



**DOMAIN DECOMPOSITION METHODS FOR  
TIME-HARMONIC ELASTIC WAVES**

BY

**Romain Brunet**

Supervised by

**Victorita Dolean**

*A thesis submitted to the  
Department of Mathematics and Statistics  
University of Strathclyde  
in partial fulfilment of the require for the degree of*

**Doctor of Philosophy**

**Academic Field Mathematics**

*This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.*

*The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.*

*Signed:*

*Date:*

## Abstract

*The construction of optimal solvers for high frequency Helmholtz-type equations is highly problematic. After discretisation of the previous equations by a finite element method, the underlying linear systems are usually large and difficult to solve both by direct and iterative methods. Domain decomposition methods are hybrid methods in the sense that they use an iterative coupling of smaller problems that are solved by direct methods, and rely on the splitting the global problem into local problems on smaller subdomains. These methods can be used as iterative solvers but also as preconditioners in a Krylov type method. That is the reason why transmission conditions between subdomains are very important.*

*In this manuscript, we start by an overview of main domain decomposition methods and focus first on their use as preconditioners. Then we consider these methods from an iterative point of view and perform a convergence study of non-overlapping and overlapping Schwarz methods with Dirichlet and Robin interface conditions, by analysing their behaviour and conclude on their convergence properties which prove to be very poor when used as solvers. The theoretical findings are illustrated by numerical results. Then we present more sophisticated methods, namely the optimised Schwarz algorithms, which use more effective transmission conditions depending on some parameters which are solutions of min-max problems.*

*The Schwarz preconditioners defined previously were one-level, meaning only the information from the neighbouring domains is used. This has the undesired consequence that the number of iterations needed to reach convergence increases with the number of subdomains. For this reason we have tested numerically two-level preconditioners, based on a coarse grid correction, this very simple idea giving promising results.*

## Acknowledgments

First of all, I would like to thank my thesis reviewers: Dr Sébastien Loisel from the University of Heriot-Watt and Dr. Gabriel Barrenechea from the University of Strathclyde, for their availability and careful reading of this manuscript which will allow me to gain a new perspective and insight on my research work. I am also thankful to staff members of the Department of Mathematics and Statistics at University of Strathclyde for their help and advice during the last years.

I would like to address my sincere thanks to Prof. Martin J. Gander from the University of Geneva for his ideas, useful comments, suggestions and for giving me with the opportunity to benefit from the collaboration of a renowned specialist of domain decomposition methods.

I would like to thank my parents and my family for supporting me throughout this thesis and my life in general. Thanks a lot to Andrew, Grant, Craig, Rebecca, Diego, Soizic, Mélinda, Marta, Céline, Vanessa, Angie, Mazze, Anna, Christoforos, Niobe, Nikita, Adriana, Beaumartin The Cottage crew and the LT822 office for making Glasgae feel like my second home, the “Parisiens”, “Bordelais”, “Palois” for their continued support when I was at home. This thesis would just not have been possible without all of them.

Last but clearly not least, I would like to express my sincere gratitude to Victorița Dolean for the continuous support of my PhD study and related research, for her patience, motivation, knowledge and joie-de-vivre. Her guidance helped in my research work and also in the writing of this thesis.

*A Mamie du Coq...*

# Contents

<b>Introduction</b>	<b>xiii</b>
Motivation . . . . .	xiii
Mathematical model . . . . .	xiv
Domain decomposition . . . . .	xix
Contents . . . . .	xx
<b>1 Domain decomposition methods</b>	<b>1</b>
1.1 State of the art . . . . .	1
1.2 The original Schwarz method and Lions' modification . . . . .	3
1.3 Schwarz methods as preconditioners . . . . .	5
1.4 Numerical experiments: one-level preconditioners . . . . .	7
<b>2 Classical Schwarz methods</b>	<b>17</b>
2.1 Classical Schwarz Algorithm . . . . .	17
Numerical experiments . . . . .	27
2.2 Optimal Schwarz algorithms and local approximations . . . . .	30
2.3 Absorbing boundary conditions . . . . .	33
Numerical experiments . . . . .	50
<b>3 Optimised Schwarz methods</b>	<b>54</b>
3.1 State of the Art . . . . .	54

3.2	One parameter family of transmissions conditions . . . . .	56
3.3	Higher order conditions . . . . .	75
3.3.1	Two-sided version of higher order conditions . . . . .	78
3.3.2	General higher order conditions . . . . .	82
3.4	Conclusions and future works . . . . .	84
<b>4</b>	<b>Grid coarse space</b>	<b>86</b>
4.1	The grid coarse space . . . . .	86
4.2	Numerical results . . . . .	87
<b>A</b>	<b>Matlab implementations</b>	<b>92</b>
A.1	Main script . . . . .	92
A.2	Rho - the computation of the convergence factor . . . . .	94
<b>B</b>	<b>FreeFem++ implementations</b>	<b>101</b>
B.1	Data files and definitions of macros . . . . .	102
B.2	RAS/ORAS . . . . .	104
B.3	GMRES . . . . .	108

# List of Figures

1	P/S-waves . . . . .	xvi
1.1	Original domain of the classical Schwarz algorithm . . . . .	3
1.2	Real part of the first component of the solution: <b>Test case 1</b> (left figure) and <b>Test case 2</b> (right figure) . . . . .	9
1.3	An example of mesh and solution in the transmission problem . . . . .	10
1.4	Uniform decomposition into 2x2, 4x4 and 6x6 domains. . . . .	10
1.5	METIS decomposition into 4, 16 and 36 domains. . . . .	11
1.6	RAS vs. ORAS, overlap = $4h$ ( $h$ - meshsize), 64 domains, uniform decomp (left), METIS (right) . . . . .	11
1.7	Convergence history for RAS (upper row) and ORAS (lower row) on uniform decompositions and overlap = $2h$ (left) and overlap= $4h$ (right) . . . . .	12
1.8	Convergence history for RAS (upper row) and ORAS (lower row) on METIS decompositions and overlap = $2h$ (left) and overlap= $4h$ (right) . . . . .	12
1.9	RAS vs. ORAS, $\delta=4h$ , 64 domains, unif. decomp (left), METIS (right) . . . . .	13
1.10	Convergence history for RAS on METIS decompositions and overlap= $2h$ (left) and overlap= $4h$ (right) . . . . .	14
1.11	Convergence history for ORAS on METIS decompositions and overlap= $2h$ (left) and overlap= $4h$ (right) . . . . .	14
2.1	Modulus of the eigenvalues of the iteration matrix for the classical Schwarz method with $C_p = 1$ , $C_s = 0.5$ , $\delta = 0.1$ . Left: for $\omega = 1$ . Right: for $\omega = 5$ . . . . .	22

2.2 Error in modulus at iteration 25 of the classical Schwarz method with 2 subdomains, where one can clearly identify the dominant mode in the error: Left:  $\omega = 1$ . Right:  $\omega = 5$ . . . . . 28

2.3 Spectrum of the iteration operator for the same example as in Figure 2.1, together with a unit circle centered around the point (1,0). Left:  $\omega = 1$ . Right:  $\omega = 5$  . . . . . 29

2.4 Convergence history for RAS and GMRES preconditioned by RAS for different values of  $\omega$  . . . . . 29

2.5 Modulus of the eigenvalues of the iteration matrix for the Schwarz method with absorbing TC without overlap and  $C_p = 1$ ,  $C_s = \frac{1}{2}$  and  $\rho = 1$ . Left: ( $\omega = 1$ ). Right: ( $\omega = 5$ ). . . . . 34

2.6 Spectrum of the iteration matrix for the Schwarz method with  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\rho = 1$ ,  $\omega = 1$  and  $\delta = 0$ . With absorbing  $\rho_{T_0}$  ( $k_e = 0$ ), general zeroth order  $\rho_{T_{0,e}}$  ( $k_e = 5$ ) and second order  $\rho_{T_2}$  ( $k_e = k$ ) TTC. . . . . 35

2.7 Spectrum of the iteration matrix for the optimised Taylor Schwarz method  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\rho = 1$ ,  $\omega = 5$ ,  $\delta = 0.1$ . With absorbing  $\rho_{T_0}$  ( $k_e = 0$ ), general zeroth order  $\rho_{T_{0,e}}$  ( $k_e = 5$ ) and second order  $\rho_{T_2}$  ( $k_e = k$ ) TTC. . . . . 39

2.8 Spectrum of the iteration matrix for the optimized Taylor Schwarz method with overlap  $\delta = 1$  and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\rho = 1$  and  $\omega = 1$ . Absorbing BC  $\rho_{T_0}$  ( $k_e = 0$ ), general zeroth order  $\rho_{T_{0,e}}$  ( $k_e = 5$ ) and second order  $\rho_{T_2}$  ( $k_e = k$ ) TTC. . . . . 46

2.9 Modulus of the eigenvalues of the iteration matrix close to  $k = \frac{\omega}{C_s}$  for the optimized Schwarz method with zeroth order TTC for  $\omega = 1$ . Left:  $\delta = 0.8$  (divergence). Middle:  $\delta = 0.9$  (convergence). Right:  $\delta = 1$  (convergence). . . . . 46

2.10 Error in modulus at iteration 60 of the iterative Schwarz method with TTC and convergence history ( $\omega = 5$ ,  $\delta = 2h$ ). . . . . 51

2.11 Error in modulus at iteration 60 of the iterative Schwarz method with TTC and convergence history ( $\omega = 5$ ,  $\delta = 6h$ ). . . . . 51

2.12 Convergence history for RAS and ORAS as solvers (left) and preconditioners (right) for  $\omega = 5$ ,  $2 \times 1$  subdomains different values of  $\delta$ . . . . . 52

2.13 Convergence history for RAS and ORAS as solvers (left) and preconditioners (right) for  $\omega = 5$ ,  $4 \times 4$  subdomains, different values of  $\delta$ . . . . . 53

3.1  $k_e < \frac{\omega}{C_p}$ : the spectrum of the iteration matrix for the non-overlapping Schwarz method with OIC and  $C_p = 1, C_s = \frac{1}{2}, \omega = 1$ . . . . . 59

3.2 For  $k_e < \frac{\omega}{C_p}$ , spectrum of the iteration matrix from the overlapping Schwarz method with OIC and  $C_p = 1, C_s = \frac{1}{2}, \omega = 1$ . . . . . 61

3.3 Comparison of the overlapping Schwarz method with zeroth order TTC and OIC for  $k_e < \frac{\omega}{C_p}$  with  $C_p = 1, C_s = \frac{1}{2}, \omega = 1, \delta = \frac{1}{10}$ . . . . . 64

3.4  $k_e \in \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$ : spectrum of the iteration matrix from the overlapping Schwarz method with OIC and  $C_p = 1, C_s = \frac{1}{2}, \delta = \frac{1}{10}$ . Left:  $\omega = 1$ . Middle:  $\omega = 5$ , Right: comparison with TTC. . . . . 66

3.5  $k_e > \frac{\omega}{C_p}$ : the spectrum of the iteration matrix from the non-overlapping Schwarz method with OIC and  $C_p = 1, C_s = \frac{1}{2}, \omega = 1$ . . . . . 67

3.6  $Z_2 - Z_1$  with  $C_p = 1, C_s = \frac{1}{2}, \omega = 1$ . Left: N=100. Right: N=1000. . . 69

3.7  $Z_2 - Z_1$  with  $C_p = 5, C_s = 1, \omega = 10$ . Left: N=100. Right: N=1000. . . 70

3.8  $k_e > \frac{\omega}{C_p}$ : the spectrum of the iteration matrix from the overlapping Schwarz method with OIC and  $C_p = 1, C_s = \frac{1}{2}, \omega = 1, \delta = \frac{1}{10}$ . . . . . 71

3.9 Comparison of the classical and the OIC Schwarz method for  $k_e > \frac{\omega}{C_s}$  with  $C_p = 1, C_s = \frac{1}{2}, \omega = 1, \delta = \frac{1}{10}$ . . . . . 71

3.10  $\frac{\omega}{C_p} < k_e < \frac{\omega}{C_s}$ : the spectrum of the iteration matrix for the overlapping Schwarz method with OIC and  $C_p = 1, C_s = \frac{1}{2}, \omega = 1, \delta = \frac{1}{10}$ . . . . . 73

3.11 Optimized convergence factor, i.e. the maximum modulus of the eigenvalues of the iteration matrix for the overlapping Schwarz method with OIC for  $\frac{\omega}{C_p} < k_e < \frac{\omega}{C_s}$  and  $\omega = 1, C_p = 1, C_s = \frac{1}{2}, \delta = \frac{1}{10}$ . . . . . 74

3.12 Left: Value of  $\frac{\omega}{C_s} - k_e^*$  obtained by minimising the convergence factor numerically compared to the asymptotic behaviour when the overlap  $\delta$  becomes small. Right: asymptotic behaviour of the convergence factor. . 74

3.13  $k_e < \frac{\omega}{C_p}$ : the spectrum of the iteration matrix from the overlapping Schwarz method with HOIC and  $C_p = 1, C_s = \frac{1}{2}, \omega = 1, \delta = \frac{1}{10}$ . . . . . 76

3.14 Spectrum of the iteration matrix for the Schwarz method with HOIC with  $C_p = 1, C_s = \frac{1}{2}$ . Left:  $\delta = 0$ . Right:  $\delta = \frac{1}{10}$ . . . . . 78

3.15 Spectrum of the iteration matrix for the overlapping Schwarz method with two-sided HOIC and  $C_p = 1, C_s = \frac{1}{2}, \omega = 1, \delta = \frac{1}{10}$ . . . . . 80

3.16 Spectrum of the iteration matrix from the Schwarz method with one-sided GHOIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ . Left:  $\delta = 0$ . Right:  $\delta = \frac{1}{10}$ . . . 84

4.1 Convergence history for RAS (upper row) and ORAS (lower row) on uniform decompositions and overlap  $=2h$  (left) and overlap $=4h$  (right) . 88

4.2 Convergence history for RAS (upper row) and ORAS (lower row) on METIS decompositions and overlap  $=2h$  (left) and overlap $=4h$  (right) . 89

4.3 Convergence history for ORAS and RAS on METIS decompositions and overlap  $=2h$  (left) and overlap $=4h$  (right) . . . . . 90

# List of Tables

- 1.1 Physical characteristics for the heterogeneous test case . . . . . 10
- 1.2 Preconditioners comparison for the test case 1 . . . . . 13
- 1.3 Preconditioners comparison for the test case 2 . . . . . 14
- 1.4 Preconditioners comparison for the heterogeneous test case 3 - Example 1 15
- 1.5 Preconditioners comparison for the heterogeneous test case 3 - Example 2 15
  
- 4.1 Preconditioners comparison for the test case 2 . . . . . 88
- 4.2 Preconditioners comparison for the heterogeneous test case 3 . . . . . 89

# Introduction

## Motivation

Propagation of waves in elastic media is a problem of undeniable practical importance that appears in geophysics which is also extremely interesting from a mathematical point of view. Both experimental and theoretical approaches have been designed because of an increasing interest in man-made ground vibration. Most materials have a very complex behaviour, so in order to fully describe it, a lot of properties need to be known. In classical elastodynamics, we are only concerned by isotropic and homogeneous materials with linear behaviour. For this kind of materials, it means that the properties of particle motion don't change according to the direction and the position. Linear-elastic behaviour means we can use laws such as the generalised Hooke's law, which says that the strain (deformation) of an elastic object or material is proportional to the stress applied to it. In several important applications - e.g. seismic exploration or earthquake prediction - one seeks to infer unknown material properties of the earth's subsurface by sending seismic waves down and measuring the scattered field which comes back, implying the solution of inverse problems.

