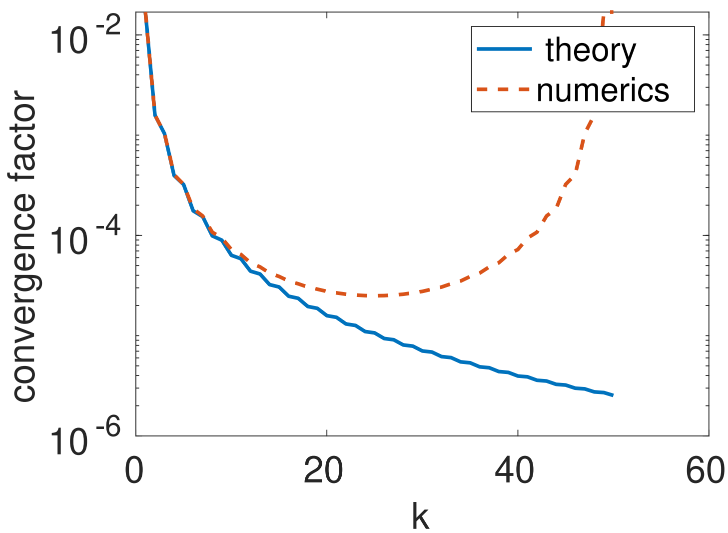


Figure 3.5: Numerical and theoretical convergence factor for $N = 51$

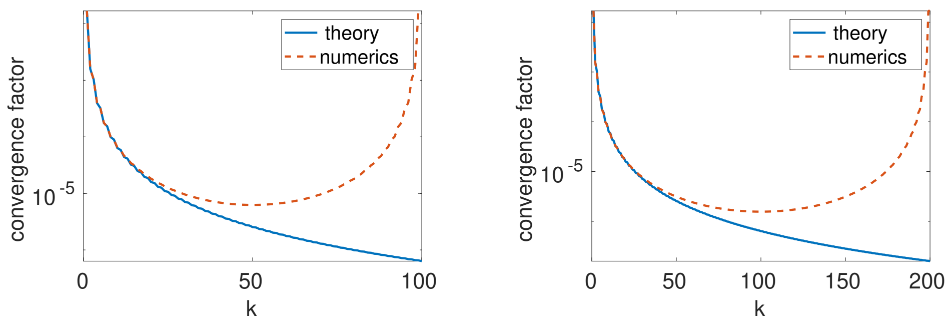
The convergence factor is then computed by making the quotient of the L^2 norm of the approximate solution between two consecutive iterations:

$$\rho_{\text{num}}(k) = \frac{\|y_{\Delta x}^{(j)}\|_{L^2((a,b)\times(0,T))}}{\|y_{\Delta x}^{(j-1)}\|_{L^2((a,b)\times(0,T))}}. \tag{3.3.17}$$

Note that $y_{\Delta x}^{(j)} = (y_{\Delta x,1}^{(j)}, y_{\Delta x,2}^{(j)})$ stands for the approximate solution on the whole domain $(a,b)\times(0,T)$ at iteration j obtained by concatenating the solutions on domain 1 and 2. For the computation, we have chosen empirically $j = 5$.

We first remark that numerical and theoretical convergence rates coincide for small frequencies but not for high frequencies. Reproducing the experiment for finer meshes on Figure 3.6 leads to the same observation. We point out that the coincidence of theoretical and numerical rates holds for about 10 frequencies for $N = 51$, 20 frequencies for $N = 101$ and 40 frequencies for $N = 201$. Therefore, for each k the numerical convergence rate tends punctually toward the theoretical one as Δx goes to 0, the convergence being not uniform with respect to k. Besides, we also remark that the numerical convergence rate is almost symmetric with respect the middle frequency

$\frac{k_{\min}+k_{\max}}{2}$, although we are not able to explain this phenomenon (a precise theoretical study of the discrete convergence rate is then required).



(a) $N = 101$ (b) $N = 201$

Figure 3.6: Numerical and theoretical convergence factor

On figure 3.7, in the case $N = 50$, we plot the error with respect of the number of iterations for three different values of k. For $k = 1$, we obtain a straight line while for the two other frequencies, there is an inflection point after about 8 iterations: we see successively two different convergence rates. We point out that after the inflection point, the three curves look parallel, meaning that, after a few iterations, the convergence rate for the three frequencies is approximately the same. The situation is similar when refining the mesh. We also notice that the convergence curve starting with a random initial guess seems to be similar to the case $k = 1$.
To visualize the phenomenon, we plot the approximate solutions $y_{\Delta x}^{(j)}$ and $\lambda_{\Delta x}^{(j)}$ for $k \in \{1, 20, 40\}$ at different iterations ($j \in \{2, 5, 7, 20\}$). For $k = 1$ (Fig. 3.8), the shape of the solution remains globally the same at each iteration. This is not the case for $k = 20$ and $k = 37$ (Fig. 3.9,3.10) wherein the solution, initialized with a high frequency initial guess seems to become smoother after a few iterations.

We also investigate the convergence of the algorithm with respect to T_1 in Figure 3.11. In that experiment, we compute the numerical convergence factor by using (3.3.17) for $j = 5$ and taking the maximum over all the frequencies k. As in the continuous case, the numerical convergence factor seems to be an increasing function of T and there is a number T^*_{num} such that the algorithm converges if $T < T^*_{num}$ while diverges for $T \geq T^*_{num}$. We remark that T^*_{num} almost coincides with the theoretical parameter T^*, although it is always smaller than the theoretical one.

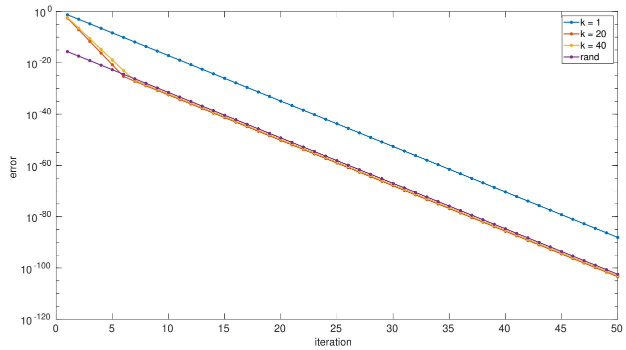
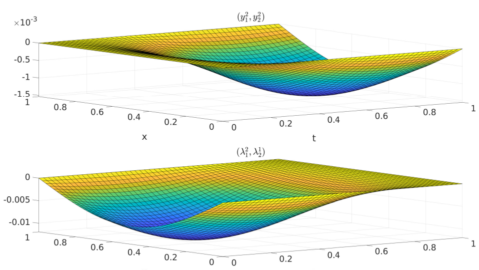
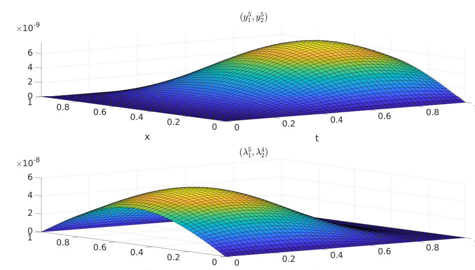


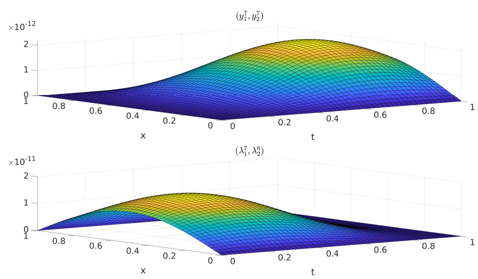
Figure 3.7: Error with respect to the number of iterations for different values of k



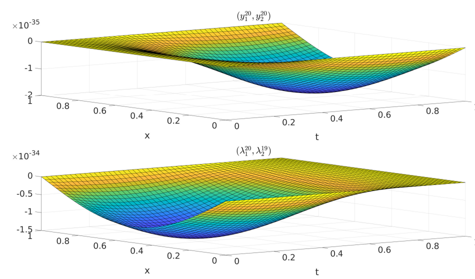
(a) $j = 2$



(b) $j = 5$

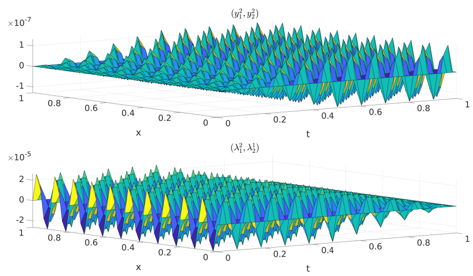


(c) $j = 7$

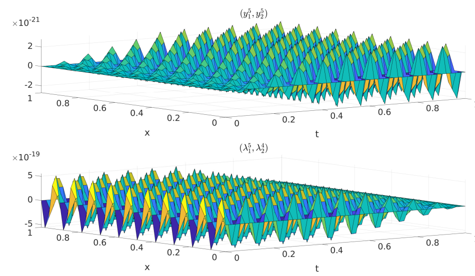


(d) $j = 20$

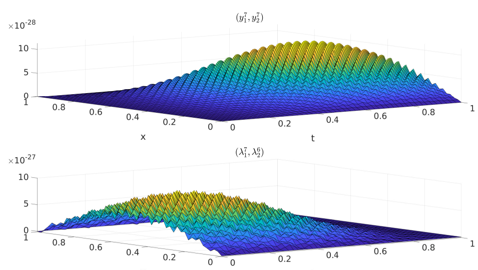
Figure 3.8: Solution $y_{\Delta x}^{(j)}$ and $\lambda_{\Delta x}^{(j)}$ for $k = 1$



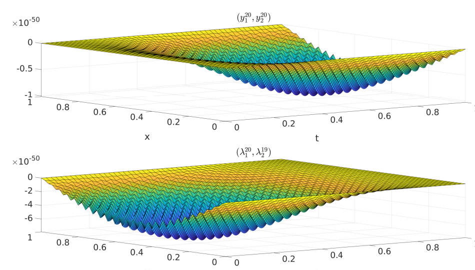
(a) $j = 2$



(b) $j = 5$

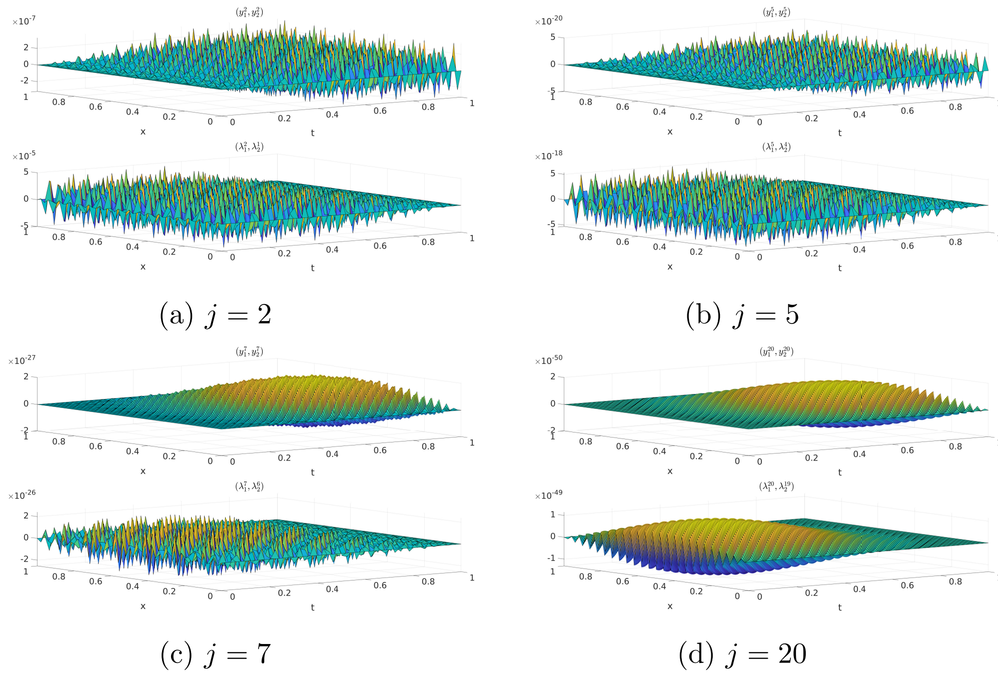


(c) $j = 7$



(d) $j = 20$

Figure 3.9: Solution $y_{\Delta x}^{(j)}$ and $\lambda_{\Delta x}^{(j)}$ for $k = 20$