In the process of solving the inverse problem (so-called "full-waveform inversion") one needs to iteratively solve the forward scattering problem, each time using an improved guess of the unknown material properties. In practice, each step is done by solving the appropriate wave equation using explicit time stepping. With this kind of methods, several phenomena, such as seismic waves in the earth and ultrasonic waves used to detect flaws in materials, can be quite accurately simulated. However in many applications the relevant signals are band-limited and it would be more efficient to solve in the frequency domain (the Helmholtz equation), except for the fact that the construction of optimal solvers for the high frequency Helmholtz equation is highly problematic.

Different numerical methods have their own range of validity and interest and numerical techniques in the space-time domain can handle almost any kind of waves in complex media but are limited mainly because of numerical dispersion or computational cost as quite an important number of time steps needs to be considered. For this reason we are interested here in the time-harmonic counterpart of Navier equations with the objective to develop new linear solvers for this equation. We will use domain decomposition methods to split the overall problem into smaller boundary value problems on subdomains and more precisely we will focus on the classical and optimised Schwarz type algorithms.

In a first instance we analyse by using the Fourier transform technique the convergence of these algorithms with the purpose of building more sophisticated and performant methods. Asymptotic results are presented and several numerical results will illustrate the theory. Due to their indefinite nature, Navier equations are difficult to solve and therefore the construction of robust algorithms is mandatory. The development of fast solvers for time harmonic problems is of great current interest and requires a combination of linear algebra (iterative methods for non-normal complex linear systems) and variational discretisations of PDEs.

## Mathematical model

An elastic material responds to an applied force by deforming and returns to its original shape upon the removal of the applied force. Thus, there is no permanent deformation within elastic behaviour. The relative geometric deformation of the solid is called *strain* and forces that occur in the solid are described as *stresses*. The linear theory of elasticity represented by the Navier-Cauchy equation models mechanical properties in a structure.

In this section, we present the fundamental equations of linearised elasticity and derive the Navier-Cauchy equation, which governs the propagation of time-harmonic waves in elastic solids. In our case, we assume small deformations which lead to linear equations. We also consider isotropic and homogeneous materials which implies that the physical coefficients are independent of the position and the direction.

In this simplified case, we have that the strain tensor  $\varepsilon(\mathbf{u})$  is linked to the stress tensor  $\sigma(\mathbf{u})$  (Hooke's law), as seen in [Gra91] leading to the following second order hyperbolic

system

$$\begin{aligned}
 \varepsilon(\mathbf{u}) &= \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \\
 \sigma(\mathbf{u}) &= 2\mu \varepsilon(\mathbf{u}) + \lambda \operatorname{div}(\mathbf{u}) \operatorname{Id}, \\
 \rho \partial_t^2 \mathbf{u} - \operatorname{div}(\sigma(\mathbf{u})) &= \mathbf{f},
 \end{aligned}
 \tag{1}$$

where  $\mathbf{u}$  is the displacement field,  $\mathbf{f}$  the source term,  $\rho$  the density that we assume real and  $\mu, \lambda \in [\mathbb{R}_+^*]^2$  the Lamé coefficients.

We also define by  $C_p, C_s$  the speeds of P- and S-waves (Figure (1))

$$\mu = \frac{E}{2(1+\nu)}, \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad C_p = \sqrt{\frac{\lambda+2\mu}{\rho}}, \quad C_s = \sqrt{\frac{\mu}{\rho}},
 \tag{2}$$

where  $E, \nu$  are the Young's modulus and the Poisson ratio. The Young modulus  $E$  is a measure of the stiffness of the solid: it describes how much force is needed to attain the given deformation and is positive. The Poisson ratio is a measure of the compressibility of the solid: it is the ratio of lateral to longitudinal strain.

The fact that  $0 < \nu < 0.5$ ,  $E > 0$  and (2) gives us

$$\mu > 0, \quad \lambda > 0, \quad C_p > \sqrt{2}C_s.
 \tag{3}$$

Earth generally opposes much less resistance to the dilations than rotations, that is why the compression waves dilation are always the first arrivals.

After plugging the tensors from (1) into the last equation we end up with the *time domain* Navier Cauchy equations

$$\rho \partial_t^2 \mathbf{u} - \mu \Delta \mathbf{u} - (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) = \mathbf{f}.
 \tag{4}$$

These equations are defined on a domain  $\Omega \subset \mathbb{R}^2$ , with Lipschitz boundary  $\Gamma = \partial\Omega$ .

Equation (4) describes both the transport of the volumic variation called pressure waves (or P-waves) represented by  $\nabla \cdot \mathbf{u}$  and small rotations called shear waves (S-waves) represented by the term  $\nabla \times \mathbf{u}$  (included in  $\Delta \mathbf{u}$ ) as illustrated in Figure (1) from [Hog].

If the source  $\mathbf{f}$  has a harmonic dependence on time, that is

$$\mathbf{f}(\mathbf{x}, t) = \widehat{\mathbf{f}}(\mathbf{x}) e^{-i\omega t},$$

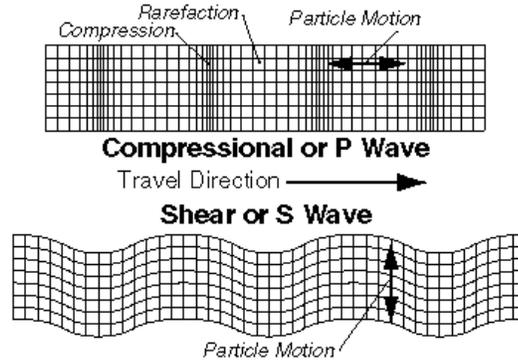


Figure 1: P/S-waves

where  $\omega$  is the angular frequency and  $\hat{\mathbf{f}}$  is the complex amplitude, then the solution will also follow a harmonic dependence

$$\mathbf{u}(\mathbf{x}, t) = \hat{\mathbf{u}}(\mathbf{x}) e^{-i\omega t},$$

where  $\hat{\mathbf{u}}$  is the complex amplitude of the oscillatory displacement field.

By replacing now  $\mathbf{f}$  and  $\mathbf{u}$  into (4) we obtain the time-harmonic counterpart of Navier-Cauchy equations

$$(5) \quad -(\omega^2 \rho \hat{\mathbf{u}} + \mu \Delta \hat{\mathbf{u}} + (\lambda + \mu) \nabla(\nabla \cdot \hat{\mathbf{u}})) = \hat{\mathbf{f}} \text{ in } \Omega.$$

For the simplicity of the notations we will abandon now the hat symbol and write the time harmonic elastic wave propagation equations as follows

$$(6) \quad -(\Delta^e + \rho\omega^2) \mathbf{u} = \mathbf{f} \text{ in } \Omega, \quad \Delta^e = [\mu\Delta + (\lambda + \mu)\nabla(\nabla \cdot)].$$

We want to solve these equations in the two dimensional domain  $\Omega \subset \mathbb{R}^2$  with a Lipschitz boundary  $\Gamma = \partial\Omega$ . By dot multiplying the equation (6) by the vector test function  $\mathbf{v}$  and integrating by parts we get

$$(7) \quad a(\mathbf{u}, \mathbf{v}) + \int_{\Gamma} \mathcal{T}_{\mathbf{n}}(\mathbf{u}) \cdot \mathbf{v} d\Gamma = 0,$$

where  $a(\mathbf{u}, \mathbf{v})$  is a bilinear form defined by

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} (\omega^2 \rho \mathbf{u} \cdot \mathbf{v} - \lambda \nabla \cdot \mathbf{u} \nabla \cdot \mathbf{v}) - \frac{\mu}{2} \int_{\Omega} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) : (\nabla \mathbf{v} + (\nabla \mathbf{v})^T),$$

and  $\mathcal{T}_{\mathbf{n}}$  is a local *traction operator*, which will be very useful in the sequel

$$\begin{aligned}
 \mathcal{T}_{\mathbf{n}}(\mathbf{u}) &= \mu \partial_{\mathbf{n}} \mathbf{u} + (\lambda + \mu) \mathbf{n} (\nabla \cdot \mathbf{u}) \\
 (8) \qquad &= \mu \partial_{\mathbf{n}} \mathbf{u} + \lambda \mathbf{n} \nabla \cdot \mathbf{u} + \mu (\nabla \mathbf{u} \cdot \mathbf{n} + \mathbf{n} \times \nabla \times \mathbf{u}) \\
 &= 2\mu \partial_{\mathbf{n}} \mathbf{u} + \lambda \mathbf{n} \nabla \cdot \mathbf{u} + \mu \mathbf{n} \times (\nabla \times \mathbf{u}),
 \end{aligned}$$

$\mathbf{n}$  being a unit outward normal vector to the boundary  $\Gamma$ .

We can define different types of boundary conditions (BC). For example, if  $\mathbf{g} \in (L^2(\Gamma))^2$

- Dirichlet boundary conditions :  $\mathbf{u} = \mathbf{g}$ ,
- "Neumann" or natural boundary conditions :  $\mathcal{T}_{\mathbf{n}}(\mathbf{u}) = \mathbf{g}$ ,
- Robin-type boundary conditions:  $(\mathcal{T}_{\mathbf{n}} + \mathcal{S})(\mathbf{u}) = \mathbf{g}$ , where  $\mathcal{S}$  is a two by two matrix valued operator which can possibly be tangential or pseudo-differential.

In our case we assume that the same results of well-posedness as in the case of the Helmholtz equation hold, that is the presence of at least one Robin-type or absorbing boundary condition insures the uniqueness and existence of the solution. This seems to be often the case in various applications. Under this assumption and supposing the appropriate discretisation method has been found (with enough points per wavelength) because of the oscillatory nature of the solution, we expect that Navier equations in the frequency domain will be very difficult to solve by iterative methods. This is somehow natural, as they are similar to the Helmholtz equation which are notoriously difficult to solve, see [EG12], and the Navier equations have further complications.

**Simple Recalls** Before going further let us recall some basic relations between the well known partial differential operators, that will be very useful in the sequel:

$$\begin{cases}
 \nabla \cdot (\nabla \phi) = \nabla^2 \phi = \Delta \phi, \\
 \nabla \cdot (\nabla \times \phi) = \nabla \times (\nabla \phi) = \mathbf{0}, \\
 \Delta \mathbf{u} = \nabla^2 \mathbf{u} = \nabla (\nabla \cdot \mathbf{u}) - \nabla \times (\nabla \times \mathbf{u}).
 \end{cases}$$

## Plane wave solutions and absorbing boundary conditions

In the following we will illustrate a particular solution of the time-harmonic elastic wave equations in the open space, also known as plane wave solution. By using the notion of the plane wave we can also introduce the notion of *absorbing boundary condition* as

being the condition exactly verified by those plane waves. Let us first consider the unit vector  $\mathbf{d}$  and  $\mathbf{d}^\perp$  (orthogonal vector to  $\mathbf{d}$ ). Note that in the three dimensional case  $\mathbf{d}^\perp$  is not unique but can be any vector in an orthogonal plane to  $\mathbf{d}$ .

Let us now consider the function:

$$(9) \quad \mathbf{u}_d^{pw} = \alpha \underbrace{\mathbf{d} e^{i\kappa_p \mathbf{x} \cdot \mathbf{d}}}_{\mathbf{u}_p} + \beta \underbrace{\mathbf{d}^\perp e^{i\kappa_s \mathbf{x} \cdot \mathbf{d}}}_{\mathbf{u}_s},$$

where  $\alpha$  and  $\beta$  are constants and the coefficients  $\kappa_p$  and  $\kappa_s$  are defined as follows:

$$(10) \quad \kappa_p = \frac{\omega}{c_p}, \quad \kappa_s = \frac{\omega}{c_s}.$$

An easy computation shows on one side that

$$\nabla \times \mathbf{u}_p = 0, \quad \nabla \cdot \mathbf{u}_s = 0,$$

and both  $\mathbf{u}_p$  and  $\mathbf{u}_s$  are solutions to the homogeneous time-harmonic Navier equations (5) in the free space. In this sense we can say that the plane wave solutions defined by (9) define an orthogonal decomposition using curl free and divergence free components.

Let us consider now a computational domain  $\Omega$  of boundary  $\Gamma = \partial\Omega$ . If we apply now the traction operator (8) to the two components of the plane wave solution which propagates in the direction defined by the outward normal  $\mathbf{d} = \mathbf{n}$  to the boundary  $\Gamma$  defined above we obtain:

$$\begin{aligned} \mathcal{T}^{(\mathbf{n})}(\mathbf{u}_p) &= 2\mu \nabla \mathbf{u}_p \cdot \mathbf{n} + \lambda \mathbf{n} \cdot (\nabla \cdot \mathbf{u}_p) = i\rho\omega c_p (\mathbf{n} \otimes \mathbf{n}) \mathbf{u}_p, \\ \mathcal{T}^{(\mathbf{n})}(\mathbf{u}_s) &= \mu \mathbf{n} \cdot (\nabla \times \mathbf{u}_s) = i\rho\omega c_s (\mathbf{n}^\perp \otimes \mathbf{n}^\perp) \mathbf{u}_s. \end{aligned}$$

Since  $\mathbf{u}_p$  and  $\mathbf{u}_s$  are orthogonal respectively to  $\mathbf{n}^\perp$  and  $\mathbf{n}$  we conclude that

$$\mathcal{T}^{(\mathbf{n})}(\mathbf{u}_n^{pw}) = i\rho\omega \left( c_p (\mathbf{n} \otimes \mathbf{n}) + c_s (\mathbf{n}^\perp \otimes \mathbf{n}^\perp) \right) \mathbf{u}_n^{pw} =: i\sigma_n \mathbf{u}_n^{pw}.$$

We can therefore infer that the Robin boundary condition

$$(11) \quad \boxed{\left( \mathcal{T}^{(\mathbf{n})} - i\rho\omega \left( c_p (\mathbf{n} \otimes \mathbf{n}) + c_s (\mathbf{n}^\perp \otimes \mathbf{n}^\perp) \right) \right) \mathbf{u} = \left( \mathcal{T}^{(\mathbf{n})} - i\sigma_n \right) \mathbf{u} = 0}$$

is exact for the plane waves defined by (9). This condition is also called *absorbing boundary condition*.