(a) $j = 2$

(b) $j = 5$

(c) $j = 7$

(d) $j = 20$

Figure 3.10: Solution $y_{\Delta x}^{(j)}$ and $\lambda_{\Delta x}^{(j)}$ for $k = 37$

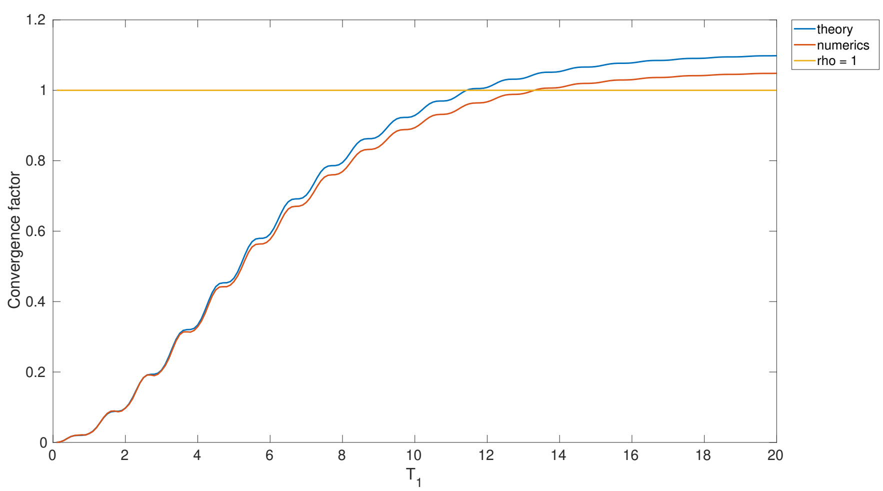


Figure 3.11: Convergence factor with respect to T_1

Finally, we consider the case of several subdomains, using the parallel inherited algorithm: to be more specific, we fixed T, Δt and Δx. It means that when using more subdomains, the subproblems to solve have a smaller size. We plot the error with respect to the number of iterations starting from a random initial guess. We see that the convergence rate deteriorates as the the number of subdomains increases, meaning that our algorithm is not scalable. This is not surprising since this is a classical feature of domain decomposition methods that can be overcome by the addition of a coarse grid solver (see [38, 27, 25, 24, 12]).

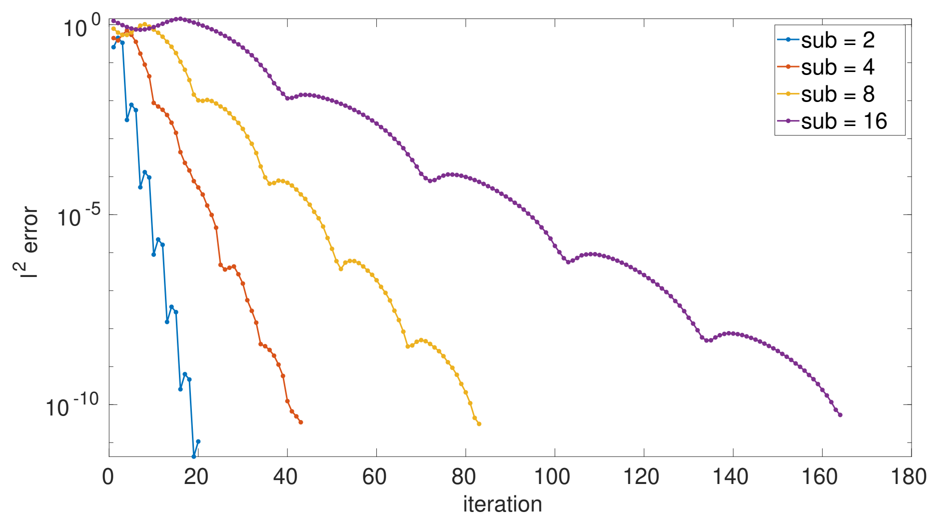


Figure 3.12: Convergence with respect to the number of iterations for 2, 4, 8 and 16 subdomains in time

Chapter 4

Variants of algorithm

4.1 Description

In the chapter, we continue to work on the 'toy problem', that is in one dimension, with $\gamma = 0$, no overlap $\delta = 0$, and $\alpha = 1$, $T_1 = T/2$. We have defined the inherited algorithm that converges for $T \leq T_0$ but diverges for T large.

The aims of this chapter is to study two different algorithms: the first one is the inherited algorithm with relaxation, while the second one, which contains relaxation inherently, is a Dirichlet-Neumann type algorithm.

4.2 Relaxed Inherited algorithm

Let us first describe the relaxed Inherited algorithm. It iterates on $\underline{\lambda}^n, \underline{\lambda}^{\prime,n}$ as follows:

- Step 1, domain 1: Final transmission conditions in $(0, T_1)$:

$$\lambda_1^{n+1}(T_1) = \underline{\lambda}^n, \quad \partial_t \lambda_1^{n+1}(T_1) = \underline{\lambda}'^n,$$

- Step 2, domain 2: Initial transmission conditions in (T_1, T):

$$y_2^{n+1}(T_1) = y_1^{n+1}(T_1), \quad \partial_t y_2^{n+1}(T_1) = \partial_t y_1^{n+1}(T_1),$$

- Step 3, relaxation step:

$$\begin{pmatrix} \underline{\lambda}^{n+1} \\ \underline{\lambda}'^{n+1} \end{pmatrix} = \theta \begin{pmatrix} \lambda_2^{n+1} \\ \partial_t \lambda_2^{n+1} \end{pmatrix} + (1-\theta) \begin{pmatrix} \underline{\lambda}^n \\ \underline{\lambda}'^n \end{pmatrix}$$

The iteration matrix is

$$M_\theta = (1-\theta) I + \theta M, \tag{4.2.1}$$

where M is defined in (3.2.51) in the general case, and is given by (3.3.5) for the toy model. The eigenvalue of M_θ are given by

$$\lambda_\pm(\theta) = (1-\theta) + \theta \lambda_\pm,$$

where $\lambda^\pm$ (depending on T, a, b and k) are real negative and given by (3.3.9).

We first prove that we are always able to find a parameter θ such that the relaxed inherited algorithm converges:

Theorem 4.2.1. *Let $T > 0$, $(a, b) \in \mathbf{R}^2$ such that $b > a$ and let*

$$\theta^* = \frac{2}{(1 + \max\limits_{k \in [k_{\min}, k_{\max}]} |\lambda_-|)}.$$

For any $\theta \in (0, \theta^*)$, the relaxed inherited algorithm converges. In particular, if the inherited algorithm converges without relaxation, the relaxed inherited converges for any $\theta \in (0, 1]$.

Proof. The algorithm converges if and only if, for any $k \in [k_{\min}, k_{\max}]$,

$$-1 < \lambda_\pm(\theta) < 1,$$

that is to say

$$-1 < 1 - \theta(1 - \lambda_\pm) < 1 \Leftrightarrow -2 < -\theta(1 - \lambda_\pm) < 0.$$

Because $\lambda_\pm$ are always negative (see (3.3.9)), $(1 - \lambda^\pm) > 0$, and the second inequality is always fulfilled for $\theta > 0$. Then, the first equality becomes

$$\theta < \frac{2}{(1 - \lambda_\pm)} \quad \forall k \in [k_{\min}, k_{\max}]. \tag{4.2.2}$$

Reminding that $\lambda_-(k) < \lambda_+(k) < 0$ for any $k \in [k_{\min}, k_{\max}]$, we have

$$1 < 1 - \lambda_+ < 1 - \lambda_-.$$

Therefore, the condition (4.2.2) is equivalent to

$$\theta < \frac{2}{(1 - \lambda_-)} \quad \forall k \in [k_{\min}, k_{\max}].$$

Noticing that

$$\min_{k \in [k_{\min}, k_{\max}]} \frac{2}{(1 - \lambda_-)} = \frac{2}{(1 + \max\limits_{k \in [k_{\min}, k_{\max}]} |\lambda_-|)} = \theta^*,$$

then, for any $\theta \in (0, \theta^*)$, the algorithm converges.

If the inherited algorithm (without relaxation) converges, then

$$\max_{k \in [k_{\min}, k_{\max}]} |\lambda_-| < 1$$

so that $\theta^* > 1$. Consequently, the relaxed inherited algorithm converges for any $\theta \in (0, 1)$. □

Remark 4.

- *Because it seems numerically that* $\max_{k\in[k_{\min},k_{\max}]}|\lambda^-| = |\lambda^-(k_{\min})|$ *(see Section 3.3.2), presumably,*

$$\theta^* = \frac{2}{(1+|\lambda_-(k_{\min})|)}.$$

- *For* $\theta = 0$*, and* $\theta = \theta^*$*, the associated convergence factor is equal to* 1*.*

Let us define

$$\rho_{\text{relax}}(\theta) = \max_{k\in[k_{\min},k_{\max}]} \max_{\pm} |\lambda_\pm(\theta)|.$$

The next natural question is to know if we can choose θ that minimizes the convergence factor ρ_{relax}, namely if we can solve

$$\min_{\theta\in[0,\theta^*]} \rho_{\text{relax}}(\theta). \tag{4.2.3}$$

Theorem 4.2.2. *In the case* $k_{\max} < +\infty$*, this problem has a solution.*

Proof. the function ρ_{relax} is continuous. Indeed, the function $\lambda_\pm(\theta)$ are continuous functions on $[k_{\min}, k_{\max}] \times [0,\theta^*]$. Therefore, $\max_{k\in[k_{\min},k_{\max}]}|\lambda_\pm(\theta)|$ are continuous with respect to θ, and consequently ρ_{relax}. Because $[0,\theta^*]$ is compact, the function ρ_{relax} reaches a minimum. □

It seems not trivial to obtain a closed formula for the optimal θ. However, the case $k_{max} = +\infty$ appears easy to solve.

Theorem 4.2.3. *Assume that* $k_{max} = +\infty$ *and that the function* $k \mapsto \lambda_-(k)$ *is increasing. The minimum of* ρ_{relax} *is reached in*

$$\theta_{opt} = \frac{2}{2-\lambda_-(k_{min})} < \min(\theta^*, 1). \tag{4.2.4}$$

The corresponding convergence factor is equal to $1-\theta_{opt}$*.*

Proof. Let

$$\theta_1 = \frac{1}{1-\lambda_-(k_{\min})}$$

We remark that $\theta_1 < \theta_{\text{opt}} < \min(\theta^*, 1)$. Indeed,

$$\theta_1 < \theta^* \quad \Leftrightarrow \quad \lambda_-(k_{\min}) < 0 \quad \text{and} \quad 2-\lambda_-(k_{\min}) > 1-\lambda_-(k_{\min}).$$

Let us consider separately the intervals $[0,\theta_1]$ and $[\theta_1,\theta^*]$.