## Domain decomposition

After discretisation of the previous equations by a finite element method, the underlying linear systems are usually large and difficult to solve both by direct and iterative methods. Direct methods are very robust and provide the exact solution (up to the machine precision) after a finite numbers of steps but they are limited by memory requirements, which make the solution of the linear system beyond a given size impossible to obtain in practice. Iterative methods, on the other hand, generate a sequence that approximates the solution of the problem and the convergence to the appropriate solution depends on the properties of the matrix such as the condition number (in the case of symmetric positive definite matrices or more generally by the field of values for indefinite matrices).

Domain decomposition methods are hybrid methods in the sense that we use an iterative coupling of smaller problems that are solved by direct methods. By this kind of technique we hope to eliminate the inconvenient features of the two classes of methods (direct and iterative) and preserve only their advantages. The main idea behind these hybrid methods is to split the problem defined on the global domain into local problems on smaller subdomains, which can be solved independently, in parallel, and then communicate the results to the other domains in an iterative manner.

We distinguish two types of methods: *overlapping* and *non-overlapping* domain decomposition methods. In the case of non-overlapping decompositions, the subdomains have in common only the interface (the artificial boundary created by the decomposition). In the overlapping case subdomains have in common more than just the interface, which can lead to better convergence on one hand, but on the other hand redundant information has to be stored locally which can be costly from a computational point of view.

Domain decomposition methods can be used as either as iterative solvers or as preconditioners in a Krylov type method. Both aspects will be treated in this thesis. Their use as solvers is rather limited as the convergence might be very slow but it is very helpful as one can gain a lot of insight on the behaviour of these methods. That is the reason why the transmission conditions between subdomains are very important. In order to use them as solvers one can first define more effective transmission conditions at the interface between subdomains, depending on some parameters, by thus obtaining a new class of methods called *optimised Schwarz methods*. These parameters can be optimised by sophisticated techniques in order to obtain the best convergence possible of the iterative method.

The most common use of the Schwarz methods is as preconditioners which means that instead of solving the global problem defined by

$$\mathbf{A}U = \mathbf{F} \quad \text{we solve} \quad \mathbf{M}^{-1}\mathbf{A}U = \mathbf{M}^{-1}\mathbf{F}.$$

If  $\mathbf{M}^{-1}$  is a good approximation of  $\mathbf{A}^{-1}$ , then the spectral properties of  $\mathbf{M}^{-1}\mathbf{A}$  are much better than those of  $\mathbf{A}$ . The previous preconditioner is based on the decomposition into subdomains.

## Content and contributions

One of the objectives of this thesis is the development of new domain decomposition methods for the elastodynamics equations in frequency regime. We will consider these methods both from an iterative point of view (by designing and analysing Schwarz algorithms with optimised transmission conditions) and also as preconditioners in a Krylov method by exploring numerically their behaviour on some reference test cases.

**Chapter 1** In this chapter we present an overview of main domain decomposition methods and we will focus on their use as preconditioners. More precisely we will chose the simplest possible methods, that is those based on Dirichlet or Robin transmission conditions (absorbing boundary conditions) at the interface between domains. These preconditioners, called RAS and ORAS have been extensively studied in the literature but to our knowledge this is the first numerical study on the time-harmonic elastic waves equations. We perform several numerical experiments on simple two-dimensional test cases, on uniform and METIS decompositions.

**Chapter 2** In this chapter we will perform a convergence study of non-overlapping and overlapping Schwarz methods with Dirichlet and Robin interface conditions by using the Fourier transform technique. We will analyse their behaviour and conclude on their convergence properties which prove to be very poor when used as solvers. Numerical results illustrate the theoretical findings.

**Chapter 3** The conclusions from the previous chapter motivated the introduction of more sophisticated methods using more effective transmission conditions. The convergence analysis shows that is quite complicated to build a better method by minimising the maximum of the convergence factor over a range of relevant

frequencies. Since an analytical study seems out of reach, we use asymptotic methods and numerical optimisation.

**Chapter 4** In this chapter we first present a two level method. The second level is based on a coarse grid correction inspired by a method introduced by Graham et al in [GSV17a] for the Helmholtz equations and then extended to Maxwell's equations in [BDG<sup>+</sup>17]. In a first instance we only wish to explore the potential of the method on several academic test cases. The first results seem quite encouraging as we obtain as expected a convergence which is weakly dependent of the number of subdomains for the homogeneous test cases. However the two-level preconditioner performs less well in the case of heterogeneous problems.

**Appendix A** includes numerical optimisation with Matlab used in Chapter 3.

**Appendix B** contains FreeFem++ codes used to generate the numerical results.

The content of the Chapters 1 and 2 gave raise to the following contributions:

- R. Brunet, V. Dolean, M.J. Gander, *Can classical Schwarz methods for time-harmonic elastic waves converge?*, accepted for publication in the proceedings of the XXV International Conference on Domain Decomposition Methods, 2018.
- R. Brunet, V. Dolean, M.J. Gander, *Analysis of natural Schwarz algorithms and preconditioners for the solution of time-harmonic elastic waves*, paper submitted for publication.

# Chapter 1

## Domain decomposition methods and preconditioners

With the increasing demand for high-resolution simulations for complex systems and the availability of supercomputers, it has become necessary to have robust and efficient algorithms. That is, independent or weakly dependent of the physical properties of the medium such as the frequency. Computational efficiency is measured in terms of scalability (that is the optimal use of resources, leading to the smallest time to solution possible). Domain decomposition (DD) algorithms are very suitable candidates.

### 1.1 State of the art

We will start with a short introduction and a non exhaustive state of the art on domain decomposition methods. What we commonly call *classical Schwarz method* was introduced for the first time in [Sch70] in the purpose of proving the existence and uniqueness of the solution of a Dirichlet Poisson boundary value problem on a domain composed of the union of a rectangle and a circle (as seen in Figure 1.1). For those irregularly shaped domains, Fourier transform techniques (in absence of the modern functional analysis, these were the only available mathematical tools) were not applicable. The method consisted in an alternate iteration which was converging towards the solution of the boundary value problem (BVP). Later on, it has been shown that this method is in fact equivalent to a block Gauss Seidel type iteration where each of blocks of the global matrix corresponds to the discretisation of the Laplace operator on the local subdomains.

Even if the method was discovered in the 19th century, it has regained a lot of interest in the 20th century with the advent of the parallel computers. Indeed, a *parallel version* of it was introduced by P.-L. Lions in [Lio88], which represents in fact only a slight modification of the original method, yielding into a fully parallel algorithm whose algebraic counterpart is a block-Jacobi method.

Since the sequence of works of Lions (presented on the occasion on of the first international domain decomposition conferences) the literature on the topic covering various aspects of the field has been considerably enriched. We would like to mention several books and reference monographs. They are very different in content and approach and respond to various needs of mathematicians and other scientists. Among them, we could cite [SBG96] which presents the methods essentially from an algebraic point of view and by using matrix formulations of problems, illustrating them on different applications. Another reference book by Quarteroni and Valli [QV99] defines and analyses these methods on the continuous versions of BVP and PDE models, being less focused on computational aspects. Later on, Toselli and Widlund [TW05] discuss in their monograph, domain decomposition methods for finite element discretisations presenting rigorous analysis for a variety of problems and an overview of the properties of these as preconditioners. The newest book from V. Dolean, P. Jolivet and F. Nataf [DJN15], in addition to [TW05], includes also the optimized methods, new advances in coarse spaces and provides implementations in an open-source finite element software.

In this work we are interested in two different aspects of domain decomposition methods. First, their use as solvers, where we try to optimise their performance by designing new transmission conditions and secondly as preconditioners. The used of new, more sophisticated interface transmission conditions, is part of the subtopic called *Optimized Schwarz methods* which has expanded considerably in the past decades. Its origin can be found in [Lio90], where for the first time the author proposed the use more effective conditions at the interfaces between the subdomains than the usual Dirichlet or Neumann boundary conditions. These new conditions insure the convergence of the iterative version of the non-overlapping algorithms applied to a Poisson BVP. During the past two decades a rich literature and an important number of works were developed on this topic, with applications to various domains and equations.

There are several variants of domain decomposition methods used as preconditioners. The most popular is called Additive Schwarz (AS) and has been extensively analysed in [TW05] for a large class of symmetric positive definite problems. Its main property is the preservation of the symmetry of the preconditioner which makes it easy to analyse. However there is another variant called Restricted Additive Schwarz (RAS) which was

introduced by X.-C. Cai and M. Sarkis in [CS99] and whose convergence properties were proved to be better than those of the AS method. Same authors also proposed in [CS99] a Restricted Multiplicative Schwarz (RMS) preconditioner whose convergence has been analysed in [NS02]. There is also a continuous interpretation at the matrix level of the RAS in [EG03] that helps to explain why this method converges faster than AS and why it represents the natural discrete counterpart of the continuous Schwarz algorithm. We need to note that Optimised Schwarz methods can also be used as preconditioners and are known under the name of Optimized RAS (ORAS), Optimized MS (OMS) and Optimized AS (OAS) preconditioners. The effective application of these methods as preconditioners is illustrated in [SCGT07].

## 1.2 The original Schwarz method and Lions' modification

The first domain decomposition method was introduced by H. Schwarz in order to solve the following Poisson equation on an irregular domain  $\Omega$  as shown in Figure 1.1

$$(1.1) \quad \begin{cases} -\Delta u = f & \text{in } \Omega \\ u = g & \text{on } \partial\Omega \end{cases}$$

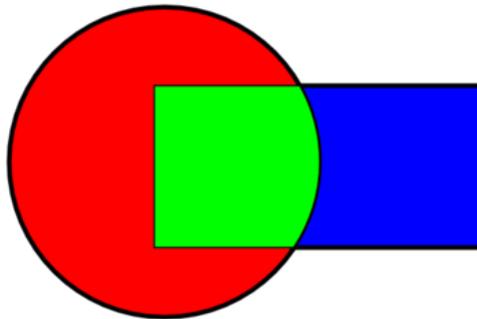


Figure 1.1: Original domain of the classical Schwarz algorithm

To solve problem (1.1) on the union of circle ( $\Omega_1$ ) and rectangle ( $\Omega_2$ ), Schwarz built an iterative method which consists in computing successive approximations on the local

subdomains on which the solution could be computed by using Fourier series and then exchanging the data between neighbouring subdomains. He proved the convergence of the iterative method to a solution meaning that the solution on the whole domain exists.

This method is now known as the *classical Schwarz method* and can be simply described as follows: given an initial guess  $u_2^0$  one solves iteratively by alternating the successive solves on both subdomains

$$(1.2) \quad \begin{cases} -\Delta u_1^{n+1} = f & \text{in } \Omega_1 \\ u_1^{n+1} = 0 & \text{on } \partial\Omega \cap \partial\Omega_1 \\ u_1^{n+1} = u_2^n & \text{on } \partial\Omega_1 \setminus \partial\Omega \end{cases} \quad \begin{cases} -\Delta u_2^{n+1} = f & \text{in } \Omega_2 \\ u_2^{n+1} = 0 & \text{on } \partial\Omega \cap \partial\Omega_2 \\ u_2^{n+1} = u_1^{n+1} & \text{on } \partial\Omega_2 \setminus \partial\Omega. \end{cases}$$

According to this definition, one can see that the solution on subdomain  $\Omega_2$  at  $n + 1$  iteration depends on the solution on subdomain  $\Omega_1$ , which is the reason why this algorithm is not parallel and its convergence is very slow. Moreover, in the case of non-overlapping subdomain the algorithm does not converge.

Later on, P.-L. Lions modified the classical Schwarz method (1.2) and proposed the following fully parallel algorithm that starting from an initial guess  $(u_1^0, u_2^0)$  solves in parallel the local problems then iterates

$$(1.3) \quad \begin{cases} -\Delta u_1^{n+1} = f & \text{in } \Omega_1 \\ u_1^{n+1} = 0 & \text{on } \partial\Omega \cap \partial\Omega_1 \\ u_1^{n+1} = u_2^n & \text{on } \partial\Omega_1 \setminus \partial\Omega \end{cases} \quad \begin{cases} -\Delta u_2^{n+1} = f & \text{in } \Omega_2 \\ u_2^{n+1} = 0 & \text{on } \partial\Omega \cap \partial\Omega_2 \\ u_2^{n+1} = u_1^n & \text{on } \partial\Omega_2 \setminus \partial\Omega. \end{cases}$$

The now parallel algorithm (1.3) is convergent but only for *overlapping subdomains* and the converge remains very slow. However, it can be proven that the bigger the overlap, the faster the convergence.

On the other hand, due to its simplicity it can be generalised easily to a well posed boundary value problem defined by the positive definite partial differential operator  $\mathcal{L}$

$$\begin{cases} \mathcal{L}u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

with some further modifications to other categories of problems.

The convergence of such an algorithm can further be improved by using more sophis-

ticated boundary conditions at the interfaces between subdomains

$$(1.4) \quad \begin{cases} -\Delta u_1^{n+1} = f & \text{in } \Omega_1 \\ u_1^{n+1} = g & \text{on } \Omega_1 \cap \partial\Omega \\ (\partial_{n_1} + p_1) u_1^{n+1} = (\partial_{n_1} + p_1) u_2^n & \text{on } \Omega_2 \cap \partial\Omega_1 \end{cases}$$

$$(1.5) \quad \begin{cases} -\Delta u_2^{n+1} = f & \text{in } \Omega_2 \\ u_2^{n+1} = g & \text{on } \Omega_2 \cap \partial\Omega \\ (\partial_{n_2} + p_2) u_2^{n+1} = (\partial_{n_2} + p_2) u_1^n & \text{on } \Omega_1 \cap \partial\Omega_2 \end{cases}$$

where  $p_1, p_2$  are well chosen constants. The first algorithm of this type has been proposed by Lions and converges even in the case of non-overlapping domains. The constants  $p_1, p_2$  can be computed by analytical or numerical techniques in order to achieve the best convergence possible of the method. We will study in more detail this kind of algorithms applied to the Navier equations in the next chapter.

It can be shown that the Schwarz method defined by (1.3) is equivalent to a block Jacobi algorithm (see [DJN15, Chapter 1.2]), in which the blocks correspond to local problems on each subdomain. Such a method is known to converge quite slowly and therefore the use of preconditioned Krylov accelerations is recommended. In this case the preconditioners will be inspired by the overlapping decompositions as we will show in the next section.

Moreover, it has been shown for the first time in [SCGT07] that from practical point of view, the use of Lions type algorithms and optimised Schwarz methods as preconditioners, is quite natural and can be simply understood within the same formalism as the classical overlapping Schwarz methods as we will see in the next section.