- For $\theta \in [0, \theta_1]$, the quantities

$$(1-\theta) + \theta\lambda_{\pm}$$

are both positive for any $k \in [k_{\min}, +\infty)$. Indeed,

$$(1-\theta) + \theta\lambda_{+}(k) \geq (1-\theta) + \theta\lambda_{-}(k) \geq (1-\theta) + \theta\lambda_{-}(k_{\min}) > 0.$$

As a result,

$$\rho_{\text{relax}}(\theta) = (1-\theta) + \theta\lambda_{+}(k).$$

Besides, reminding that $\lambda_{+}(k) < 0$ and $\lim_{k\to+\infty} \lambda_{+}(k) = 0$

$$(1-\theta) + \theta\lambda_{+}(k) \leq (1-\theta) \quad \text{and} \quad \lim_{k\to+\infty} (1-\theta) + \theta\lambda_{+}(k) = (1-\theta).$$

Therefore

$$\rho_{\text{relax}}(\theta) = 1-\theta.$$

- For $\theta \in [\theta_1, \theta^*]$, we still have

$$(1-\theta) + \theta\lambda_{+}(k) \leq (1-\theta), \quad \lim_{k\to+\infty} (1-\theta) + \theta\lambda_{\pm}(k) = (1-\theta).$$

and $(1-\theta) + \theta\lambda_{-}(k) \leq (1-\theta) + \theta\lambda_{+}(k)$. The only difference is that $(1-\theta) + \theta\lambda_{-}(k)$ take negative values (for small k). As a result, using again the monotony of $\lambda_{-}(k)$, (see Fig. 4.1)

$$\rho_{\text{relax}}(\theta) = \max\{(1-\theta), -((1-\theta) + \theta\lambda_{-}(k_{\min}))\}.$$

Note that

$$(1-\theta) = (\theta-1) - \theta\lambda_{-}(k_{\min})$$

if and only if $\theta = \theta_{\text{opt}}$. Therefore, for $\theta \in [\theta_1, \theta_{\text{opt}}]$, the maximum is $1-\theta$ while for $\theta \in [\theta_{\text{opt}}, \theta^*]$, the maximum is $(\theta-1) - \theta\lambda_{-}(k_{\min})$.

To summarize,

$$\rho_{\text{relax}}(\theta) = \begin{cases} 1-\theta & \text{if } \theta < \theta_{\text{opt}} \\ (\theta-1) - \theta\lambda_{-}(k_{\min}) & \text{if } \theta \in [\theta_{\text{opt}}, \theta^*]. \end{cases}$$

Since $\theta \mapsto 1-\theta$ is decreasing and $\theta \mapsto \theta(1-\lambda_{-}(k_{\min})) - 1$ is increasing, the minimum of ρ_{relax} is reached in θ_{opt}. □

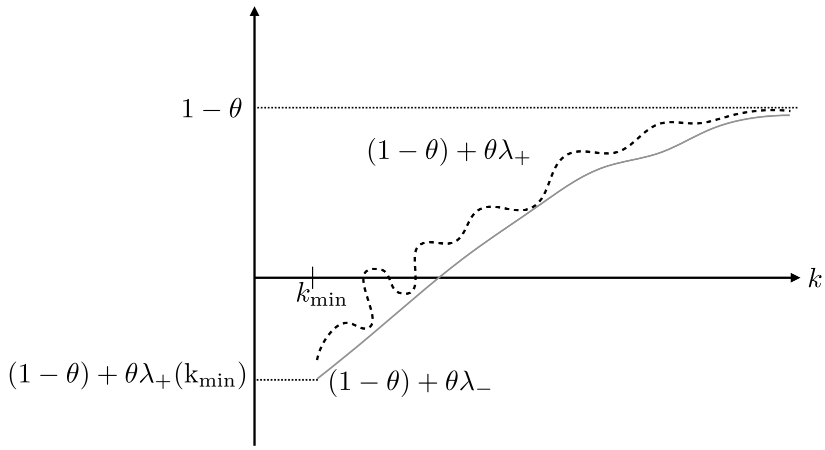
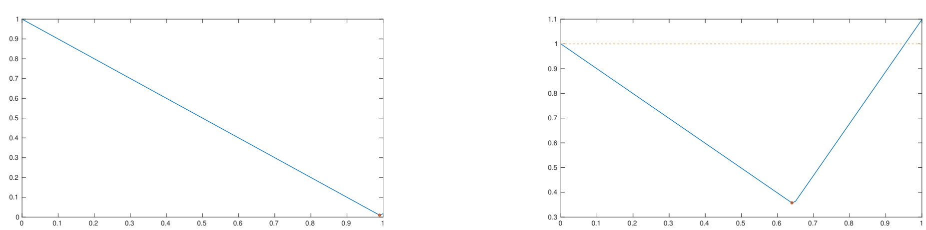


Figure 4.1: Schematic representation of $(1-\theta)+\theta\lambda_\pm(k)$ for $\theta \in [\theta_1, \theta^*]$.

The previous theorems are illustrated on Figure 4.2 for two different values of T ($T = 1$ and $T = 10$). We have chosen $a = 0$, $b = 1$ and $k_{\max} = 100$. The red cross corresponds to the value θ^*, namely the largest value of k that makes the relaxed algorithm converging. The circle corresponds to the optimal parameter θ_{opt} computed using the formula (4.2.4). We notice that it provides a good estimation even for $k_{\max}$ is finite. This is probably due to the fact that $\lambda_\pm$ goes very fast to 0 as $k_{\max}$ goes to infinity.



(a) $T = 1$ (b) $T = 40$

Figure 4.2: Convergence factor with respect to θ

Remark 5. *If the case $k_{\max}$ large were similar to the case $k_{\max} = +\infty$ (this is not proved), then the previous lemma is an encouraging result if we* *think of solving the corresponding interface problem using GMRES (seeing the inherited algorithm as a preconditioner). Indeed, the GMRES algorithm is usually better than the relaxation one.*

We made some numerical tests on the discretized problem using the finite volume method described in Chapter 2. In Figure 4.3, we evaluate the theoretical and numerical convergence factor as a function of T and θ. We remark that theoretical and numerical convergence factor coincide. Then, we plot on Figure 4.4, the error with respect to the number of iterations for $T = 1$ and $T = 40$ for two values of θ: $\theta = 1$ (no relaxation) and $\theta = \theta_{opt}$ (computed using formula 4.2.4). In the first case, the inherited algorithm already converges but the convergence rate is improved using a relaxed parameter. In the second case $T = 40$, the inherited algorithm diverges without relaxation but adding the relaxation parameter permits to recover the convergence.

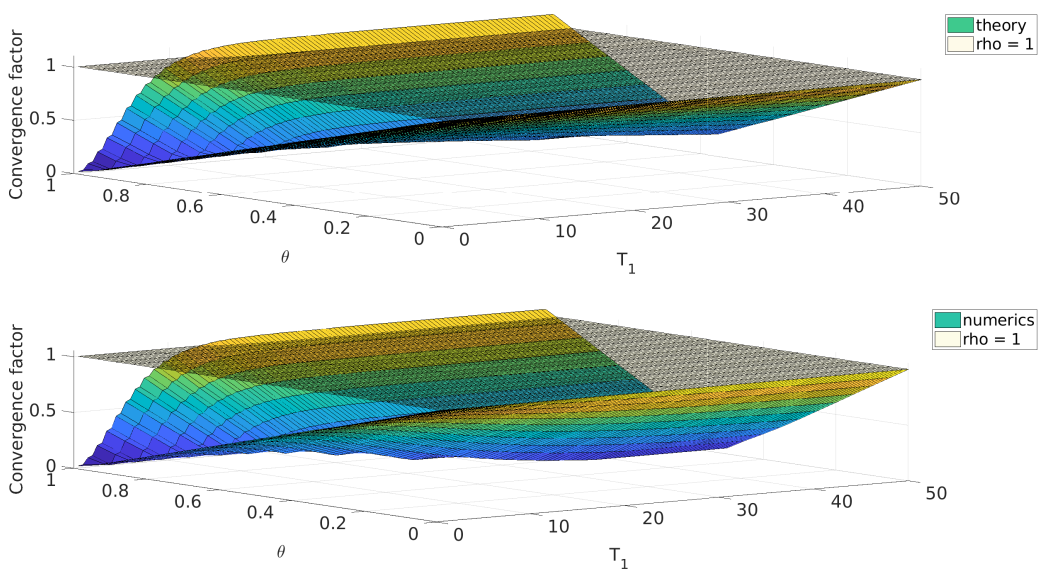
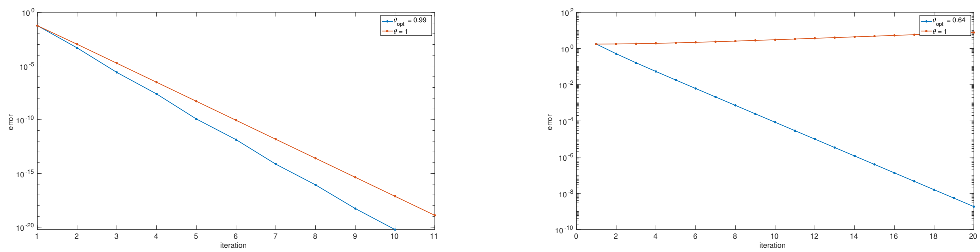


Figure 4.3: theoretical and numerical convergence factors with respect to T_1 and θ



(a) $T = 1$ (b) $T = 40$

Figure 4.4: Error with respect to the number of iterations with and without relaxation

4.3 The Dirichlet-Neumann algorithm

4.3.1 The Poisson equation

The Dirichlet-Neumann algorithm was invented by Bjorstad and Widlund in 1986 [9] for elliptic problems. We first recall the definition and properties for the Poisson problem on a rectangle divided into two sub rectangles, without overlap. The domain in x is $(0, L)$ and in y it is $(-a_1, a_2)$. The goal is to solve $-\Delta u = f$ in Ω, with Dirichlet boundary condition $u = 0$ all around.

Ω is divided into $\Omega_1 = (0, L) \times (-a_1, 0)$ and $\Omega_2 = (0, L) \times (0, a_2)$. One step of the algorithm is defined as follows. Given $g^n(x)$ define the iterate n in Ω_1 by

$$\begin{aligned} &-\Delta u_1^n = f \text{ in } \Omega_1, \\ &u_1^n = 0 \text{ in } \partial\Omega_1 \cap \partial\Omega, \\ &u_1^n = g^n \text{ in } \partial\Omega_1 \setminus \partial\Omega. \end{aligned} \tag{4.3.1}$$

$$\begin{aligned} &-\Delta u_2^n = f \text{ in } \Omega_2, \\ &u_2^n = 0 \text{ in } \partial\Omega_2 \cap \partial\Omega, \\ &\partial_y u_2^n = \partial_y u_1^n \text{ in } \partial\Omega_2 \setminus \partial\Omega. \end{aligned} \tag{4.3.2}$$

$$g^{n+1} = \theta u_2^n + (1 - \theta) g^n, \tag{4.3.3}$$

with $\theta \in (0, 1)$. The error is solution of the same algorithm with zero data. It is computed by series in x as in previous chapter. Write

$$g^n = \sum_{k \geq 1} \hat{g}^n(k) \sin \xi_k x,$$

with $\xi_k = k\pi/L$. Then

$$u_j^n = \sum \hat{u}_j^n(\xi_k, y) \sin \xi_k x,$$

with the transmission conditions

$$\hat{u}_1^n(\xi_k, 0) = \hat{g}^n(k), \quad \partial_y \hat{u}_2^n(\xi_k, 0) = \partial_y \hat{u}_1^n(\xi_k, 0).$$

We separate 3 cases

Case 1 $a_1 = a_2 = +\infty$. Then

$$\hat{u}_1^n(\xi_k, y) = \alpha_1^n(k) e^{\xi_k y}, \quad \hat{u}_2^n = \alpha_2^n(k) e^{-\xi_k y}.$$

Use the transmission conditions

$$\alpha_1^n = \hat{g}^n, \quad \alpha_2^n = -\alpha_1^n.$$

Therefore

$$\hat{g}^{n+1} = (1 - 2\theta)\hat{g}^n.$$

The convergence factor is $1 - 2\theta$. For $\theta = \frac{1}{2}$ the algorithm converges in one iteration. For $\theta \in (0, 1)$,

$$\|\hat{g}^{n+1}\| = |1 - 2\theta| \|\hat{g}^n\|,$$

and by Parseval identity, the sequence g^n converges geometrically.

Case 2 $a_1 = a_2 := a < +\infty$

Then

$$\hat{u}_1^n(\xi_k, y) = \hat{g}^n(k) \frac{\sinh(\xi_k(y + a))}{\sinh(\xi_k a)}, \quad \hat{u}_2^n = \alpha_2^n(k) \sinh(\xi_k(y - a)).$$

Use the transmission conditions for $\hat{u}_2^n$

$$\alpha_2^n \cosh(\xi_k a) = \hat{g}^n(k) \frac{\cosh(\xi_k a)}{\sinh(\xi_k a)}$$

Therefore $\alpha_2^n = \frac{\hat{g}^n}{\sinh(\xi_k a)}$, and

$$\hat{g}^{n+1} = (1 - 2\theta)\hat{g}^n.$$

Case 3 $a_1 \neq a_2 = +\infty$ The same analysis gives

$$\hat{g}^{n+1} = \hat{g}^n \left[1 - \theta(1 + \frac{\tanh(\xi_k a_2)}{\tanh(\xi_k a_1)})\right].$$

Then the convergence factor is $\rho(\xi) = 1 - \theta(1 + \frac{\tanh(\xi a_2)}{\tanh(\xi a_1)})$. The algorithm converges if for all k, its absolute value is smaller than 1. It diverges if there exists a k for which it is larger than 1 (see [43])

$$\begin{aligned}
\partial_\xi \rho &= -\theta \left[\frac{a_2}{\cosh^2(\xi a_2)\tanh(\xi a_1)} - \frac{a_1 \tanh(\xi a_2)}{\sinh^2(\xi a_1)}\right] \\
&= -\theta \left[\frac{a_2 \cosh(\xi a_1)}{\cosh^2(\xi a_2)\sinh(\xi a_1)} - \frac{a_1 \sinh(\xi a_2)}{\cosh(\xi a_2)\sinh^2(\xi a_1)}\right] \\
&= -\frac{\theta(a_2 \cosh(\xi a_1)\sinh(\xi a_1) - a_1 \sinh(\xi a_2)\cosh(\xi a_2))}{\cosh^2(\xi a_2)\sinh^2(\xi a_1)} \\
&= -\frac{\theta(a_2 \sinh(2\xi a_1) - a_1 \sinh(2\xi a_2))}{2\cosh^2(\xi a_2)\sinh^2(\xi a_1)} \\
&= -\frac{a_2 a_1 \xi\theta}{2\cosh^2(\xi a_2)\sinh^2(\xi a_1)}(\frac{\sinh(2\xi a_1)}{2\xi a_1} - \frac{\sinh(2\xi a_2)}{2\xi a_2}) \\
&= \frac{a_2 a_1 \xi\theta}{2\cosh^2(\xi a_2)\sinh^2(\xi a_1)}(\frac{\sinh(2\xi a_2)}{2\xi a_2} - \frac{\sinh(2\xi a_1)}{2\xi a_1})
\end{aligned}$$

The function $x \to \sinh(x)/x$ is increasing, furthermore

$$\lim_{\xi \to 0} \rho(\xi) = 1 - \theta(1 + \frac{a_2}{a_1})$$

$$\lim_{\xi \to +\infty} \rho(\xi) = 1 - 2\theta \in (-1, 1),$$

- If $a_1 > a_2$, ρ is decreasing in ξ. Furthermore $\rho(0) \in (-1, 1)$, we have for all values of θ, for all values of ξ, $|\rho(\xi)| < 1$, and the algorithm converges. Can we find an optimal θ. There are three cases, depending on θ:
  - $\rho(0) > \rho(+\infty) > 0$, which is equivalent to $1 - 2\theta > 0 \iff \theta < \frac{1}{2}$. Then
  $$\max_\xi |\rho(\xi)| = \rho(0) = 1 - \theta(1 + \frac{a_2}{a_1}).$$
  - $> \rho(0) > 0 > \rho(+\infty)$, or equivalently $\frac{1}{2} < \theta < 1/(1 + \frac{a_2}{a_1})$, then
  $$\max_\xi |\rho(\xi)| = \max(\rho(0), -\rho(+\infty)) = \max(2\theta - 1, 1 - \theta(1 + \frac{a_2}{a_1})).$$
  - $0 > \rho(0) > \rho(+\infty)$, or equivalently $\frac{1}{1+\frac{a_2}{a_1}} < \theta < 1$, then
  $$\max_k |\rho(k)| = -\rho(+\infty) = 2\theta - 1$$

Therefore the optimal θ is when $2\theta - 1 = 1 - \theta(1 + \frac{a_2}{a_1})$ that is

$$\theta_{opt} = \frac{2}{3 + \frac{a_2}{a_1}}.$$

- If $a_1 < a_2$, ρ is increasing in ξ. There are three cases, depending on θ:
  - $0 < \rho(0) < \rho(+\infty)$, which is equivalent to $0 < 1 - \theta(1 + \frac{a_2}{a_1}) < 1 - 2\theta$ or $\theta < 1/(1 + \frac{a_2}{a_1})$. Then

    $$\max_{\xi} |\rho(\xi)| = \rho(+\infty) = 1 - 2\theta.$$

  - $\rho(0) < 0 < \rho(+\infty)$, or equivalently $1/(1 + \frac{a_2}{a_1}) < \theta < \frac{1}{2}$, then

    $$\max_{\xi} |\rho(\xi)| = \max(-\rho(0), \rho(+\infty)) = \max(1 - 2\theta, -1 + \theta(1 + \frac{a_2}{a_1})).$$

  - $0 < \rho(0) < \rho(+\infty) < 0$, or equivalently $\theta > \frac{1}{2}$, then

    $$\max_{\xi} |\rho(\xi)| = -\rho(0) = -1 + \theta(1 + \frac{a_2}{a_1})$$

We have two conclusions

1. The algorithm diverges for $\theta > \theta_{lim} = \frac{2}{1 + \frac{a_2}{a_1}}$,
2. The maximum of $|\rho|$ is minimal for

$$\theta_{opt} = \frac{2}{3 + \frac{a_2}{a_1}}.$$

For all values of $\theta \in (0, \theta_{opt})$, the algorithm converges.

4.3.2 Dirichlet Neumann algorithm for the control problem

This question of symmetry does not arise in the previous algorithm. In order to take this phenomenon into account we change a little bit the definition of the intervals in time. The time interval for the control problem is

$(-T_1, T_2)$, and will be divided into $(-T_1, 0)$ and $(0, T_2)$. Motivated by the paper by Gander and Yongxiang Liu [48], we create a new algorithm which is called Dirichlet Neumann algorithm. We solve the optimality system 2.1.8 by changing the transmission conditions in (3.1.1,3.1.2). Let G^0 a function of x, define recursively an algorithm on G^n by

In subdomain $O_1 = (-T_1, 0)$,

State equation

$$\partial_{tt} y_1^{n+1} - \Delta y_1^{n+1} - \frac{1}{\alpha}\lambda_1^{n+1} = 0, \qquad \text{in } \Omega \times O_1, \tag{4.3.4a}$$

with boundary condition

$$y_1^{n+1} = 0 \qquad \text{in } \partial\Omega \times O_1, \tag{4.3.4b}$$

with initial data

$$\begin{aligned} y_1^{n+1}(\cdot, -T_1) &= y^{(0)}, & &, \\ \partial_t y_1^{n+1}(\cdot, -T_1) &= y^{(1)}, & &\text{in } \Omega, \end{aligned} \tag{4.3.4c}$$

Adjoint equation

$$\partial_{tt} \lambda_1^{n+1} - \Delta \lambda_1^{n+1} + y_1^{n+1} = \widehat{y}, \qquad \text{in } \Omega \times O_1, \tag{4.3.4d}$$

with boundary condition

$$\lambda_1^{n+1} = 0 \qquad \text{in } \partial\Omega \times O_1, \tag{4.3.4e}$$

with the transmission final conditions

$$(y_1^{n+1}, \lambda_1^{n+1})(\cdot, 0) = G^n(\cdot, 0). \tag{4.3.4f}$$

In subdomain $O_2 = (0, T_2)$,

State equation

$$\partial_{tt} y_2^{n+1} - \Delta y_2^{n+1} - \frac{1}{\alpha}\lambda_2^{n+1} = 0, \qquad \text{in } \Omega \times O_2, \tag{4.3.5a}$$

with boundary condition

$$y_2^{n+1} = 0 \qquad \text{in } \partial\Omega \times O_2, \tag{4.3.5b}$$

with initial transmission conditions

$$\begin{aligned} &\partial_t y_2^{n+1}(\cdot, 0) = \partial_t y_1^{n+1}(\cdot, 0), && \text{in } \Omega, \\ &\partial_t \lambda_2^{n+1}(\cdot, 0) = \partial_t \lambda_1^{n+1}(\cdot, 0), && \text{in } \Omega. \end{aligned} \tag{4.3.5c}$$

ADJOINT EQUATION

$$\partial_{tt}\lambda_2^{n+1} - \Delta\lambda_2^{n+1} + y_2^{n+1} = \widehat{y}, \qquad \text{in } \Omega \times O_2, \tag{4.3.5d}$$

with boundary condition

$$\lambda_2^{n+1} = 0 \qquad \text{in } \partial\Omega \times O_2. \tag{4.3.5e}$$

with the final conditions

$$\begin{aligned} &\lambda_2^{n+1}(\cdot, T_2) = 0, \\ &\partial_t \lambda_2^{n+1}(\cdot, T) = \gamma(y_2^{n+1}(\cdot, T_2) - \widehat{z}), && \text{in } \Omega \end{aligned} \tag{4.3.5f}$$

Relaxation

$$G^{n+1} = \theta(y_2^{n+1}, \lambda_2^{n+1})(\cdot, 0) + (1-\theta)G^n. \tag{4.3.5g}$$

The lines in red, (4.3.4f, 4.3.5c,4.3.5g) are the new transmission conditions.

In the Schwarz algorithm, the well-posedness of the subproblems was given by the fact that they were the optimality systems of convex functions. It does not seem to be the case here, so we must manage differently. We use the sine decomposition defined in chapter 3, for the well-posedness and for the convergence too. We start with the infinite domain (in time) case which it is a good warming. For simplicity we assume that $\hat{y} = 0$.

4.3.3 Infinite domains in time

We use the notations of chapter 3.Start with the problem in subdomain 1, whose solution we call $Y_1 = (y, \lambda)$. By the equation we write it as

$$Y_1(x,t) = \sum_{k\geq 1} \hat{Y}_1(k,t)\Phi_k(x),$$

and use the boundary conditions

$$Y_1(\cdot, -\infty) = 0, \quad Y_1(\cdot, 0) = G.$$

They are very easy to implement into the series as follows.

$$Y_1 = \cos(Nt)\hat{G} + \sin(Nt)\tilde{G}. \tag{4.3.6}$$

$\tilde{G}$ has now to be determined such as to enforce the condition at infinity. To that end, use formulas (3.2.9), and determine the behaviour at $-\infty$ of the sine and cosine terms, using formulas (3.2.9,3.2.7), and the behavior at infinity of the real hyperbolic sine and cosine functions.

$$\cosh \nu_2 t \sim \frac{1}{2}e^{-\nu_2 t} := X, \quad \sinh \nu_2 t \sim -\frac{1}{2}e^{-\nu_2 t}.$$

X is the large parameter. For $\cos Nt$ use

$$\operatorname{Re}\cos \nu t \sim X \cos \nu_1 t, \quad \operatorname{Im}\cos \nu t \sim X \sin \nu_1 t,$$

$$\implies \cos(Nt) \sim X(\cos \nu_1 t I + \sin \nu_1 t J).$$

For $\sin Nt$ write

$$\operatorname{Re}\sin \nu t \sim X \sin \nu_1 t, \quad \operatorname{Im}\sin \nu t \sim -X \cos \nu_1 t,$$

$$\implies \sin(Nt) \sim X(\sin \nu_1 t I - \cos \nu_1 t J).$$

Insert into (4.3.6),

$$\hat{Y}_1 \sim X\left[(\cos \nu_1 t I + \sin \nu_1 t J)\hat{G} + (\sin \nu_1 t I - \cos \nu_1 t J)\tilde{G}\right].$$

Reorder in cosine and sine

$$\hat{Y}_1 \sim X\left[\cos \nu_1 t(\hat{G} - J\tilde{G}) + \sin \nu_1 t(J\hat{G} + \tilde{G})\right].$$