### 1.3 Schwarz methods as preconditioners

In this section we will present briefly the use of Schwarz methods as preconditioners. For the sake of simplicity we will limit the presentation to the discrete setting. Suppose that after the discretisation of the Navier equations, say by a finite element method, we obtain the following linear system

$$AU = F,$$

where  $A$  is the discretisation matrix on the domain  $\Omega$ ,  $\mathbf{U}$  is the vector of unknowns and  $\mathbf{F}$  is the right hand side. This system will be solved by a Krylov method (which in our case will be GMRES as the system is indefinite). To accelerate the performance of the Krylov method applied to this system we will consider two preconditioners inspired by an overlapping domain decomposition which are naturally parallelisable [DJN15, Chapter 3]. In order to introduce these preconditioners, we first need to define a certain number of ingredients necessary in their writing in algebraic form.

Let  $\mathcal{T}_h$  a triangulation of the computational domain and  $\{\mathcal{T}_{h,i}\}_{i=1}^N$  be a non-overlapping partition of this triangulation. Such a partition can be typically obtained by using a mesh partitioner like METIS [KK98]. The overlapping partition needed in our method is defined as follows. For an integer value  $l \geq 0$ , we build the decomposition  $\{\mathcal{T}_{h,i}^l\}_{i=1}^N$  such that  $\mathcal{T}_{h,i}^l$  is a set of all triangles from  $\mathcal{T}_{h,i}^{l-1}$  and all triangles from  $\mathcal{T}_h \setminus \mathcal{T}_{h,i}^{l-1}$  that have non-empty intersection with  $\mathcal{T}_{h,i}^{l-1}$ , and  $\mathcal{T}_{h,i}^0 = \mathcal{T}_{h,i}$ . With this definition the width of the overlap will be of  $2l$ . Furthermore, if  $W_h$  stands for the finite element space associated with  $\mathcal{T}_h$ ,  $W_{h,i}^l$  is the local finite element spaces on  $\mathcal{T}_{h,i}^l$  that is a triangulation of  $\Omega_i$ .

Let  $\mathcal{N}$  be the set of indices of degrees of freedom of the global finite element space  $W_h$  and  $\mathcal{N}_i^l$  the set of indices of degrees of freedom of the local finite element spaces  $W_{h,i}^l$  for  $l \geq 0$ . We define the restriction operators from the global set of degrees of freedom to the local one, by

$$R_i : W_h \rightarrow W_{h,i}^l.$$

At a discrete level this is a rectangular matrix  $|\mathcal{N}_i^l| \times |\mathcal{N}|$  such that if  $\mathbf{V}$  is the vector of degrees of freedom of  $v_h \in W_h$ , then  $R_i \mathbf{V}$  is the vector of degrees of freedom of  $W_h$  in  $\Omega_i$ .

The extension operator from  $W_{h,i}^l$  to  $W_h$  and its associated matrix are both then given by  $R_i^T$ .

In addition we introduce a partition of unity  $D_i$  as a diagonal matrix  $|\mathcal{N}_i^l| \times |\mathcal{N}_i^l|$  such that

$$(1.6) \quad Id = \sum_{i=1}^N R_i^T D_i R_i,$$

where  $Id \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{N}|}$  is the identity matrix.

With these ingredients at hand we can now present the RAS preconditioner firstly

introduced in [CS99], as described in [DJN15, Chapter 1.4]:

$$(1.7) \quad M_{RAS}^{-1} = \sum_{i=1}^N R_i^T D_i (R_i A R_i^T)^{-1} R_i.$$

In our experiments we will also use another very natural method, namely the Optimized RAS (ORAS) preconditioner which is based on local boundary value problem with Robin boundary conditions (absorbing boundary conditions). In this case, let  $B_i$  be the matrix associated to a discretisation of the corresponding local BVP on the domains  $\Omega_i$  with Robin boundary conditions on  $\partial\Omega_i \cap \partial\Omega_j$ . The definition is very similar to (1.7) except that  $R_i A R_i^T$  is replaced by  $B_i$ :

$$(1.8) \quad M_{ORAS}^{-1} = \sum_{i=1}^N R_i^T D_i B_i^{-1} R_i.$$

We will therefore solve the following preconditioned system by a Krylov method

$$M^{-1} A U = M^{-1} F$$

where  $M^{-1}$  is given by (1.7) or (1.8). Both versions (1.7) or (1.8) of the Schwarz preconditioners are called *one-level preconditioners*.

Note that one can prove that the Schwarz method in its iterative form is equivalent to the preconditioned fixed-point iteration

$$(1.9) \quad U^{n+1} = U^n + M^{-1} (F - A U^n),$$

We can then see that the solution of this iteration is given in a space spanned by powers of the matrix  $Id - M^{-1} A$  and that the solution can be naturally accelerated by a Krylov method [DJN15, Chapter 3].

## 1.4 Numerical experiments: one-level preconditioners

In this section we compare the standard RAS preconditioner (1.7) with the ORAS preconditioner (1.8), that is the one based Robin interface transmission conditions. These preconditioners are quite standard in the literature, however to our knowledge, their application to the time-harmonic elastodynamics equations has not been extensively studied. These preliminary tests on a few simple, two-dimensional configurations are

meant to explore the properties of the basic domain decomposition preconditioners in order to further investigate whether more performant methods can be developed.

In all cases the Krylov iterative solver is GMRES [SS86]. The stopping criterium of the algorithm is when the relative  $L^2$  norm of error is smaller than  $10^{-6}$ ,

$$\frac{\|\mathbf{U} - \mathbf{U}_n\|_{L^2(\Omega)}}{\|\mathbf{U} - \mathbf{U}_0\|_{L^2(\Omega)}} < 10^{-6},$$

where  $\mathbf{U}$  is the one domain solution and  $\mathbf{U}_m$  denotes the approximation of  $\mathbf{U}$  at the  $m$ -th iteration of the iterative solver. Therefore the algorithm is stopped when the criterium is achieved. The number of iterations will be a measure of the performance of each method as the cost per iteration is very similar.

The overlapping decomposition into subdomains can be uniform (e.g  $m \times m$  domains, with  $m$  domains in each direction) or generated by METIS [KK98]. In both cases,  $N$  denotes the total number of subdomains. In each case the boundary value problem is discretised using P1 elements and the computational domain is the unit square (test cases 1 and 2) and a disk with a heterogeneous medium composed of two parts (test case 3). We use a random initial guess for the GMRES iterative solver in all tests and we vary the size of the overlap and the type of the decomposition (uniform or using METIS). Numerical simulations were done by using the open source software Freefem++ [Hec12] which is a high level language specialised in variational discretisations of partial differential equations.

**Definition of the test cases.** In test cases 1 and 2 we simulate the wave propagation through a computational domain given by the unit square  $[0, 1]^2$  with Robin boundary conditions on the whole boundary. We solve the following boundary value problem

$$(1.10) \quad -(\Delta^e + \rho\omega^2) \mathbf{u} = \mathbf{f} \quad \text{in } \Omega, \quad \left(\mathcal{T}^{(\mathbf{n})} - i\sigma_{\mathbf{n}}\right) \mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega,$$

where  $\Delta^e = [\mu\Delta + (\lambda + \mu)\nabla(\nabla\cdot)]$ , with the source term  $\mathbf{g}$  chosen such that the exact solution is a plane wave  $\mathbf{u}^{inc}$  consisting both P- and S-waves like in (9) such that

$$(1.11) \quad \mathbf{u}^{inc} = \mathbf{u}_{\mathbf{d}}^{pw}, \quad \mathbf{d} = \left(\cos\left(\frac{\pi}{3}\right), \cos\left(\frac{\pi}{3}\right)\right)^T, \quad \alpha = \beta = 1.$$

Note that in the two-dimensional case considered here

$$(1.12) \quad \sigma_{\mathbf{n}} = \omega\rho \begin{pmatrix} c_p n_x^2 + c_s n_y^2 & (c_p - c_s)n_x n_y \\ (c_p - c_s)n_x n_y & c_p n_y^2 + c_s n_x^2 \end{pmatrix},$$

The physical coefficients are as follows

$$(1.13) \quad C_p = \sqrt{\frac{\lambda + 2\mu}{\rho}}, \quad C_s = \sqrt{\frac{\mu}{\rho}}, \quad \kappa_p = \frac{\omega}{C_p}, \quad \kappa_s = \frac{\omega}{C_s}, \quad \omega = 2\pi f$$

and

$$(1.14) \quad \mu = \frac{E}{2(1 + \nu)} = \rho C_s^2, \quad \lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)} = \rho(C_p^2 - 2C_s^2).$$

In the first test case (denoted by **Test case 1**) we fix some parameters:

$$(1.15) \quad C_p = 2, \quad C_s = 1, \quad \rho = 1, \quad \omega = 30.$$

In the second test case (denoted by **Test case 2**) we fix

$$(1.16) \quad E = 2 \cdot 10^{11}, \quad \nu = 0.3, \quad \rho = 7800, \quad f = 2 \cdot 10^4,$$

and the others are computed from the formulae written above.

These test cases do not necessarily correspond to accurate physical situations but they produce simple but oscillatory enough solutions reflecting the difficulties related to the solving of the problem.

An example of the real part of the first component of these solutions is depicted in Figure 1.2



Figure 1.2: Real part of the first component of the solution: **Test case 1** (left figure) and **Test case 2** (right figure)

The third example (denoted by **Test case 3**) is a transmission problem through a circular inhomogeneous media whose radius is 0.5 ( $\Omega_1$ ), surrounded by an infinite homogeneous material ( $\Omega_2$ ) with absorbing boundary conditions and heterogeneous physical

parameters that can be chosen as in the examples below

Examples	Domain	E	$\nu$	$\rho$	$\mu$	$\lambda$	$C_p$	$C_s$	f	$\omega$
<b>1</b>	$r < 0.5$	$2.10^{11}$	0.3	7800	$77.10^9$	$12.10^{10}$	5927	3142	$10^4$	$2\pi 10^4$
	$0.5 \leq r \leq 1$	$2.10^{11}$	0.47	7800	$68.10^9$	$11.10^{11}$	12588	2952	$10^4$	$2\pi 10^4$
<b>2</b>	$r < 0.5$	$2.10^{11}$	0.3	7800	$77.10^9$	$12.10^{10}$	5927	3142	$10^4$	$2\pi 10^4$
	$0.5 \leq r \leq 1$	$2.10^{11}$	0.25	7800	$80.10^9$	$80.10^9$	5547	3203	$10^4$	$2\pi 10^4$

Table 1.1: Physical characteristics for the heterogeneous test case

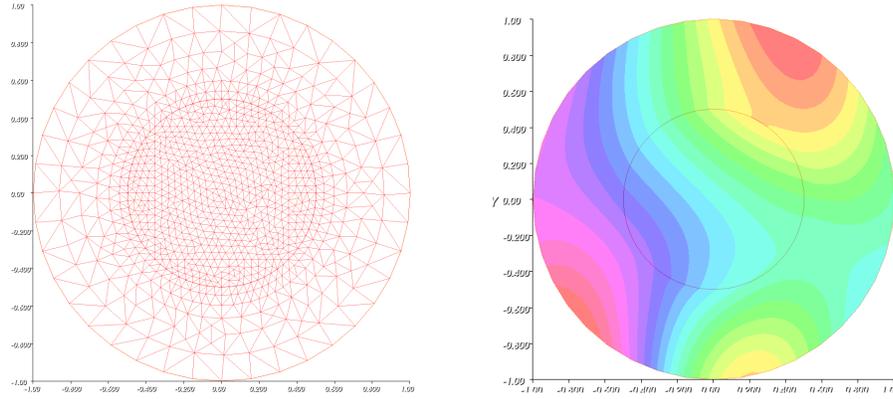


Figure 1.3: An example of mesh and solution in the transmission problem

For all these test cases we compare the performances of the RAS and ORAS preconditioners as a function of the decomposition into subdomains and the overlapping parameters. Examples of uniform and METIS decompositions can be found below

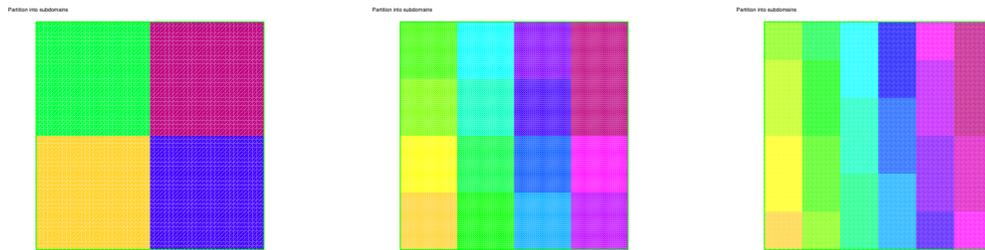


Figure 1.4: Uniform decomposition into 2x2, 4x4 and 6x6 domains.

In these preliminary test cases we deliberately focus on decompositions into  $N \times N$  domains as they correspond to the typical weak scaling tests with the one level Schwarz method used as a preconditioner. This is somehow different on what we will do in the following chapter where the analysis is focused on the decomposition into two

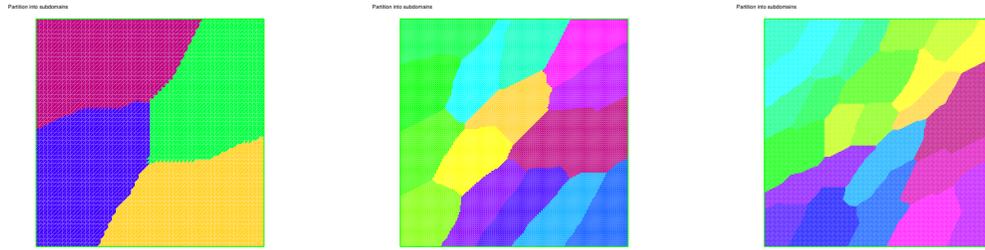


Figure 1.5: METIS decomposition into 4, 16 and 36 domains.

subdomains and for the later we will design specific numerical tests.

In the case of uniform decompositions the size of local problems is maintained fixed (e.g. equal to 20 degrees of freedom in one direction), thus the biggest problem that will be considered, contains  $8 \times 20 = 160$  degrees of freedom (dof) in one direction for a total number of  $160 \times 160 = 25600$  dofs.

**RAS and ORAS: Test case 1.** We first perform a numerical experiment on a uniform decomposition by varying the number of subdomains and the size of the overlap. In both cases we notice that when the number of subdomains increases the performance of the algorithm (in terms of number of iterations) deteriorates. As expected, when the overlap is increased, the algorithm performs better. We also notice that the ORAS preconditioner outperforms RAS (as it uses more effective transmission conditions) and that the type of the decomposition has only a little influence on the iteration count.