Use that $J^2 = -I$,

$$\hat{Y}_1 \sim X\left[\cos\nu_1 t(\hat{G} - J\tilde{G}) + \sin\nu_1 tJ(\hat{G} - J\tilde{G})\right] \sim X(\cos\nu_1 tI + \sin\nu_1 tJ)(\hat{G} - J\tilde{G}).$$

The matrix $\cos\nu_1 tI + \sin\nu_1 tJ$ is a rotation matrix, has no limit at infinity. If Y has to tend to 0 at $-\infty$, we must have $\hat{G} - J\tilde{G} = 0$. Therefore $\tilde{G} = -J\hat{G}$ and since J and N commute, we have

$$\hat{Y}_1 = (\cos(Nt) - J\sin(Nt))\hat{G}. \tag{4.3.7}$$

For each k, the problem has therefore a unique solution, and

$$\|Y_1(k,t)\| = \|\hat{G}(k)\|.$$

By Parseval, this shows that the problem is well-posed. The problem in Ω_2 has the the boundary condition.

$$\partial_t Y_2(\cdot, 0) = H.$$

Write

$$Y_2(x,t) = \sum_{k\geq 1} \hat{Y}_2(k,t)\Phi_k(x),$$

and use the boundary condition to write

$$\hat{Y}_2 = \cos(Nt)\tilde{H} + N^{-1}\sin(Nt)\hat{H}. \tag{4.3.8}$$

Enforce now the condition at infinity $Y(\cdot, +\infty) = 0$.

$$\cosh\nu_2 t \sim \frac{1}{2}e^{\nu_2 t} := X, \quad \sinh\nu_2 t \sim \frac{1}{2}e^{\nu_2 t}.$$

X is the large parameter. First by (3.2.7),

$$\cos(Nt) \sim X(\cos\nu_1 tI - \sin\nu_1 tJ),\ \sin(Nt) \sim X(\sin\nu_1 tI + \cos\nu_1 tJ).$$

Insert into (4.3.8),

$$\begin{aligned}
\hat{Y}_2 &\sim X\left[(\cos\nu_1 tI - \sin\nu_1 tJ)\tilde{H} + (\sin\nu_1 tI + \cos\nu_1 tJ)N^{-1}\hat{H}\right].\\
&\sim X\left[(\cos\nu_1 tI - \sin\nu_1 tJ)\tilde{H} + (-\sin\nu_1 tJ + \cos\nu_1 tI)JN^{-1}\hat{H}\right]\\
&\sim X(\cos\nu_1 tI - \sin\nu_1 tJ)\left[\tilde{H} + JN^{-1}\hat{H}\right]
\end{aligned}$$

.

Consequently

$$\tilde{H} + JN^{-1}\hat{H} = 0,$$

and

$$\cos(Nt)\tilde{H} + N^{-1}\sin(Nt)\hat{H} = -\cos(Nt)JN^{-1} + N^{-1}\sin(Nt)\hat{H},$$

$$\hat{Y}_2(k,t) = N^{-1}(\sin(Nt)I - \cos(Nt)J)\hat{H}. \tag{4.3.9}$$

This problem is thus well-posed (see the analysis for domain 1).

Write now the recursion

$$\begin{aligned}\hat{Y}_1^{n+1}(k,t) &= (\cos(Nt) - J\sin(Nt))\hat{G}^n(k),\\ \hat{Y}_2^{n+1}(k,t) &= N^{-1}(\sin(Nt)I - \cos(Nt)J)\partial_t Y_1^{n+1}(k,0).\end{aligned}$$

Compute $\partial_t \hat{Y}_1^{n+1}(k,0) = -JN\hat{G}^n(k)$, which gives

$$\hat{Y}_2^{n+1}(k,t) = -N^{-1}(\sin(Nt)I - \cos(Nt)J)JN\hat{G}^n(k),$$

hence, since all the matrices involved commute,

$$\hat{Y}_2^{n+1}(k,0) = -\hat{G}^n(k).$$

Therefore

$$\hat{G}^{n+1}(k) := \theta\hat{Y}_2^{n+1}(k,0) + (1-\theta)\hat{G}^n(k) = (1-2\theta)\hat{G}^n(k).$$

The convergence matrix is independent of k and equal to $(1-2\theta)I$ as in the elliptic case. Therefore the algorithm converges for any $0 < \theta < 1$. For $\theta = \frac{1}{2}$, it converges in two iterations.

4.3.4 Finite domains in time

Again in order to prove well-posedness, we decomposed in series in space, and compute explicitly for each frequency the solution of of the transform problem.

Then we use the formulas to compute the convergence factor of the algorithm.

Well-posedness of the subproblems

Domain O_1 The boundary conditions in time at $T=0$ are the same as before, so the formula (4.3.6) is still valid, but now $\tilde{G}$ is such that $y(\cdot,-T_1)=y^{(0)}$ and $\partial_t y(\cdot,-T_1)=y^{(1)}$. We proceed as in (**??**).

$$\begin{aligned}\hat{y}(k,-T_1) &= (\cos(NT_1)\hat{G}-\sin(NT_1)\tilde{G})_1=\hat{y}^{(0)},\\ \partial_t\hat{y}(k,-T_1) &= (N\sin(NT_1)\hat{G}+N\cos(NT_1)\tilde{G})_1=\hat{y}^{(1)}.\end{aligned} \tag{4.3.10}$$

Use the formulas (**??**) to develop the four terms on the left. Start with those concerning $\hat{G}$.

$$\cos(NT_1)\hat{G}=\mathrm{Re}(\cos\nu T_1)\hat{G}+\mathrm{Im}(\cos\nu T_1)J\hat{G}.$$

$$N\sin(NT_1)\hat{G}=\mathrm{Re}(\nu\sin\nu T_1)\hat{G}+\mathrm{Im}(\nu\sin\nu T_1)J\hat{G}.$$

Note that $(I\hat{G})_1=\hat{G}_1$, $(J\hat{G})_1=-\hat{G}_2$, and rewrite

$$\begin{aligned}(\cos(NT_1)\hat{G})_1 &= \mathrm{Re}(\cos\nu T_1)\hat{G}_1-\mathrm{Im}(\cos\nu T_1)\hat{G}_2\\ (N\sin(NT_1)\hat{G})_1 &= \mathrm{Re}(\nu\sin\nu T_1)\hat{G}_1-\mathrm{Im}(\nu\sin\nu T_1)\hat{G}_2.\end{aligned} \tag{4.3.11}$$

Define the matrix

$$S_1(\nu,T_1)=\begin{pmatrix}\mathrm{Re}(\cos\nu T_1) & -\mathrm{Im}(\cos\nu T_1)\\ \mathrm{Re}(\nu\sin\nu T_1) & -\mathrm{Im}(\nu\sin\nu T_1)\end{pmatrix} \tag{4.3.12}$$

then

$$\begin{pmatrix}(\cos(NT_1)G)_1\\ (N\sin(NT_1)G)_1\end{pmatrix}=S_1(\nu,T_1)G. \tag{4.3.13}$$

Proceed with the terms containing $\tilde{G}$.

$$\sin(NT_1)\tilde{G}=\mathrm{Re}(\sin\nu T_1)\tilde{G}+\mathrm{Im}(\sin\nu T_1)J\tilde{G}.$$

$$N\cos(NT_1)=(\mathrm{Re}\,\nu I+\mathrm{Im}\,\nu J)(\mathrm{Re}(\cos\nu T_1)I+\mathrm{Im}(\cos\nu T_1)J)=\mathrm{Re}(\nu\cos\nu T_1)I+\mathrm{Im}(\nu\cos\nu T_1)J$$

$$N\cos(NT_1)\tilde{G}=\mathrm{Re}(\nu\cos\nu T_1)\tilde{G}+\mathrm{Im}(\nu\cos\nu T_1)J\tilde{G}$$

Which gives

$$\begin{aligned}-(\sin(NT_1)\tilde{G})_1 &= -\mathrm{Re}(\sin\nu T_1)\tilde{G}_1+\mathrm{Im}(\sin\nu T_1)\tilde{G}_2\\ (N\cos(NT_1)\tilde{G})_1 &= \mathrm{Re}(\nu\cos\nu T_1)\tilde{G}_1-\mathrm{Im}(\nu\cos\nu T_1)\tilde{G}_2.\end{aligned} \tag{4.3.14}$$

or

$$\begin{pmatrix} -(\sin(NT_1)\tilde{G})_1 \\ (N\cos(NT_1)\tilde{G})_1 \end{pmatrix} = \tilde{S}_1(\nu, T_1)\tilde{G}, \tag{4.3.15}$$

with

$$\tilde{S}_1(\nu, T_1) = \begin{pmatrix} -\operatorname{Re}(\sin \nu T_1) & \operatorname{Im}(\sin \nu T_1) \\ \operatorname{Re}(\nu \cos \nu T_1) & -\operatorname{Im}(\nu \cos \nu T_1) \end{pmatrix} \tag{4.3.16}$$

then the system (4.3.10) can be rewritten as

$$\tilde{S}_1(\nu, T_1)\tilde{G} + S_1(\nu, T_1)\hat{G} = \begin{pmatrix} \hat{y}^{(0)} \\ \hat{y}^{(1)} \end{pmatrix} := \hat{Y}^0. \tag{4.3.17}$$

The determinant of $\tilde{S}_1(\nu, T_1)$ is

$$\det \tilde{S}_1(\nu, T_1) = \frac{1}{2}(\nu_2 \sin 2\nu_1 T_1 - \nu_1 \sinh 2\nu_2 T_1). \tag{4.3.18}$$

It can also be rewritten for further analysis

$$\det \tilde{S}_1(\nu, T_1) = \frac{T_1}{2}(\frac{\sinh(2\nu_2 T_1)}{2\nu_2 T_1} - \frac{\sin(2\nu_1 T_1)}{2\nu_1 T_1})$$

For all x, $\frac{\sinh(x)}{x} \geq 1$ and $\frac{\sin(x)}{x} \in (-1, 1)$. Therefore the determinant is positive. Furthermore it can vanish only if $\nu_1 = \nu_2 = 0$ which is impossible. Therefore it is strictly positive for all values of $k \geq 1$. When k tends to infinity ν_1 tends to infinity and ν_2 tends to 0. Therefore $\det \tilde{S}_1$ tends to $\frac{T_1}{2}$ and is bounded from below in dependence of $\xi_1 = \frac{\pi}{b-a}$ and T_1. The matrix $\tilde{S}_1(\nu, T_1)$ is uniformly invertible, and

$$\tilde{S}_1^{-1}(\nu, T_1) = \frac{1}{\det \tilde{S}_1(\nu, T_1)} \begin{pmatrix} -\operatorname{Im}(\nu \cos \nu T_1) & -\operatorname{Im}(\sin \nu T_1) \\ -\operatorname{Re}(\nu \cos \nu T_1) & -\operatorname{Re}(\sin \nu T_1) \end{pmatrix}$$

Replace now $\tilde{G}$ in (4.3.6)

$$\hat{Y}_1(k, t) = \cos(Nt)\hat{G} + \sin(Nt)\tilde{S}_1^{-1}(\nu, T_1)(Y^0 - S_1(\nu, T_1)\hat{G}), \tag{4.3.19}$$

which defines uniquely Y_1.

Domain $O_2 = (0, T_2)$ We use here formula (4.3.8) where $\tilde{H}$ and $\hat{H}$ must be relates to enforce the boundary condition $\lambda(:, T_2) = \partial_t \lambda(\cdot, T_2) = 0$, which is

$$\begin{aligned} &(\cos(NT_2)\tilde{H} + N^{-1}\sin(NT_2)\hat{H})_2 = 0, \\ &(-N\sin(NT_2)\tilde{H} + \cos(NT_2)\hat{H})_2 = 0. \end{aligned} \tag{4.3.20}$$

Use formula (3.2.14) to obtain, since $(JH)_2 = H_1$, we obtain

$$\begin{aligned} (N^{-1}\sin(NT_2)\hat{H})_2 = &\ \left[\operatorname{Re}(\nu^{-1}\sin\nu T_2)\hat{H}_2 + \operatorname{Im}(\nu^{-1}\sin\nu T_2)\hat{H}_1\right] \\ (\cos(NT_2)\hat{H})_2 = &\ \operatorname{Re}(\cos\nu T_2)\hat{H}_2 + \operatorname{Im}(\cos\nu T_2)\hat{H}_1 \\ (\cos(NT_2)\tilde{H})_2 = &\ \operatorname{Re}(\cos\nu T_2)\tilde{H}_2 + \operatorname{Im}(\cos\nu T_2)\tilde{H}_1 \\ (-N\sin(NT_2)\tilde{H})_2 = &\ -\operatorname{Re}(\nu\sin\nu T_2)\tilde{H}_2 - \operatorname{Im}(\nu\sin\nu T_2)\tilde{H}_1 \end{aligned}$$

Define the matrices

$$S_2(\nu, T_2) = \begin{pmatrix} \operatorname{Im}(\nu^{-1}\sin\nu T_2) & \operatorname{Re}(\nu^{-1}\sin\nu T_2) \\ \operatorname{Im}(\cos\nu T_2) & \operatorname{Re}(\cos\nu T_2) \end{pmatrix}, \tag{4.3.21}$$

$$\tilde{S}_2(\nu, T_2) = \begin{pmatrix} \operatorname{Im}(\cos\nu T_2) & \operatorname{Re}(\cos\nu T_2) \\ -\operatorname{Im}(\nu\sin\nu T_2) & -\operatorname{Re}(\nu\sin\nu T_2) \end{pmatrix} \tag{4.3.22}$$

And the final condition (4.3.20) takes the form

$$\tilde{S}_2(\nu, T_2)\tilde{H} + S_2(\nu, T_2)\hat{H} = 0.$$

The determinant of $\tilde{S}_2$ is

$$\det \tilde{S}_2(\nu, T_2) = \frac{1}{2}(\nu_1 \sinh(2\nu_2 T_2) + \nu_2 \sin(2\nu_1 T_2)). \tag{4.3.23}$$

The same argument as in the other subdomain shows that it is uniformly bounded from below. Therefore $\tilde{S}_2$ is invertible, and we can replace $\tilde{H}$ in (4.3.8)

$$\hat{Y}_2(k, t) = \left[-\cos(Nt)\tilde{S}_2^{-1}(\nu, T_2)S_2(\nu, T_2) + N^{-1}\sin(Nt)\right]\hat{H}, \tag{4.3.24}$$

which defines uniquely $\hat{Y}_2$ from $\hat{H}$.

Convergence of the algorithm

Collect the informations from the previous section, and apply it to the errors, that is $Y^0 = 0$. Y_1^{n+1} is the solution of the subproblem in $(-T_1, 0)$ withe data G^n at $t = 0$, thus

$$\hat{Y}_1^{n+1} = \left[\cos(Nt)I - \sin(Nt)\tilde{S}_1^{-1}(\nu, T_1)S_1(\nu, T_1)\right]\hat{G}^n.$$

Compute now

$$\partial_t \hat{Y}_1^{n+1} = \left[-N\sin(Nt)I - N\cos(Nt)\tilde{S}_1^{-1}(\nu, T_1)S_1(\nu, T_1)\right]\hat{G}^n,$$

therefore

$$\hat{H}^{n+1} := \partial_t \hat{Y}_1^{n+1}(k, 0) = -N\tilde{S}_1^{-1}(\nu, T_1)S_1(\nu, T_1)\hat{G}^n.$$

Now $\hat{Y}_2^{n+1}$ is the solution of the subproblem in $(0, T_2)$ with data $\hat{H}^{n+1}$:

$$\hat{Y}_2^{n+1}(k, t) = \left[-\cos(Nt)\tilde{S}_2^{-1}(\nu, T_2)S_2(\nu, T_2) + N^{-1}\sin(Nt)\right]\hat{H}^{n+1},$$

and the value at $t = 0$ is

$$\hat{Y}_2^{n+1}(k, 0) = -\tilde{S}_2^{-1}(\nu, T_2)S_2(\nu, T_2)\hat{H}^{n+1} = \tilde{S}_2^{-1}(\nu, T_2)S_2(\nu, T_2)N\tilde{S}_1^{-1}(\nu, T_1)S_1(\nu, T_1)\hat{G}^n.$$

defining the convergence matrix by

$$M = \tilde{S}_2^{-1}(\nu, T_2)S_2(\nu, T_2)N\tilde{S}_1^{-1}(\nu, T_1)S_1(\nu, T_1),\ M_{DN}(k, \theta, T_1, T_2) = (1-\theta)I + \theta M, \tag{4.3.25}$$

we find for the iteration

$$\hat{G}^{n+1} = M_{DN}\hat{G}^n.$$

The convergence properties

A long computation shows that

Lemma 6.

$$\tilde{S}_1^{-1}S_1 = \frac{1}{\det \tilde{S}_1}\begin{pmatrix} -\nu_2\cos^2\nu_1 t & \nu_1\sinh^2\nu_2 t \\ -\nu_1\cosh^2\nu_2 t & \nu_2\sin^2\nu_1 t \end{pmatrix}$$

$$N\tilde{S}_1^{-1}S_1 = \frac{1}{\det \tilde{S}_1}\begin{pmatrix} \nu_1\nu_2(\cosh^2\nu_2 t - \cos^2\nu_1 t) & \nu_1^2\sinh^2\nu_2 t - \nu_2^2\sin^2\nu_1 t \\ -(\nu_1^2\cosh^2\nu_2 t + \nu_2^2\cos^2\nu_1 t) & \nu_1\nu_2(\cosh^2\nu_2 t - \cos^2\nu_1 t) \end{pmatrix}$$

$$\tilde{S}_2^{-1} S_2 = \frac{1}{|\nu|^2 \det \tilde{S}_2} \begin{pmatrix} \nu_1\nu_2(\cosh^2 \nu_2 t - \cos^2 \nu_1 t) & -(\nu_1^2 \cosh^2 \nu_2 t + \nu_2^2 \cos^2 \nu_1 t) \\ \nu_1^2 \sinh^2 \nu_2 t - \nu_2^2 \sin^2 \nu_1 t & \nu_1\nu_2(\cosh^2 \nu_2 t - \cos^2 \nu_1 t) \end{pmatrix},$$

with

$$\det \tilde{S}_1(t) = \frac{1}{2}(\nu_2 \sin 2\nu_1 t - \nu_1 \sinh 2\nu_2 t), \det \tilde{S}_2(t) = \frac{1}{2}(\nu_2 \sin 2\nu_1 t + \nu_1 \sinh 2\nu_2 t).$$

Symmetric case, $T_1 = T_2$

Theorem 4.3.1. *The iteration matrix without relaxation M has two negative eigenvalues, given by*

$$\begin{aligned} u &= \nu_1\nu_2(\cosh^2 \nu_2 T_1 - \cos^2 \nu_1 T_1), \\ v &= \nu_1^2 \cosh^2 \nu_2 T_1 + \nu_2^2 \cos^2 \nu_1 T_1, \\ w &= \nu_1^2 \sinh^2 \nu_2 T_1 - \nu_2^2 \sin^2 \nu_1 T_1 \end{aligned} \tag{4.3.26}$$

$$\lambda^{\pm} = \frac{4\big(u^2 + \frac{v^2+w^2}{2} \pm |\nu|^2 \sqrt{u^2 + (\frac{v+w}{2})^2}\,\big)}{|\nu|^2(\nu_2^2 \sin^2 2\nu_1 T_1 - \nu_1^2 \sinh^2 2\nu_2 T_1)} \tag{4.3.27}$$

Proof.

$$\tilde{S}_2^{-1} S_2(T_1) = \frac{1}{|\nu|^2 \det \tilde{S}_2(T_1)} \begin{pmatrix} u & -v \\ w & u \end{pmatrix}, \; N\tilde{S}_1^{-1} S_1(T_1) = \frac{1}{\det \tilde{S}_1(T_1)} \begin{pmatrix} u & w \\ -v & u \end{pmatrix}$$

Therefore

$$(\tilde{S}_2^{-1} S_2 N \tilde{S}_1^{-1} S_1)(T_1) = \frac{1}{|\nu|^2 \det \tilde{S}_2(T_1) \det \tilde{S}_1(T_1)} R, \; R = \begin{pmatrix} u^2 + v^2 & u(w-v) \\ u(w-v) & u^2 + w^2 \end{pmatrix}$$

The matrix R is symmetric, its invariants are

$$\operatorname{Tr} R = 2u^2 + v^2 + w^2 > 0, \; \det R = (u^2 + vw)^2 > 0.$$

The reduced discriminant of the characteristic equation is

$$(\operatorname{Tr} R/2)^2 - \det R = (v-w)^2 (u^2 + (\frac{v+w}{2})^2) > 0.$$

Therefore R has two positive eigenvalues,

$$u^2 + \frac{v^2+w^2}{2} \pm \sqrt{(v-w)^2 (u^2 + (\frac{v+w}{2})^2)}.$$

Now

$$\det \tilde{S}_2 \det \tilde{S}_1 = \frac{1}{4}(\nu_2^2 \sin^2 2\nu_1 t - \nu_1^2 \sinh^2 2\nu_2 t) < 0,$$

which gives formula (4.3.27).

□

These formulas are very useful to study the convergence of the algorithm.

Lemma 7. *For fixed k (defined by a discretization) the asymptotic in T_1 is*

$$\lambda^{\pm} = -(1 \pm \frac{|\nu|}{\nu_1} \cosh^{-2}(\nu_2 T_1) + o(\cosh^{-2}(\nu_2 T_1)). \tag{4.3.28}$$

For fixed T_1 (defined by the problem) the asymptotics in ν_1 (which is equivalent to k) is

$$\lambda^{+} = -4\frac{\nu_1^2}{T_1^2} + o(\nu_1^2), \quad \lambda^{-} = -\frac{T_1^2}{4\nu_1^2} + o(\nu_1^{-2}). \tag{4.3.29}$$

Proof. To obtain the expansion in T_1, just use the same arguments as in chapter 3. The expansion in ν_1 is more tricky since the two eigenvalues don't have the same order of magnitude. Just expand the determinant and the trace of M at first order in ν_1 to get the sum and product of the eigenvalues

$$\lambda^{+} + \lambda^{-} \sim -4\frac{\nu_1^2}{T_1^2}, \quad \lambda^{+}\lambda^{-} \sim 1.$$

From this we can finish the proof. □

These asymptotics give us informations on the convergence behavior.