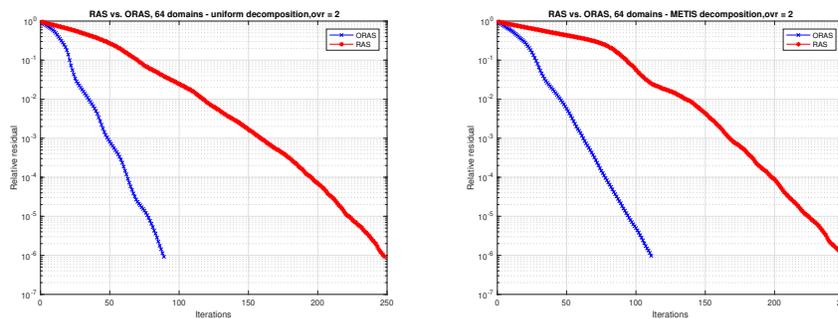


Figure 1.6: RAS vs. ORAS, overlap  $=4h$  ( $h$  - meshsize), 64 domains, uniform decomp (left), METIS (right)

The convergence history is repeated on uniform and METIS decompositions as follows

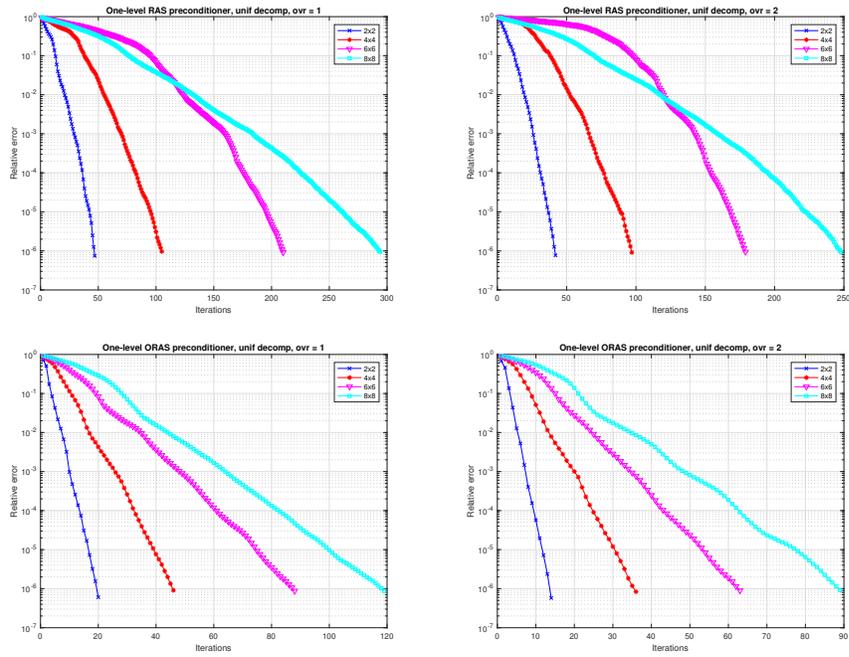


Figure 1.7: Convergence history for RAS (upper row) and ORAS (lower row) on uniform decompositions and overlap =  $2h$  (left) and overlap= $4h$  (right)

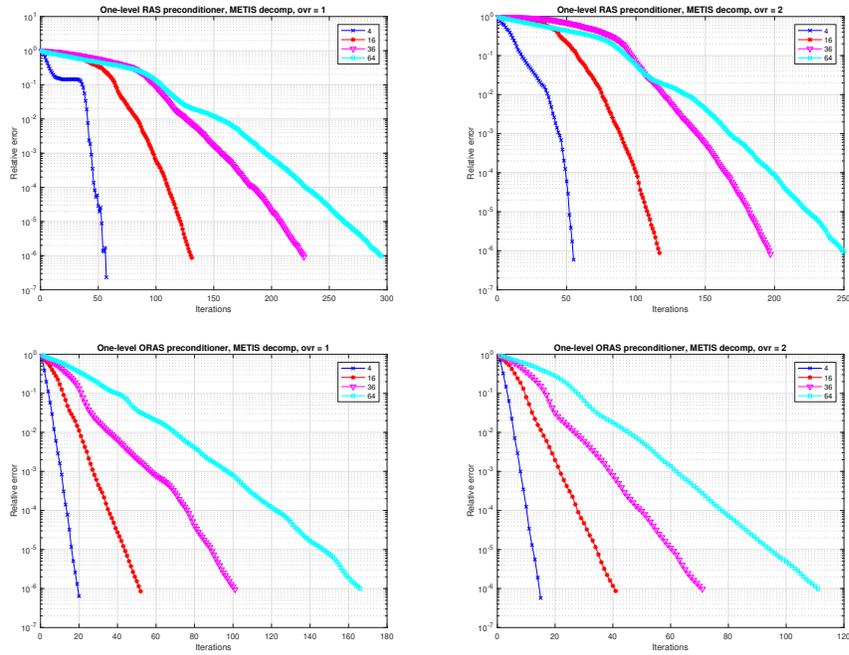


Figure 1.8: Convergence history for RAS (upper row) and ORAS (lower row) on METIS decompositions and overlap =  $2h$  (left) and overlap= $4h$  (right)

A numerical summary of the results of the previous figures is found in the table below

N	Overlap = 2h				Overlap = 4h			
	RAS		ORAS		RAS		ORAS	
	Unif	MTS	Unif	MTS	Unif	MTS	Unif	MTS
<b>4</b>	46	57	20	20	42	55	14	15
<b>16</b>	105	131	46	52	97	117	36	41
<b>36</b>	210	229	88	101	179	197	63	71
<b>64</b>	294	295	119	166	248	250	89	111

Table 1.2: Preconditioners comparison for the test case 1

In conclusion, these preliminary tests show that the one-level preconditioner is not scalable, that is the iterations increase linearly with respect to the number of subdomains in one direction and that the ORAS preconditioner is clearly better than RAS leading to an iteration count that is roughly half of that of the latter.

**RAS and ORAS: Test case 2.** We first perform similar numerical experiments as before while we presume that the problem will be more difficult to solve as the solution is more oscillatory. This is also reflected in the iteration count of the RAS and ORAS algorithms. We notice again that the ORAS preconditioner outperforms RAS as below

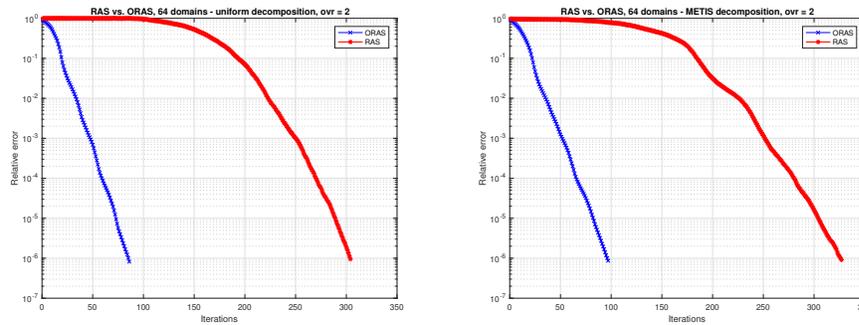


Figure 1.9: RAS vs. ORAS,  $\delta=4h$ , 64 domains, unif. decomp (left), METIS (right)

A numerical summary can be found in the table 1.3. Again these preliminary tests show that the one-level preconditioner is not scalable, that is the iterations increase linearly with respect to the number of subdomains in one direction and that the ORAS preconditioner is clearly better than RAS leading to an iteration count that is roughly on third of that of the latter. We can notice that with respect to the **Test case 1**, despite the solution being more oscillatory and the problem potentially more difficult to solve, the ORAS preconditioner is quite robust unlike the RAS preconditioner.

N	Overlap = $2h$				Overlap = $4h$			
	RAS		ORAS		RAS		ORAS	
	Unif	MTS	Unif	MTS	Unif	MTS	Unif	MTS
<b>4</b>	62	74	19	18	80	81	13	14
<b>16</b>	135	143	43	46	142	151	33	39
<b>36</b>	208	273	72	78	192	250	60	66
<b>64</b>	347	361	111	123	304	327	86	97

Table 1.3: Preconditioners comparison for the test case 2

**RAS and ORAS: Test case 3.** We perform numerical experiments on a METIS decomposition of the geometry defined in Figure 1.3 on the two problems with heterogeneous coefficients given in the Table 1.1. We start first with **Example 1**.

The convergence history of the two algorithms for the first heterogeneous case is depicted in Figures 1.10 and 1.11.

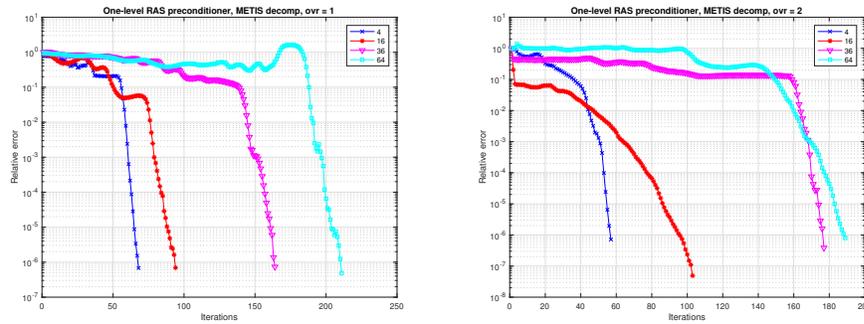


Figure 1.10: Convergence history for RAS on METIS decompositions and overlap= $2h$  (left) and overlap= $4h$  (right)

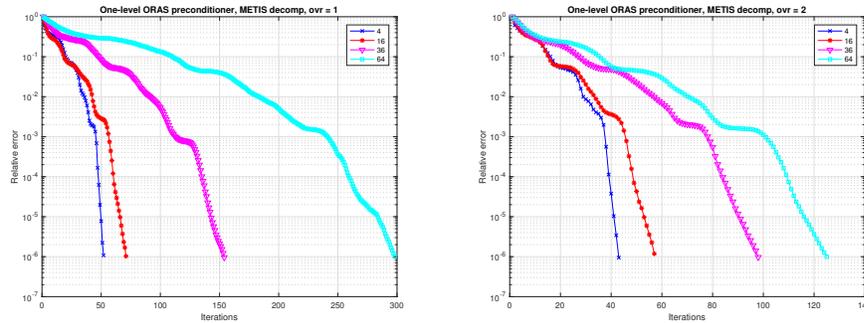


Figure 1.11: Convergence history for ORAS on METIS decompositions and overlap= $2h$  (left) and overlap= $4h$  (right)

A numerical summary can be found in the table 1.4.

N	Overlap = $2h$		Overlap= $4h$	
	RAS	ORAS	RAS	ORAS
<b>4</b>	68	52	57	43
<b>16</b>	94	71	103	57
<b>36</b>	164	154	177	98
<b>64</b>	211	298	189	125

Table 1.4: Preconditioners comparison for the heterogeneous test case 3 - Example 1

We notice here that unlike for the two other test cases the ORAS preconditioner deteriorates considerably as the number of subdomains increases when the overlap is minimal (one mesh size on each size of the non-overlapping partition). This can be explained by the fact that the domains being too small with respect to the wavelength the absorbing boundary conditions are less efficient. In turn when the overlap is larger we retrieve the behaviour we already expect with roughly 30% less iterations for ORAS with respect to RAS when the number of subdomains increases. As before and as expected, the performance deteriorates with the increase of the number of domains.

We now perform the same tests on the Example 2 from table 1.1. A numerical summary of these results can be found in the table 1.5.

N	Overlap = $2h$		Overlap= $4h$	
	RAS	ORAS	RAS	ORAS
<b>4</b>	72	49	62	42
<b>16</b>	124	59	96	49
<b>36</b>	205	99	177	73
<b>64</b>	275	128	174	86

Table 1.5: Preconditioners comparison for the heterogeneous test case 3 - Example 2

In this case the behaviour is as expected with a huge gain in number of iteration (more than half) when using ORAS with respect to RAS. Also, we could notice only a slight increase in the number of iterations when the overlap is sufficiently big.

These first preliminary tests on the time-harmonic elastic wave equations by using domain decomposition preconditioners from the literature, but adapted to our problem (specific absorbing boundary conditions at the interfaces) show on one side that the use of more sophisticated interface transmission conditions is needed as even the simplest

ones can lead to an important gain. On the other side, a deeper understanding of the impact of the interface transmission conditions will help us to further improve the convergence. Note that the convergence can also be improved by using two-level algorithms thus achieving a performance independent of the number of subdomains.

The Freefem++ codes used in these implementations can be found in the Appendix chapter B.

## Chapter 2

# Classical Schwarz methods for time-harmonic elastic waves

The purpose of this chapter is to analyse the convergence of the classical Schwarz method (and several other variants) in its iterative version by using the Fourier transform technique. This analysis will reveal quite an usual behaviour which is very different from what we observe in the case of Helmholtz or Maxwell's equations. As a consequence, we will try to improve this algorithm by constructing more effective interface transmission conditions. The simplest ones (which are of Robin type) are low order approximations of transparent boundary conditions, that we call absorbing boundary conditions. Despite their simplicity, the analysis of the underlying algorithms is already quite tedious (an asymptotic analysis and numerical algorithms are needed) and again the conclusions are very different from what we obtain in the case of the Helmholtz equation.

### 2.1 Classical Schwarz Algorithm

In this section we start by the definition of the classical Schwarz algorithm in a simple geometrical configuration and we continue by its analysis. For the time being we will limit ourselves to a two dimensional domain, knowing that a similar analysis can be performed in the three dimensional case.

We are interested in solving the Navier equations in the frequency domain

$$(2.1) \quad -(\Delta^e + \omega^2 \rho) \mathbf{u} = \mathbf{f} \quad \text{in } \Omega,$$

where the operator  $\Delta^e$  is defined by

$$(2.2) \quad \Delta^e \mathbf{u} = \mu \Delta \mathbf{u} + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}).$$

For the sake of the analysis only, we decompose the domain  $\Omega := \mathbb{R}^2$  into two unbounded possibly overlapping subdomains  $\Omega_1 := (-\infty, \delta) \times \mathbb{R}$  and  $\Omega_2 := (0, \infty) \times \mathbb{R}$ ,  $\delta \geq 0$ . On this simple configuration the convergence analysis is relatively easy to perform while providing some important insight on the behaviour of the algorithm.

Let's consider now what we call classical Schwarz algorithm

$$(2.3) \quad \begin{aligned} -(\Delta^e + \omega^2 \rho) \mathbf{u}_1^n &= \mathbf{f} && \text{in } \Omega_1, \\ \mathbf{u}_1^n &= \mathbf{u}_2^{n-1} && \text{on } x = \delta, \\ -(\Delta^e + \omega^2 \rho) \mathbf{u}_2^n &= \mathbf{f} && \text{in } \Omega_2, \\ \mathbf{u}_2^n &= \mathbf{u}_1^{n-1} && \text{on } x = 0. \end{aligned}$$

We can also build a class of optimised versions by changing the interface transmission conditions:

$$(2.4) \quad \begin{aligned} -(\Delta^e + \omega^2 \rho) \mathbf{u}_1^n &= \mathbf{f} && \text{in } \Omega_1, \\ (\mathcal{T}_1 + \mathcal{S}_1) \mathbf{u}_1^n &= (\mathcal{T}_1 + \mathcal{S}_1) \mathbf{u}_2^{n-1} && \text{on } x = \delta, \\ -(\Delta^e + \omega^2 \rho) \mathbf{u}_2^n &= \mathbf{f} && \text{in } \Omega_2, \\ (\mathcal{T}_2 + \mathcal{S}_2) \mathbf{u}_2^n &= (\mathcal{T}_2 + \mathcal{S}_2) \mathbf{u}_1^{n-1} && \text{on } x = 0, \end{aligned}$$

where the traction operators  $\mathcal{T}_j$ ,  $j = 1, 2$ , which plays for the Navier equations the role of a Neumann condition, is defined by

$$(2.5) \quad T_j(\mathbf{u}) = 2\mu \frac{\partial \mathbf{u}}{\partial n_j} + \lambda \mathbf{n}_j \nabla \cdot \mathbf{u} + \mu \mathbf{n}_j \times \nabla \times \mathbf{u}.$$

The operators we chose for the transmission condition  $\mathcal{S}_j$  are two by two matrix valued operators.

Navier equations in frequency domain are very difficult to solve by iterative methods. Their nature is similar to the nature of the Helmholtz equation which are notoriously difficult, see [EG12], and Navier equations have further complications, as we will see.

Our analysis will be based on Fourier transform in the  $y$  direction. Let us denote by  $k \in \mathbb{R}$  the Fourier symbol and  $\hat{\mathbf{u}}(x, k)$  the Fourier transformed solution

$$(2.6) \quad \hat{\mathbf{u}}(x, k) = \mathfrak{F}(\mathbf{u}) = \int_{-\infty}^{\infty} e^{-iky} \mathbf{u}(x, y) dy, \quad \mathbf{u}(x, y) = \mathfrak{F}^{-1}(\hat{\mathbf{u}}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iky} \hat{\mathbf{u}}(x, k) dk.$$

We first investigate if the classical Schwarz algorithm (2.3) is convergent by computing its convergence factor in the Fourier space. Note that all the computations that follow have been partially or entirely performed by using the formal computing tool `Maple`.

Here starts the theorem summarising the results of the section.

**Theorem 2.1** (Convergence analysis of the classical Schwarz algorithm). *(i) For a given initial guess  $(\mathbf{u}_1^0 \in (L^2(\Omega_1)^2), (\mathbf{u}_2^0 \in (L^2(\Omega_2)^2)$ , the classical Schwarz algorithm with overlap has the following convergence factor for each Fourier mode*

$$(2.7) \quad \rho_{cla}(k, \omega, C_p, C_s, \delta) = \max\{|r_+|, |r_-|\},$$

where

$$(2.8) \quad r_{\pm} = \frac{X^2}{2} + e^{-\delta(\lambda_1 + \lambda_2)} \pm \frac{1}{2} \sqrt{X^2 (X^2 + 4e^{-\delta(\lambda_1 + \lambda_2)})}, \quad X = \frac{k^2 + \lambda_1 \lambda_2}{k^2 - \lambda_1 \lambda_2} (e^{-\lambda_1 \delta} - e^{-\lambda_2 \delta}).$$

Here,  $\lambda_{1,2} \in \mathbb{C}$  and are the roots of the characteristic equation of the Fourier transformed Navier equations

$$(2.9) \quad \lambda_1 = \sqrt{k^2 - \frac{\omega^2}{C_s^2}}, \quad \lambda_2 = \sqrt{k^2 - \frac{\omega^2}{C_p^2}}.$$

*(ii) The convergence factor of the overlapping classical Schwarz method (2.3) applied to the Navier equations (2.1) verifies the following*

$$\rho_{cla}(k, \omega, C_p, C_s, \delta) \begin{cases} = 1, & k \in [0, \frac{\omega}{C_p}] \cup \left\{ \frac{\omega}{C_s} \right\}, \\ > 1, & k \in \left( \frac{\omega}{C_p}, \frac{\omega}{C_s} \right), \\ < 1, & k \in \left( \frac{\omega}{C_s}, \infty \right), \end{cases}$$

*if the overlap is small enough. Therefore the algorithm is convergent for higher frequencies while being divergent for all the others.*

*(iii) The maximum of the convergence factor of the classical Schwarz method (2.3) applied to the Navier equations (6) behaves for small overlap  $\delta$  asymptotically as follows*

$$\max_k (\max |r_{\pm}|) = 1 + \frac{\sqrt{2} C_s \omega \left( 3C_p^2 - \sqrt{C_p^4 + 8C_s^4} \right) \sqrt{C_p^2 \sqrt{C_p^4 + 8C_s^4} - C_p^4 - 2C_s^4}}{C_p (C_p^2 + C_s^2)^{\frac{3}{2}} \left( \sqrt{C_p^4 + 8C_s^4} - C_p^2 \right)} \delta.$$

*Proof.* (i) By linearity it suffices to consider only the case  $\mathbf{f} = 0$  and analyse the convergence to the zero solution (superposition principle for linear equations).

After a Fourier transform with respect to  $y$  direction, (2.1) becomes

$$(2.10) \quad \begin{cases} [(\lambda + 2\mu) \partial_x^2 + (\rho\omega^2 - \mu k^2)] \hat{u}_x + ik(\mu + \lambda) \partial_x \hat{u}_z = 0 \\ [\mu \partial_x^2 + (\rho\omega^2 - (\lambda + 2\mu) k^2)] \hat{u}_z + ik(\mu + \lambda) \partial_x \hat{u}_x = 0 \end{cases}$$

This is an ODE system whose solution is obtained after computing the roots  $r$  of its characteristic equation

$$(2.11) \quad \begin{bmatrix} (\lambda + 2\mu)r^2 + \rho\omega^2 - \mu k^2 & ik(\mu + \lambda)r \\ ik(\mu + \lambda)r & \mu r^2 + \rho\omega^2 - (\lambda + 2\mu)k^2 \end{bmatrix} \begin{bmatrix} \hat{u}_x \\ \hat{u}_z \end{bmatrix} = 0.$$

A simple computation shows that these roots are  $\pm\lambda_1$  and  $\pm\lambda_2$  where  $\lambda_{1,2}$  are given by (2.9). Therefore the general form of the solution can be written as:

$$(2.12) \quad \hat{\mathbf{u}}(x, k) = \alpha_1 \mathbf{v}_+ e^{\lambda_1 x} + \beta_1 \mathbf{v}_- e^{-\lambda_1 x} + \alpha_2 \mathbf{w}_+ e^{\lambda_2 x} + \beta_2 \mathbf{w}_- e^{-\lambda_2 x},$$

where  $\mathbf{v}_\pm$  and  $\mathbf{w}_\pm$  are obtained by successively replacing these roots into (2.11)

$$(2.13) \quad \mathbf{v}_+ = \begin{pmatrix} 1 \\ \frac{i\lambda_1}{k} \end{pmatrix}, \quad \mathbf{v}_- = \begin{pmatrix} 1 \\ -\frac{i\lambda_1}{k} \end{pmatrix}, \quad \mathbf{w}_+ = \begin{pmatrix} -\frac{i\lambda_2}{k} \\ 1 \end{pmatrix}, \quad \mathbf{w}_- = \begin{pmatrix} \frac{i\lambda_2}{k} \\ 1 \end{pmatrix}.$$

Coefficients  $\alpha_{1,2}$  and  $\beta_{1,2}$  are uniquely determined by the transmission conditions and  $\lambda_{1,2}$  are defined in (2.9). Because the local solutions are vanishing at infinity, subdomain solutions in the Fourier transformed domain are

$$(2.14) \quad \hat{\mathbf{u}}_1(x, k) = \alpha_1 \mathbf{v}_+ e^{\lambda_1 x} + \alpha_2 \mathbf{w}_+ e^{\lambda_2 x}, \quad \hat{\mathbf{u}}_2(x, k) = \beta_1 \mathbf{v}_- e^{-\lambda_1 x} + \beta_2 \mathbf{w}_- e^{-\lambda_2 x}.$$

Before using the iteration we will rewrite the local solutions at iteration  $n$  as

$$(2.15) \quad \begin{aligned} \hat{\mathbf{u}}_1^n &= \alpha_1^n \mathbf{v}_+ e^{\lambda_1 x} + \alpha_2^n \mathbf{w}_+ e^{\lambda_2 x} = \begin{bmatrix} e^{\lambda_1 x} & -\frac{i\lambda_2}{k} e^{\lambda_2 x} \\ \frac{i\lambda_1}{k} e^{\lambda_1 x} & e^{\lambda_2 x} \end{bmatrix} \begin{pmatrix} \alpha_1^n \\ \alpha_2^n \end{pmatrix} =: M_x \boldsymbol{\alpha}^n, \\ \hat{\mathbf{u}}_2^n &= \beta_1^n \mathbf{v}_- e^{-\lambda_1 x} + \beta_2^n \mathbf{w}_- e^{-\lambda_2 x} = \begin{bmatrix} e^{-\lambda_1 x} & \frac{i\lambda_2}{k} e^{-\lambda_2 x} \\ -\frac{i\lambda_1}{k} e^{-\lambda_1 x} & e^{-\lambda_2 x} \end{bmatrix} \begin{pmatrix} \beta_1^n \\ \beta_2^n \end{pmatrix} =: N_x \boldsymbol{\beta}^n, \end{aligned}$$

then we plug (2.15) into the interface iterations of (2.3)

$$(2.16) \quad \begin{aligned} M_\delta \boldsymbol{\alpha}^n &= N_\delta \boldsymbol{\beta}^{n-1} \Leftrightarrow \boldsymbol{\alpha}^n = M_\delta^{-1} N_\delta \boldsymbol{\beta}^{n-1}, \\ N_0 \boldsymbol{\beta}^n &= M_0 \boldsymbol{\alpha}^{n-1} \Leftrightarrow \boldsymbol{\beta}^n = N_0^{-1} M_0 \boldsymbol{\alpha}^{n-1}, \end{aligned}$$

which leads to

$$(2.17) \quad \begin{aligned} \boldsymbol{\alpha}^{n+1} &= (M_\delta^{-1} N_\delta N_0^{-1} M_0) \boldsymbol{\alpha}^{n-1} =: R_\delta^1 \boldsymbol{\alpha}^{n-1}, \\ \boldsymbol{\beta}^{n+1} &= (N_0^{-1} M_0 M_\delta^{-1} N_\delta) \boldsymbol{\beta}^{n-1} =: R_\delta^2 \boldsymbol{\beta}^{n-1}. \end{aligned}$$

where  $R_\delta^{1,2}$  are the iteration matrices which are spectrally equivalent.  $R_\delta^1$  is given by

$$(2.18) \quad R_\delta^1 = \begin{bmatrix} e^{-\delta(\lambda_1+\lambda_2)} X_2^2 \frac{\lambda_1}{\lambda_2} + e^{-2\lambda_1\delta} X_1^2 & X_1 X_2 \left( e^{-2\lambda_1\delta} - e^{-\delta(\lambda_1+\lambda_2)} \right) \\ X_1 X_2 \frac{\lambda_1}{\lambda_2} \left( e^{-\delta(\lambda_1+\lambda_2)} - e^{-2\lambda_2\delta} \right) & e^{-\delta(\lambda_1+\lambda_2)} X_2^2 \frac{\lambda_1}{\lambda_2} + e^{-2\lambda_2\delta} X_1^2 \end{bmatrix},$$

where

$$(2.19) \quad X_1 = \frac{k^2 + \lambda_1 \lambda_2}{k^2 - \lambda_1 \lambda_2}, \quad X_2 = -i \frac{2k \lambda_2}{k^2 - \lambda_1 \lambda_2}.$$

After some computations, we get the eigenvalues  $(r_+, r_-)$  of  $R_\delta^1$

$$(2.20) \quad r_\pm = \frac{X^2}{2} + e^{-\delta(\lambda_1+\lambda_2)} \pm \frac{1}{2} \sqrt{X^2 (X^2 + 4e^{-\delta(\lambda_1+\lambda_2)})}, \quad X = \frac{k^2 + \lambda_1 \lambda_2}{k^2 - \lambda_1 \lambda_2} \left( e^{-\lambda_1\delta} - e^{-\lambda_2\delta} \right),$$

The convergence factor of the classical Schwarz algorithm is given by the spectral radius of its iteration matrix  $R_\delta^1$  (or  $R_\delta^2$ ). This definition will be used for the following results and can be extended to optimised algorithms studied in Chapter 3. Therefore in this case, we get

$$(2.21) \quad \rho_{cla}(k, \omega, C_p, C_s, \delta) = \rho(R_\delta^1) = \max\{|r_+|, |r_-|\}.$$

We notice that in the case without overlap ( $\delta = 0$ ), (2.20) gives  $r_\pm = 1$  ( $R_\delta^1 = \text{Id}$ ). Therefore, the Schwarz algorithm is not convergent for all frequencies. This conclusion is consistent with what we know for Helmholtz equations where the non-overlapping algorithm is not convergent for all modes when Dirichlet interface conditions are used. Before analysing the overlapping case, in order to gain more insight, we start with a numerical experiment illustrated in Figure 2.1.

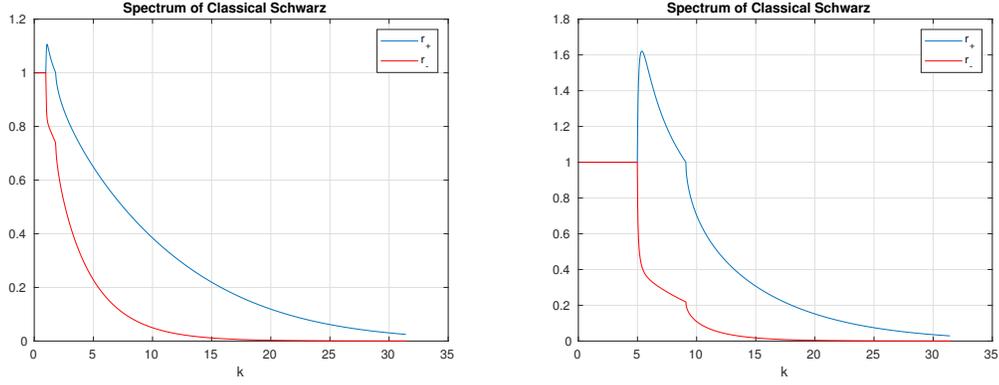


Figure 2.1: Modulus of the eigenvalues of the iteration matrix for the classical Schwarz method with  $C_p = 1$ ,  $C_s = 0.5$ ,  $\delta = 0.1$ . Left: for  $\omega = 1$ . Right: for  $\omega = 5$ .