Theorem 4.3.2 (Convergence behavior of the Dirichlet Neumann algorithm)**.**

- *For any T, there is a k_0, for any $k \geq k_0$, $\rho(M) > 1$. Consequently, there is no θ such that the continuous algorithm converges.*

- *For the discrete algorithm associated to k_{max}, suppose $\rho(M)$ is reached for $-\lambda_+(k_{max})$. Then there exists a θ_0, for $0 < \theta < \theta_0$, the algorithm converges.*

Proof. 1. The first item only follow from Theorem 7, using the fact that

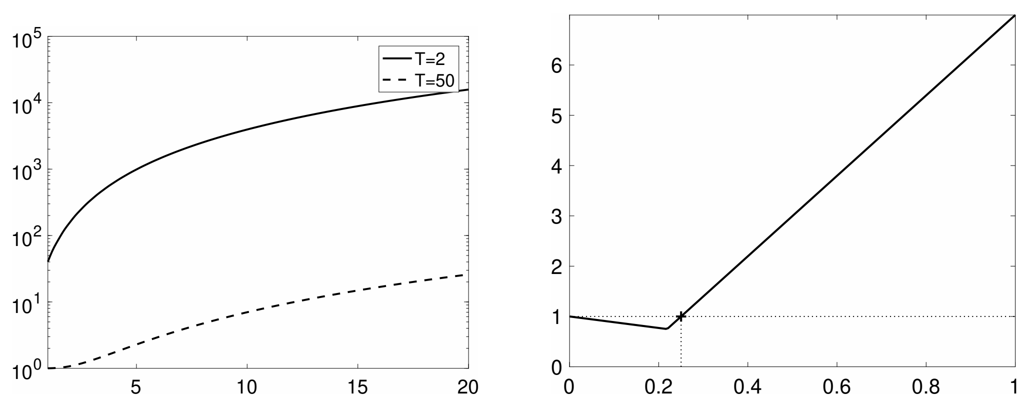
$$\lim_{k \to +\infty} \rho(M) = +\infty.$$

2. The second proof is entirely similar to the proof of Theorem 4.2.1, yielding

$$\theta_0 = \frac{2}{1 - \lambda_+(k_{max})}.$$

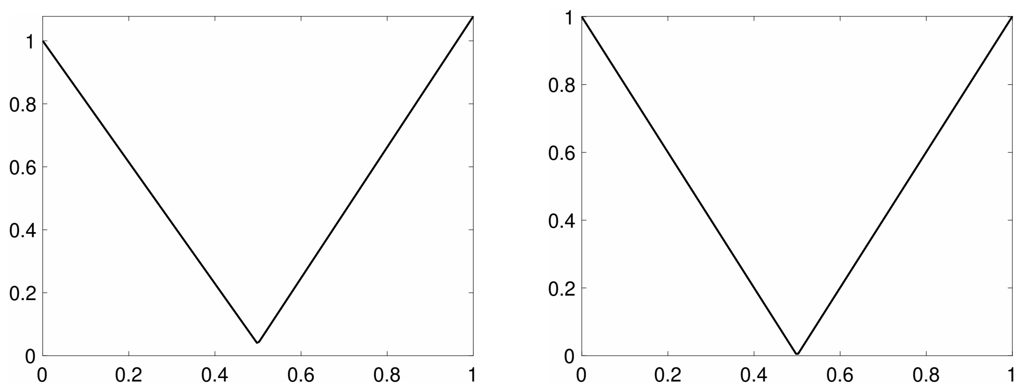
□

The previous theorems are illustrated on Figure 4.5. On Figure 4.5a, we have plotted the convergence factor without relaxation. As demonstrated, the convergence factor blows up as k goes to infinity. Then, we present on Figure 4.5b the convergence factor of the relaxed algorithm for $T = 100$ and $k_{\max} = 20$. The parameter θ has to be chosen very small to make the algorithm converge: indeed, θ_0 goes to 0 as $k_{\max}$ increases. The experience is reproduced for $T = 500$ and $T = 1000$ on Figure 4.6. Note that the optimum is reached for θ around 0.5, which corresponds to the best parameter for $T_1 = \infty$. As also predicted, the optimal convergence rate is approaching 0 as T increases.



(a) Convergence factor w.r.t k (b) Convergence factor w.r.t. θ

Figure 4.5: Illustrations of the convergence of the D.N. algorithm with and without relaxation from Theorem 4.3.2



(a) $T = 500$

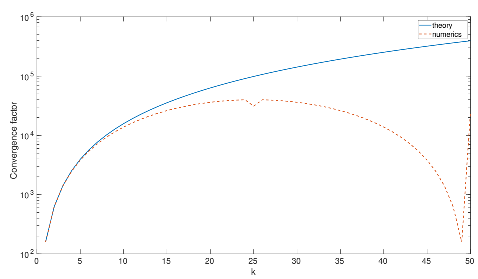
(b) $T = 1000$

Figure 4.6: Convergence factor w.r.t. θ

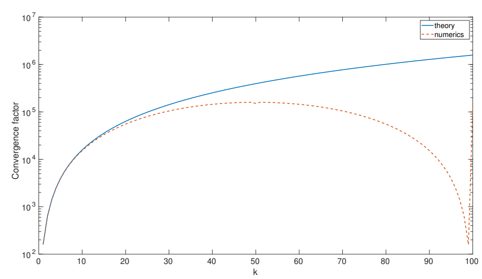
Remark 6. *The behavior of the Dirichlet-Neumann algorithm is opposite to that of the inherited algorithm: it is better converging for low frequency, while the inherited algorithm converges better for low frequencies.*

4.3.5 Numerical comparison

We finally evaluate the performances of the Dirichlet Neumann algorithm for the discrete finite volume scheme described in Chapter 2. Figure 4.7 displays the numerical and convergence factor without relaxation for $N = 50$ and $N = 100$. As for the inherited algorithm, theoretical and numerical convergence factors coincide for low frequencies but not for large ones. Here again, it probably means that the numerical convergence factor tends to the theoretical one but non uniformly in frequency. Besides, the surprising behaviour of the numerical convergence factor for the largest frequency is not understood.



(a) $N = 51$



(b) $N = 101$

Figure 4.7: Numerical and theoretical convergence factor with respect to k for the unrelaxed Dirichlet Neumann algorithm

Chapter 5

Bibliography

Bibliography

[1] Fatiha Alabau-Boussouira, Roger Brockett, Olivier Glass, Jérôme Le Rousseau, Enrique Zuazua, Sylvain Ervedoza, and Enrique Zuazua. The wave equation: Control and numerics. *Control of Partial Differential Equations: Cetraro, Italy 2010, Editors: Piermarco Cannarsa, Jean-Michel Coron*, pages 245–339, 2012.

[2] M Asch and G Lebeau. Geometrical aspects of exact boundary controllability for the wave equation-a numerical study. *ESAIM: Control, Optimisation and Calculus of Variations*, 3:163–212, 1998.

[3] Aurélien Babarit. On the park effect in arrays of oscillating wave energy converters. *Renewable Energy*, 58:68–78, 2013.

[4] Andrew T. Barker and Martin Stoll. Domain decomposition in time for PDE-constrained optimization. *Comput. Phys. Commun.*, 197:136–143, 2015.

[5] JD Benamou. Domain decomposition, optimal control of systems governed by partial differential equations, and synthesis of feedback laws. *Journal of optimization theory and applications*, 102:15–36, 1999.

[6] Jean-David Benamou. A domain decomposition method for control problems. In *DD9 Proceedings*, pages 266–273. Citeseer, 1998.

[7] Jean-David Benamou and Bruno Desprès. A domain decomposition method for the Helmholtz equation and related optimal control problems. *J. Comput. Phys.*, 136(1):68–82, 1997.

[8] Daniel Bennequin, M Gander, and Laurence Halpern. A homographic best approximation problem with application to optimized Schwarz

waveform relaxation. *Mathematics of Computation*, 78(265):185–223, 2009.

[9] Petter E. Bjorstad and Olof B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM Journal on Numerical Analysis*, 23(6):1097–1120, 1986.

[10] Filipa Caetano, Martin J Gander, Laurence Halpern, Jérémie Szeft//el, et al. Schwarz waveform relaxation algorithms for semilinear reaction-diffus//ion equations. *Networks Heterog. Media*, 5(3):487–505, 2010.

[11] T. Carraro, Michael Geiger, and R. Rannacher. Indirect multiple shooting for nonlinear parabolic optimal control problems with control constraints. *SIAM J. Sci. Comput.*, 36(2):452–481, 2014.

[12] Faycal Chaouqui, Gabriele Ciaramella, Martin J Gander, and Tommaso Vanzan. On the scalability of classical one-level domain-decomposition methods. *Vietnam Journal of Mathematics*, 46:1053–1088, 2018.

[13] Guy Chavent. *Nonlinear least squares for inverse problems: theoretical foundations and step-by-step guide for applications*. Springer Science & Business Media, 2010.

[14] Gabriele Ciaramella, Laurence Halpern, and Luca Mechelli. Convergence analysis and optimization of a Robin Schwarz waveform relaxation method for periodic parabolic optimal control problems, 2022.

[15] Gabriele Ciaramella, Felix Kwok, and Georg Müller. A Nonlinear Optimized Schwarz Preconditioner for Elliptic Optimal Control Problems. In *Domain Decomposition Methods in Science and Engineering XXVI*, pages 391–398. Springer, 2023.

[16] Gabriele Ciaramella and Luca Mechelli. An overlapping waveform relaxation preconditioner for economic optimal control problems with state constraints. In *Domain Decomposition Methods in Science and Engineering XXVI*, pages 461–469. Springer, 2023.

[17] Nicolae Cîndea and Arnaud Münch. A mixed formulation for the direct approximation of the control of minimal L^2-norm for linear type wave equations. *Calcolo*, 52(3):245–288, 2015.

[18] Xiaoying Dai and Yvon Maday. Stable parareal in time method for first- and second-order hyperbolic systems. *SIAM J. Sci. Comput.*, 35(1):a52–a78, 2013.

[19] Federico Danieli and Andrew J Wathen. All-at-once solution of linear wave equations. *Numerical Linear Algebra with Applications*, 28(6):e2386, 2021.

[20] Bérangère Delourme and Laurence Halpern. A complex homographic best approximation problem. Application to optimized Robin-Schwarz algorithms, and optimal control problems. *SIAM J. Numer. Anal.*, 59(3):1769–1810, 2021.

[21] Xiaodi Deng and Matthias Heinkenschloss. A parallel-in-time gradient-type method for discrete time optimal control problems. *Preprint, Department of Computational and Applied Mathematics, Rice University*, 2016.

[22] Victorita Dolean, Martin J Gander, Walid Kheriji, Felix Kwok, and Roland Masson. Nonlinear preconditioning: How to use a nonlinear Schwarz method to precondition Newton's method. *SIAM Journal on Scientific Computing*, 38(6):A3357–A3380, 2016.

[23] Victorita Dolean, Pierre Jolivet, and Frédéric Nataf. *An introduction to domain decomposition methods: algorithms, theory, and parallel implementation.* SIAM, 2015.

[24] Victorita Dolean, Pierre Jolivet, Pierre-Henri Tournier, Laure Combe, Stéphane Operto, and Sebastian Riffo. Large-scale finite-difference and finite-element frequency-domain seismic wave modelling with multi-level domain-decomposition preconditioner. *arXiv preprint arXiv:2103.14921*, 2021.

[25] Victorita Dolean, Frédéric Nataf, Robert Scheichl, and Nicole Spillane. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. *Computational Methods in Applied Mathematics*, 12(4):391–414, 2012.

[26] Maksymilian Dryja and Olof B Widlund. *Additive Schwarz methods for elliptic finite element problems in three dimensions.* New York University. Courant Institute of Mathematical Sciences. Computer . . . , 1991.

[27] Olivier Dubois, Martin J Gander, Sébastien Loisel, Amik St-Cyr, and Daniel B Szyld. The optimized Schwarz method with a coarse grid correction. *SIAM Journal on Scientific Computing*, 34(1):A421–A458, 2012.

[28] Matthew Emmett and Michael Minion. Toward an efficient parallel in time method for partial differential equations. *Communications in Applied Mathematics and Computational Science*, 7(1):105–132, 2012.

[29] R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. Maclachlan, and J. B. Schroder. Parallel time integration with multigrid. *SIAM J. Sci. Comput.*, 36(6):c635–c661, 2014.

[30] Martin J Gander. Optimized schwarz methods. *SIAM Journal on Numerical Analysis*, 44(2):699–731, 2006.

[31] Martin J. Gander. Analysis of the parareal algorithm applied to hyperbolic problems using characteristics. *Bol. Soc. Esp. Mat. Apl., $S\vec{e}MA$*, 42:21–35, 2008.

[32] Martin J. Gander. 50 years of time parallel time integration. In *Multiple shooting and time domain decomposition methods. MuS-TDD, Heidelberg, Germany, May 6–8, 2013*, pages 69–113. Cham: Springer, 2015.

[33] Martin J Gander et al. Schwarz methods over the course of time. *Electron. Trans. Numer. Anal*, 31(5):228–255, 2008.

[34] Martin J Gander and Stefan Güttel. Paraexp: A parallel integrator for linear initial-value problems. *SIAM Journal on Scientific Computing*, 35(2):C123–C142, 2013.