- (ii) The proof is quite technical. To simplify the notation, we define for the case when the roots  $\lambda_{1,2}$  in (2.9) are complex the quantities

$$(2.22) \quad i\bar{\lambda}_1 := \lambda_1 = i\sqrt{\frac{\omega^2}{C_s^2} - k^2}, \quad i\bar{\lambda}_2 := \lambda_2 = i\sqrt{\frac{\omega^2}{C_p^2} - k^2}.$$

We have to treat five cases: three intervals for  $k$ , and two values  $k \in \{\frac{\omega}{C_p}, \frac{\omega}{C_s}\}$  separating the intervals: in the first interval  $k \in (0, \frac{\omega}{C_p})$ ,  $\lambda_{1,2} \in i\mathbb{R}_+$ , and the eigenvalues (2.8) become

$$r_{\pm} = \frac{X^2}{2} + e^{-i\delta(\bar{\lambda}_1 + \bar{\lambda}_2)} \pm \frac{1}{2} \sqrt{X^2 \left( X^2 + 4e^{-i\delta(\bar{\lambda}_1 + \bar{\lambda}_2)} \right)}, \quad X = \frac{k^2 - \bar{\lambda}_1 \bar{\lambda}_2}{k^2 + \lambda_1 \lambda_2} \left( e^{-i\bar{\lambda}_1 \delta} - e^{-i\bar{\lambda}_2 \delta} \right).$$

The square of their modulus is given by

$$(2.23) \quad |r_{\pm}|^2 = 1 + \underbrace{\frac{\sqrt{A^2 + B^2} + (x^2 + y^2)^2}{4} + e_r (x^2 - y^2) + 2xye_i}_{\text{Part}_1} \pm \frac{\sqrt{2}}{2} \times \left( \underbrace{\frac{x^2 - y^2 + 2e_r}{2} \left( \sqrt{A^2 + B^2} + A \right)^{\frac{1}{2}} + \text{csgn}(B - iA) (xy + e_i) \left( \sqrt{A^2 + B^2} - A \right)^{\frac{1}{2}}}_{\text{Part}_2} \right),$$

where the complex sign is defined as

$$(2.24) \quad \text{csgn}(x) = \begin{cases} 1 & 0 < \Re(x) \quad \text{or} \quad \Re(x) = 0 \ \& \ 0 < \Im(x), \\ -1 & \Re(x) > 0 \quad \text{or} \quad \Re(x) = 0 \ \& \ \Im(x) > 0, \end{cases}$$

and we introduced the quantities  $e_r$ ,  $e_i$ ,  $x$  and  $y$  such as

$$\begin{aligned} e_r &:= -\sin(\delta(\bar{\lambda}_1 + \bar{\lambda}_2)), \\ e_i &:= \cos(\delta(\bar{\lambda}_1 + \bar{\lambda}_2)), \\ x &:= \Re(X) = \frac{k^2 - \bar{\lambda}_1 \bar{\lambda}_2}{k^2 + \bar{\lambda}_1 \bar{\lambda}_2} (\cos(\bar{\lambda}_1 \delta) - \cos(\bar{\lambda}_2 \delta)), \\ y &:= \Im(X) = -\frac{k^2 - \bar{\lambda}_1 \bar{\lambda}_2}{k^2 + \bar{\lambda}_1 \bar{\lambda}_2} (\sin(\bar{\lambda}_1 \delta) - \sin(\bar{\lambda}_2 \delta)). \end{aligned}$$

The terms  $A$  and  $B$  appearing in the square root are real and defined by  $A + iB := X^2 \left( X^2 + 4e^{-i\delta(\bar{\lambda}_1 + \bar{\lambda}_2)} \right)$ , which gives after some computations

$$\begin{aligned} A &= (x^2 - y^2)^2 - 4x^2 y^2 - 8e_i x y + 4e_r (x^2 - y^2), \\ B &= 4(xy + e_i)(x^2 - y^2) + 8e_r x y. \end{aligned}$$

Then we obtain by a direct computation that

$$\begin{aligned} \sqrt{A^2 + B^2} &= (x^2 + y^2) \sqrt{(x^2 + y^2)^2 + 8e_r (x^2 - y^2) + 16e_i x y + 16} \\ &= \frac{16(k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^2 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2))}{(k^2 + \bar{\lambda}_1 \bar{\lambda}_2)^2} \\ &\quad \times \left( 1 - \frac{2(k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^2 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2))}{(k^2 + \bar{\lambda}_1 \bar{\lambda}_2)^2} + \frac{(k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^4 \sin^4(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2))}{(k^2 + \bar{\lambda}_1 \bar{\lambda}_2)^4} \right) \\ &= \frac{16 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2)) \left( (k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^2 \cos^2(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2)) + 4\bar{\lambda}_1 k^2 \bar{\lambda}_2 \right)}{(k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^{-2} (k^2 + \bar{\lambda}_1 \bar{\lambda}_2)^4}, \end{aligned}$$

and

$$\begin{aligned} x^2 - y^2 &= -\frac{4(k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^2 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2)) \cos(\delta(\bar{\lambda}_1 + \bar{\lambda}_2))}{(k^2 + \bar{\lambda}_1 \bar{\lambda}_2)^2}, \\ x^2 + y^2 &= \frac{4(k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^2 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2))}{(k^2 + \bar{\lambda}_1 \bar{\lambda}_2)^2}, \\ xy &= \frac{2(k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^2 \sin(\delta(\bar{\lambda}_1 + \bar{\lambda}_2)) \sin^2(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2))}{(k^2 + \bar{\lambda}_1 \bar{\lambda}_2)^2}. \end{aligned}$$

We now show that Part<sub>1</sub> in (2.23) vanishes identically;

We get on the one hand

$$(2.25) \quad \frac{(x^2 + y^2)^2}{4} = 4 \frac{(k^2 - \bar{\lambda}_1 \bar{\lambda}_2)^4 \sin^4(\frac{\delta}{2}(\bar{\lambda}_1 - \bar{\lambda}_2))}{(k^2 + \bar{\lambda}_1 \bar{\lambda}_2)^4},$$

and on the other hand, we have

$$(2.26) \quad \begin{aligned} \frac{\sqrt{A^2+B^2}}{4} &= \frac{(k^2-\bar{\lambda}_1\bar{\lambda}_2)^4 \sin^2(\delta(\bar{\lambda}_1-\bar{\lambda}_2))}{(k^2+\bar{\lambda}_1\bar{\lambda}_2)^4} + \frac{16 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1-\bar{\lambda}_2))\bar{\lambda}_1 k^2 \bar{\lambda}_2}{(k^2-\bar{\lambda}_1\bar{\lambda}_2)^{-2}(k^2+\bar{\lambda}_1\bar{\lambda}_2)^4}, \\ e_r(x^2-y^2) &= -\frac{4(k^2-\bar{\lambda}_1\bar{\lambda}_2)^2 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1-\bar{\lambda}_2)) \cos^2(\delta(\bar{\lambda}_1+\bar{\lambda}_2))}{(k^2+\bar{\lambda}_1\bar{\lambda}_2)^2}, \\ 2e_i xy &= -\frac{4(k^2-\bar{\lambda}_1\bar{\lambda}_2)^2 \sin^2(\delta(\bar{\lambda}_1+\bar{\lambda}_2)) \sin^2(\frac{\delta}{2}(\bar{\lambda}_1-\bar{\lambda}_2))}{(k^2+\bar{\lambda}_1\bar{\lambda}_2)^2}, \end{aligned}$$

and we obtain by adding the three terms from (2.26) to each other

$$(2.27) \quad -\frac{4(k^2-\bar{\lambda}_1\bar{\lambda}_2)^4 \sin^4(\frac{\delta}{2}(\bar{\lambda}_1-\bar{\lambda}_2))}{(k^2+\bar{\lambda}_1\bar{\lambda}_2)^4}.$$

This leads, by adding (2.25) and (2.27) indeed to  $\text{Part}_1 \equiv 0$ . We next show that also  $\text{Part}_2$  in (2.23) vanishes identically: we get

$$\begin{aligned} \frac{x^2-y^2}{2} + e_r &= \cos(\delta(\bar{\lambda}_1+\bar{\lambda}_2)) \left( 1 - 2 \frac{(k^2-\bar{\lambda}_1\bar{\lambda}_2)^2 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1-\bar{\lambda}_2))}{(k^2+\bar{\lambda}_1\bar{\lambda}_2)^2} \right), \\ xy + e_i &= -\sin(\delta(\bar{\lambda}_1+\bar{\lambda}_2)) \left( 1 - 2 \frac{(k^2-\bar{\lambda}_1\bar{\lambda}_2)^2 \sin^2(\frac{\delta}{2}(\bar{\lambda}_1-\bar{\lambda}_2))}{(k^2+\bar{\lambda}_1\bar{\lambda}_2)^2} \right), \end{aligned}$$

and for the term involving  $A$  and  $B$

$$\begin{aligned} \sqrt{\sqrt{A^2+B^2} \pm A} &= 4 \frac{k^2-\bar{\lambda}_1\bar{\lambda}_2}{(k^2+\bar{\lambda}_1\bar{\lambda}_2)^2} \sin(\frac{\delta}{2}(\bar{\lambda}_1-\bar{\lambda}_2)) \sqrt{1 \mp \cos(2\delta(\bar{\lambda}_1+\bar{\lambda}_2))} \\ &\quad \times \sqrt{(k^2-\bar{\lambda}_1\bar{\lambda}_2)^2 \cos^2(\frac{\delta}{2}(\bar{\lambda}_1-\bar{\lambda}_2)) + 4k^2\bar{\lambda}_1\bar{\lambda}_2}. \end{aligned}$$

By analyzing the signs of the different terms, we obtain for the complex sign

$$\text{csgn}(B-iA) = \text{sg}(\cos(\delta(\bar{\lambda}_1+\bar{\lambda}_2)) \sin(\delta(\bar{\lambda}_1+\bar{\lambda}_2))),$$

and after a lengthy computation we obtain

$$\begin{aligned} \text{Part}_2 &= C_k \times \left( \sqrt{1 + \cos(2\delta(\bar{\lambda}_1+\bar{\lambda}_2))} \sin(\delta(\bar{\lambda}_1+\bar{\lambda}_2)) \right. \\ &\quad \left. - \text{csgn}(B-iA) \cos(\delta(\bar{\lambda}_1+\bar{\lambda}_2)) \sqrt{1 - \cos(2\delta(\bar{\lambda}_1+\bar{\lambda}_2))} \right), \end{aligned}$$

where  $C_k \in \mathbb{R}^*$  ( $= \mathbb{R} \setminus \{0\}$ ) is a complicated factor depending on  $k$ . A direct computation for the second factor of  $\text{Part}_2$  shows that independently of the value

of  $\text{csgn}(B - iA)$ , we get  $\text{Part}_2 \equiv 0$ . We can thus conclude from (2.23) that

$$\rho_{cla}(k, \omega, C_p, C_s, \delta) = \max\{|r_+|, |r_-|\} = |r_+| = |r_-| = 1$$

and therefore the algorithm stagnates in the first interval  $k \in [0, \frac{\omega}{C_p})$ , see the first interval in Figure 2.1. At the boundary between the first and second interval, where  $k = \frac{\omega}{C_p}$ , we have that  $\lambda_2 = 0$  and  $\lambda_1 \in i\mathbb{R}_+^*$ , and therefore (2.8) becomes

$$r_{\pm} = \frac{1}{2}(1 + e^{-2i\bar{\lambda}_1\delta}) \pm \frac{1}{2}\sqrt{(1 - e^{-2i\bar{\lambda}_1\delta})^2}, \quad X = e^{-i\bar{\lambda}_1\delta} - 1,$$

and  $\Re(1 - e^{-2i\bar{\lambda}_1\delta}) = 1 - \cos(2\bar{\lambda}_1\delta)$  being positive we have equivalently

$$r_+ = 1, \quad r_- = e^{-2i\bar{\lambda}_1\delta} \quad \implies \quad \rho_{cla}\left(\frac{\omega}{C_p}, \omega, C_p, C_s, \delta\right) = \max\{|r_+|, |r_-|\} = 1,$$

and hence the algorithm stagnates also when the first interval is closed on the right, i.e. for  $k \in [0, \frac{\omega}{C_p}]$ . In the second interval,  $k \in (\frac{\omega}{C_p}, \frac{\omega}{C_s})$ , we have that  $\lambda_1 \in i\mathbb{R}_+^*$  and  $\lambda_2 \in \mathbb{R}_+^*$ , and hence (2.8) becomes

$$r_{\pm} = \frac{X^2}{2} + e^{-\delta(i\bar{\lambda}_1 + \lambda_2)} \pm \frac{1}{2}\sqrt{X^2(X^2 + 4e^{-\delta(i\bar{\lambda}_1 + \lambda_2)})}, \quad X = \frac{k^2 + i\bar{\lambda}_1\lambda_2}{k^2 - i\bar{\lambda}_1\lambda_2}(e^{-i\bar{\lambda}_1\delta} - e^{-\lambda_2\delta}).$$

We compute the modulus of the eigenvalues and expand them for overlap parameter  $\delta$  small to find

$$(2.28) \quad |r_+| = 1 + \underbrace{\frac{2\omega^2\lambda_2\bar{\lambda}_1^2}{C_p^2(k^4 + \bar{\lambda}_1^2\lambda_2^2)}}_{C_2(k)}\delta + \mathcal{O}(\delta^2), \quad |r_-| = 1 - \frac{2\omega^2\lambda_2k^2}{C_s^2(k^4 + \bar{\lambda}_1^2\lambda_2^2)}\delta + \mathcal{O}(\delta^2).$$

We thus obtain that  $\rho_{cla}(k, \omega, C_p, C_s, \delta) = \max\{|r_+|, |r_-|\}$  is bigger than one for  $\delta$  small and the method diverges, see the middle interval in Figure 2.1<sup>1</sup>. Between the second and third interval, where  $k = \frac{\omega}{C_s}$ , we have that  $\lambda_1 = 0$  and  $\lambda_2 = \frac{\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p} > 0$ , and hence (2.8) becomes

$$r_{\pm} = \frac{1}{2}\left(1 + e^{-2\lambda_2\delta}\right) \pm \frac{1}{2}\sqrt{(1 - e^{-2\lambda_2\delta})^2}.$$

We thus obtain

$$r_+ = 1, \quad r_- = e^{-2\lambda_2\delta} \quad \implies \quad \rho_{cla}\left(\frac{\omega}{C_s}, \omega, C_p, C_s, \delta\right) = \max\{|r_+|, |r_-|\} = 1,$$

---

<sup>1</sup>Numerically we observe that also for a large overlap, the algorithm diverges, see Figure 2.1, but this seems to be difficult to prove.

and the algorithm stagnates for  $k = \frac{\omega}{C_s}$ . In the last interval,  $k \in (\frac{\omega}{C_s}, \infty)$ ,  $\lambda_{1,2} \in \mathbb{R}_+^*$  and by expanding  $r_{\pm} > 0$  from (2.8) for  $\delta$  small, we get

$$r_+ = 1 - \frac{2\lambda_2\omega^2}{C_s^2(k^2 - \lambda_1\lambda_2)}\delta + \mathcal{O}(\delta^2) < 1, \quad r_- = 1 - \frac{2\lambda_1\omega^2}{C_p^2(k^2 - \lambda_1\lambda_2)}\delta + \mathcal{O}(\delta^2) < 1,$$

since  $k^2 - \lambda_1\lambda_2 > 0$ . We can thus conclude that

$$\rho_{cla}(k, \omega, C_p, C_s, \delta) = \max\{|r_+|, |r_-|\} < 1,$$

see the last interval in Figure 2.1, where we also see that  $\lim_{k \rightarrow \infty} r_{\pm} = 0$ , since all the real exponentials involved in the expressions of  $r_{\pm}$  are decreasing to 0 as  $k$  increases.