[35] Martin J. Gander and Ernst Hairer. Analysis for parareal algorithms applied to Hamiltonian differential equations. *J. Comput. Appl. Math.*, 259:2–13, 2014.

[36] Martin J Gander, Laurence Halpern, and Frédéric Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. *SIAM Journal on Numerical Analysis*, 41(5):1643–1681, 2003.

[37] Martin J. Gander, Laurence Halpern, Johann Rannou, and Juliette Ryan. A direct time parallel solver by diagonalization for the wave equation. *SIAM J. Sci. Comput.*, 41(1):a220–a245, 2019.

[38] Martin J Gander, Laurence Halpern, and Kévin Santugini Repiquet. Discontinuous coarse spaces for DD-methods with discontinuous iterates. In *Domain Decomposition Methods in Science and Engineering XXI*, pages 607–615. Springer, 2014.

[39] Martin J. Gander and Felix Kwok. Schwarz methods for the time-parallel solution of parabolic control problems. In *Domain decomposition methods in science and engineering XXII. Proceedings of the 22nd international conference on domain decomposition methods, Lugano, Switzerland, September 16–20, 2013*, pages 207–216. Cham: Springer, 2016.

[40] Martin J. Gander, Felix Kwok, and Bankim C. Mandal. Dirichlet-Neumann and Neumann-Neumann waveform relaxation for the wave equation. In *Domain decomposition methods in science and engineering XXII. Proceedings of the 22nd international conference on domain decomposition methods, Lugano, Switzerland, September 16–20, 2013*, pages 501–509. Cham: Springer, 2016.

[41] Martin J Gander, Felix Kwok, and Bankim C Mandal. Convergence of substructuring methods for elliptic optimal control problems. In *Domain Decomposition Methods in Science and Engineering XXIV 24*, pages 291–300. Springer, 2018.

[42] Martin J. Gander, Felix Kwok, and Bankim C. Mandal. Convergence of substructuring methods for elliptic optimal control problems. In *Domain decomposition methods in science and engineering XXIV. Proceedings of the 24th international conference, Svalbard, Norway, February 6–10, 2017*, pages 291–300. Cham: Springer, 2018.

[43] Martin J Gander, Felix Kwok, and Bankim C Mandal. Dirichlet–Neumann waveform relaxation methods for parabolic and hyperbolic problems in multiple subdomains. *BIT Numerical Mathematics*, 61:173–207, 2021.

[44] Martin J. Gander, Felix Kwok, and Julien Salomon. PARAOPT: a parareal algorithm for optimality systems. *SIAM J. Sci. Comput.*, 42(5):a2773–a2802, 2020.

[45] Martin J. Gander, Felix Kwok, and Hui Zhang. Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT. *Comput. Vis. Sci.*, 19(3-4):59–74, 2018.

[46] Martin J Gander, Jun Liu, Shu-Lin Wu, Xiaoqiang Yue, and Tao Zhou. Paradiag: Parallel-in-time algorithms based on the diagonalization technique. *arXiv preprint arXiv:2005.09158*, 2020.

[47] Martin J Gander and Yongxiang Liu. On the definition of dirichlet and neumann conditions for the biharmonic equation and its impact on associated schwarz methods. In *Domain Decomposition Methods in Science and Engineering XXIII*, pages 303–311. Springer, 2017.

[48] Martin J Gander and Yongxiang Liu. Is there more than one dirichlet–neumann algorithm for the biharmonic problem? *SIAM Journal on Scientific Computing*, 43(3):A1881–A1906, 2021.

[49] Martin J Gander, Thibaut Lunet, Daniel Ruprecht, and Robert Speck. A unified analysis framework for iterative parallel-in-time algorithms. *arXiv preprint arXiv:2203.16069*, 2022.

[50] Martin J. Gander and Martin Neumüller. Analysis of a new space-time parallel multigrid algorithm for parabolic problems. *SIAM J. Sci. Comput.*, 38(4):a2173–a2208, 2016.

[51] Martin J. Gander and Stefan Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.

[52] Sebastian Götschel and Michael L. Minion. An efficient parallel-in-time method for optimization with parabolic PDEs. *SIAM J. Sci. Comput.*, 41(6):c603–c626, 2019.

[53] Wolfgang Hackbusch. Parabolic multi-grid methods. Computing methods in applied sciences and engineering VI, Proc. 6th Int. Symp., Versailles 1983, 189-197 (1984)., 1984.

[54] Thomas Hagstrom, Reginal P Tewarson, and Aron Jazcilevich. Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems. *Applied Mathematics Letters*, 1(3):299–302, 1988.

[55] Laurence Halpern, Caroline Japhet, and Jérémie Szeftel. Optimized schwarz waveform relaxation and discontinuous galerkin time stepping

for heterogeneous problems. *SIAM Journal on Numerical Analysis*, 50(5):2588–2611, 2012.

[56] Matthias Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *J. Comput. Appl. Math.*, 173(1):169–198, 2005.

[57] Matthias Heinkenschloss and Michael Herty. A spatial domain decomposition method for parabolic optimal control problems. *J. Comput. Appl. Math.*, 201(1):88–111, 2007.

[58] Matthias Heinkenschloss and Hoang Nguyen. Neumann-Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems. *SIAM J. Sci. Comput.*, 28(3):1001–1028, 2006.

[59] Andreas Hessenthaler, Ben S Southworth, David Nordsletten, Oliver Röhrle, Robert D Falgout, and Jacob B Schroder. Multilevel convergence analysis of multigrid-reduction-in-time. *SIAM Journal on Scientific Computing*, 42(2):A771–A796, 2020.

[60] Juan-Antonio Infante and Enrique Zuazua. Boundary observability for the space-discretizations of the $1-d$ wave equation. *C. R. Acad. Sci., Paris, Sér. I, Math.*, 326(6):713–718, 1998.

[61] Caroline Japhet. *Méthode de décomposition de domaine et conditions aux limites artificielles en mécanique des fluides: méthode Optimisée d'Orde 2.* PhD thesis, Université Paris-Nord-Paris XIII, 1998.

[62] Felix Kwok. On the time-domain decomposition of parabolic optimal control problems. In *Domain decomposition methods in science and engineering XXIII. Proceedings of the 23rd international conference, Jeju Island, South Korea, July 6–10, 2015*, pages 55–67. Cham: Springer, 2017.

[63] Felix Kwok and Benjamin W Ong. Schwarz waveform relaxation with adaptive pipelining. *SIAM Journal on Scientific Computing*, 41(1):A339–A364, 2019.

[64] J. E. Lagnese, Günter Leugering, and E. J. P. G. Schmidt. *Modeling, analysis and control of dynamic elastic multi-link structures.* Syst. Control Found. Appl. Boston, MA: Birkhäuser, 1994.

[65] John E Lagnese and Günter Leugering. Time-domain decomposition of optimal control problems for the wave equation. *Systems & control letters*, 48(3-4):229–242, 2003.

[66] Liliana León and Enrique Zuazua. Boundary controllability of the finite-difference space semi-discretizations of the beam equation. *ESAIM: Control, Optimisation and Calculus of Variations*, 8:827–862, 2002.

[67] Jacques Louis Lions. *Optimal control of systems governed by partial differential equations*, volume 170. Springer, 1971.

[68] Pierre Louis Lions. Interprétation stochastique de la méthode alternée de schwarz. *CR Acad. Sci. Paris*, 268:325–328, 1978.

[69] Pierre-Louis Lions. On the schwarz alternating method. iii: a variant for nonoverlapping subdomains. In *Third international symposium on domain decomposition methods for partial differential equations*, volume 6, pages 202–223. SIAM Philadelphia, 1990.

[70] Pierre-Louis Lions et al. On the schwarz alternating method. i. In *First international symposium on domain decomposition methods for partial differential equations*, volume 1, page 42. Paris, France, 1988.

[71] PL Lions. On the schwarz alternating method, ii, t. chan, r. glowinski, j. périaux, and o. widlund, editors. In *Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, Los Angeles, California*, pages 47–70, 1988.

[72] Wenyue Liu and Keying Ma. An iterative non-overlapping domain decomposition method for optimal boundary control problems governed by parabolic equations. *IAENG, Int. J. Appl. Math.*, 46(3):291–297, 2016.

[73] Paola Loreti and Michel Mehrenberger. An Ingham type proof for a two-grid observability theorem. *ESAIM, Control Optim. Calc. Var.*, 14(3):604–631, 2008.

[74] Ch. Lubich and A. Ostermann. Multi-grid dynamic iteration for parabolic equations. *BIT*, 27:216–234, 1987.

[75] Yvon Maday and Gabriel Turinici. A parareal in time procedure for the control of partial differential equations. *Comptes Rendus Mathematique*, 335(4):387–392, 2002.

[76] Bankim C. Mandal. Neumann-Neumann waveform relaxation algorithm in multiple subdomains for hyperbolic problems in 1d and 2d. *Numer. Methods Partial Differ. Equations*, 33(2):514–530, 2017.

[77] Ludovic Métivier. *Une méthode d'inversion non linéaire pour l'imagerie sismique haute résolution.* PhD thesis, Université Paris-Nord-Paris XIII, 2009.

[78] Ludovic Métivier, Patrick Lailly, Florence Delprat-Jannaud, and Laurence Halpern. A 2d nonlinear inversion of well-seismic data. *Inverse Problems*, 27(5):055005, 2011.

[79] Frédéric Nataf, Francois Rogier, and Eric de Sturler. *Optimal interface conditions for domain decomposition methods.* PhD thesis, CMAP Ecole Polytechnique, 1994.

[80] Mihaela Negreanu and Enrique Zuazua. Convergence of a multigrid method for the controllability of a 1-D wave equation. *C. R., Math., Acad. Sci. Paris*, 338(5):413–418, 2004.

[81] Hieu Nguyen and Richard Tsai. A stable parareal-like method for the second order wave equation. *J. Comput. Phys.*, 405:26, 2020. Id/No 109156.

[82] Benjamin Ong, Scott High, and Felix Kwok. Pipeline schwarz waveform relaxation. In *Domain Decomposition Methods in Science and Engineering XXII*, pages 363–370. Springer, 2016.

[83] Benjamin W. Ong and Bankim C. Mandal. Pipeline implementations of Neumann-Neumann and Dirichlet-Neumann waveform relaxation methods. *Numer. Algorithms*, 78(1):1–20, 2018.

[84] Alfio Quarteroni and Alberto Valli. *Domain decomposition methods for partial differential equations.* Oxford University Press, 1999.

[85] R Tyrrell Rockafellar. *Convex analysis*, volume 11. Princeton university press, 1997.

[86] Daniel Ruprecht. Wave propagation characteristics of Parareal. *Comput. Vis. Sci.*, 19(1-2):1–17, 2018.

[87] Soura Sana and Bankim C Mandal. Dirichlet-neumann and neumann-neumann waveform relaxation algorithms for time fractional sub-diffusion and diffusion-wave equations. *arXiv preprint arXiv:2212.12366*, 2022.

[88] H. A. Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, 1870.

[89] Andrea Toselli and Olof Widlund. *Domain decomposition methods-algorithms and theory*, volume 34. Springer Science & Business Media, 2004.

[90] Stefan Ulbrich. Generalized SQP methods with "parareal" time-domain decomposition for time-dependent PDE-constrained optimization. In *Real-time PDE-constrained optimization*, pages 145–168. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2007.

[91] Stefan Ulbrich. Preconditioners based on "parareal" time-domain decomposition for time-dependent pde-constrained optimization. In *Multiple Shooting and Time Domain Decomposition Methods: MuS-TDD, Heidelberg, May 6-8, 2013*, pages 203–232. Springer, 2015.

[92] Chen-Ye Wang, Yao-Lin Jiang, and Zhen Miao. Time domain decomposition of parabolic control problems based on discontinuous Galerkin semi-discretization. *Appl. Numer. Math.*, 176:118–133, 2022.

[93] Enrique Zuazua. Propagation, observation, and control of waves approximated by finite difference methods. *SIAM Rev.*, 47(2):197–243, 2005.