- (iii) The maximum of the convergence factor is attained in the middle interval where the algorithm is divergent and this quantity is larger than one. In this case for a fixed  $k$ , the convergence factor is given by (2.28) with  $C_2(k)$  being the positive quantity in front of  $\delta$ . Computing the maximum of (2.28) is equivalent to the computation of the maximum of  $C_2(k)$ . By taking the derivative w.r.t  $k$  we get

$$\frac{dC_2(k)}{dk} = \frac{-2k(k^4 C_s^2 (C_p^2 + C_s^2) + k^2 \omega^2 (C_p^2 - 2C_s^2) - \omega^4)}{(k^2 (C_p^2 + C_s^2) - \omega^2)^2 \sqrt{k^2 - \frac{\omega^2}{C_p^2}}}.$$

We solve the equation  $\frac{dC_2(k)}{dk} = 0$  w.r.t.  $k$  and we obtain three real solutions

$$\left[ 0, \frac{\sqrt{2}\omega}{2C_s} \sqrt{\frac{2C_s^2 - C_p^2 + \sqrt{C_p^4 + 8C_s^4}}{C_p^2 + C_s^2}}, -\frac{\sqrt{2}\omega}{2C_s} \sqrt{\frac{2C_s^2 - C_p^2 + \sqrt{C_p^4 + 8C_s^4}}{C_p^2 + C_s^2}} \right],$$

and two purely imaginary ones,

$$\left[ \frac{\sqrt{2}\omega}{2C_s} \sqrt{\frac{2C_s^2 - C_p^2 - \sqrt{C_p^4 + 8C_s^4}}{C_p^2 + C_s^2}}, -\frac{\sqrt{2}\omega}{2C_s} \sqrt{\frac{2C_s^2 - C_p^2 - \sqrt{C_p^4 + 8C_s^4}}{C_p^2 + C_s^2}} \right].$$

We are looking for a positive real local maximum, therefore the good candidate is

$$k_s = \frac{\sqrt{2}\omega}{2C_s} \sqrt{\frac{2C_s^2 - C_p^2 + \sqrt{C_p^4 + 8C_s^4}}{C_p^2 + C_s^2}} \in \mathbb{R}_+^*.$$

By computing the second order derivative at our critical point we get

$$\begin{aligned} \frac{d^2 C}{dk^2}(k_s) &= \frac{32C_s^3 C_p \sqrt{2} (C_p^4 + 8C_s^4) \left(\sqrt{C_p^4 + 8C_s^4} - C_p^2\right)^{-3}}{\omega \left(C_p^2 \sqrt{C_p^4 + 8C_s^4} - C_p^4 - 2C_s^4\right)^{\frac{3}{2}} \sqrt{C_p^2 + C_s^2}} \\ &\times \left( \frac{C_p^8 - C_p^6 C_s^2 + 7C_p^4 C_s^4 - 5C_p^2 C_s^6 + 4C_s^8}{\sqrt{C_p^4 + 8C_s^4}} - (C_p^6 - C_p^4 C_s^2 + 3C_p^2 C_s^4 - C_s^6) \right) < 0 \end{aligned}$$

and we can conclude it is a local maximum.

In conclusion the maximum value of the convergence factor for a small  $\delta$  is:

$$\begin{aligned} \max_k(\rho) &= 1 + C(k_s)\delta \\ &= 1 + \frac{\sqrt{2}C_s\omega \left(3C_p^2 - \sqrt{C_p^4 + 8C_s^4}\right) \sqrt{C_p^2 \sqrt{C_p^4 + 8C_s^4} - C_p^4 - 2C_s^4}}{C_p(C_p^2 + C_s^2)^{\frac{3}{2}} \left(\sqrt{C_p^4 + 8C_s^4} - C_p^2\right)} \delta. \end{aligned}$$

□

## Numerical experiments

We illustrate this divergence of the iterative version of the Schwarz algorithm which can be written as (1.9)

$$\mathbf{U}^{n+1} = \mathbf{U}^n + M^{-1}(\mathbf{F} - \mathbf{A}\mathbf{U}^n),$$

in a numerical experiment (Figure 2.2) in which we choose the same parameters  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\rho = 1$  and overlap  $\delta = \frac{1}{10}$  as in Figure 2.1. We discretise the time-harmonic Navier equations using P1 finite elements on the domain  $\Omega = (-1, 1) \times (0, 1)$  and decompose it into two overlapping subdomains  $\Omega_1 = (-1, 2h) \times (0, 1)$  and  $\Omega_2 = (-2h, 1) \times (0, 1)$  with  $h = \frac{1}{40}$ , in other words we build a uniform decomposition with the overlapping parameter  $\delta = 4h$ .

We show in Figure 2.2 the error in modulus at iteration 25 of the classical Schwarz method, on the left for  $\omega = 1$  and on the right for  $\omega = 5$ .

The error is computed with respect to the solution of the algebraic system obtain on the global domain  $\mathbf{A}\mathbf{U} = \mathbf{F}$  by a direct method. Note that in more realistic test cases

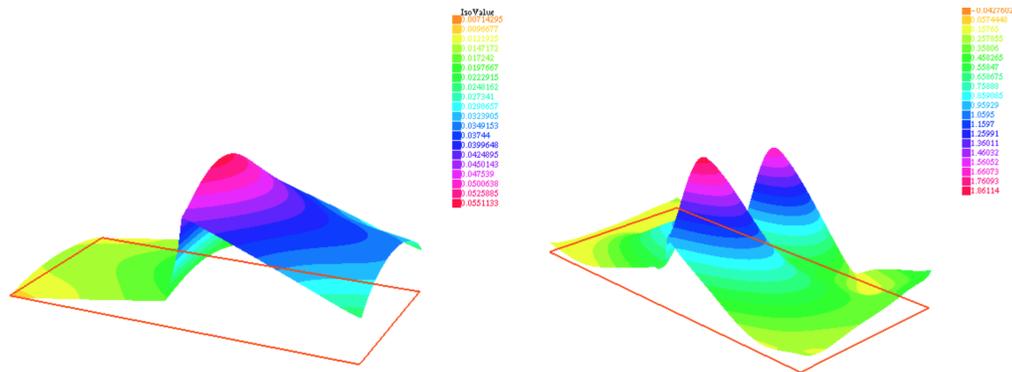


Figure 2.2: Error in modulus at iteration 25 of the classical Schwarz method with 2 subdomains, where one can clearly identify the dominant mode in the error: Left:  $\omega = 1$ . Right:  $\omega = 5$ .

this reference solution is not available. In those cases the global relative residual will be a measure of the convergence.

In the first case ( $\omega = 1$ ) since the lowest frequency that can be represented on the mesh is  $k = \pi$  this would leave outside the interval of frequencies on which the method is divergent, that is  $\left[\frac{\omega}{C_p}, \frac{\omega}{C_s}\right] = [1, 2]$ , which means the method will converge very slowly to a solution. The dominant mode of the error is the lowest frequency that can be represented on the mesh, that is  $|\sin(ky)|$  with  $k = \pi$ . We can notice that the error decreases from  $7.89e - 1$  to  $5e - 2$  after 25 iterations.

In contrast, for  $\omega = 5$  the method is diverging, since the interval of frequencies on which the method is divergent, is given by  $\left[\frac{\omega}{C_p}, \frac{\omega}{C_s}\right] = [5, 10]$ . We notice that the error after a certain number of iterations is increasing and will be  $6.5e - 1$  after 25 iterations and the diverging mode in Figure 2.2 on the right has two bumps along the interface, which corresponds well to the mode  $|\sin(ky)|$  along the interface for  $k = 2\pi \approx 6$ . This seems to be the fastest diverging mode that can be already seen from the analysis in Figure 2.1 on the right.

One might wonder if the classical Schwarz method is nevertheless a good preconditioner for a Krylov method, which can happen also for divergent stationary methods, like for example the Additive Schwarz Method applied to the Laplace problem, which is also not convergent as an iterative method [EG03], but useful as a preconditioner. To investigate this, it suffices to plot the spectrum of the identity matrix minus the iteration operator in the complex plane, which corresponds to the preconditioned systems one would like to solve. We see in Figure 2.3 that the part of the spectrum that leads to a contraction factor  $\rho_{cla}$  with modulus bigger than one lies unfortunately close to zero in the complex

plane, and that is where the residual polynomial of the Krylov method must equal one. Therefore we can infer that the classical Schwarz method will also not work well as a preconditioner.

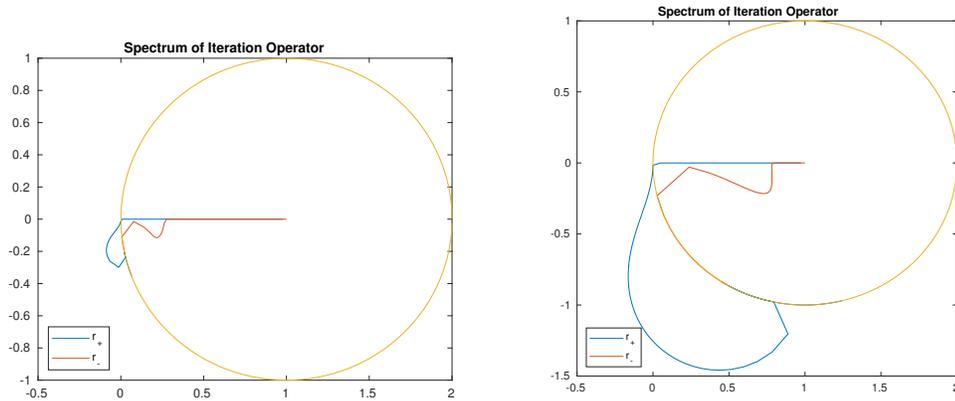


Figure 2.3: Spectrum of the iteration operator for the same example as in Figure 2.1, together with a unit circle centered around the point  $(1, 0)$ . Left:  $\omega = 1$ . Right:  $\omega = 5$

This is also confirmed by the numerical results shown in Figure 2.4,

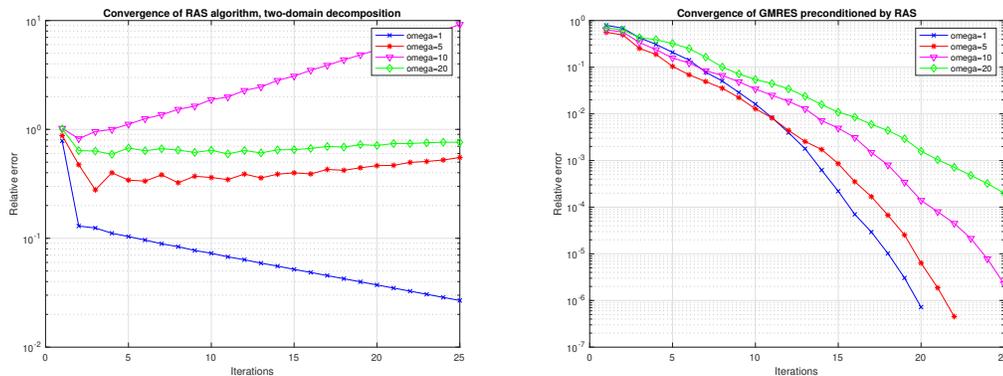


Figure 2.4: Convergence history for RAS and GMRES preconditioned by RAS for different values of  $\omega$

where we used first the classical Schwarz method as a solver and then as preconditioner for GMRES. We see that GMRES now makes the method converge, but convergence depends strongly on  $\omega$  and slows down when  $\omega$  grows. Note that in the case when the Schwarz algorithm is used as a solver the lack of convergence is caused by the singular and diverging modes (this behaviour can be also seen in the case of the Helmholtz

equations) whereas when the Schwarz method is used as a preconditioner, the Krylov method takes care naturally of these diverging modes (and therefore the convergence on the right of Figure 2.4 is not an artefact of the initial guess).

## 2.2 Optimal Schwarz algorithms and local approximations

We just saw that the Schwarz method so far is slowed down by an entire range of low frequency components. For a better performance, we need to improve the algorithm especially in that range and make it the smallest possible.

Let us start by introducing the *optimal Schwarz algorithm* which uses transparent boundary conditions (TBC) as transmission conditions in (2.4) as defined in the following

**Theorem 2.2** (Convergence of the optimal Schwarz algorithm.). *If one chooses in the general Schwarz algorithm (2.4) the operators  $\mathcal{S}_j$  with the Fourier symbols*

$$(2.29) \quad \begin{aligned} \widehat{\mathcal{S}}_1(1,1) &= \rho \frac{\lambda_1 \omega^2}{k^2 - \lambda_1 \lambda_2}, & \widehat{\mathcal{S}}_2(1,1) &= \widehat{\mathcal{S}}_1(1,1), \\ \widehat{\mathcal{S}}_1(1,2) &= +ik\rho \left( 2C_s^2 - \frac{\omega^2}{k^2 - \lambda_1 \lambda_2} \right), & \widehat{\mathcal{S}}_2(1,2) &= -\widehat{\mathcal{S}}_1(1,2), \\ \widehat{\mathcal{S}}_1(2,1) &= -ik\rho \left( 2C_s^2 - \frac{\omega^2}{k^2 - \lambda_1 \lambda_2} \right), & \widehat{\mathcal{S}}_2(2,1) &= -\widehat{\mathcal{S}}_1(2,1), \\ \widehat{\mathcal{S}}_1(2,2) &= \rho \frac{\lambda_2 \omega^2}{k^2 - \lambda_1 \lambda_2}, & \widehat{\mathcal{S}}_2(2,2) &= \widehat{\mathcal{S}}_1(2,2), \end{aligned}$$

where  $\lambda_1, \lambda_2$  are given by (2.9) the resulting algorithm converges in two iterations.

*Proof.* The TBC from (2.4) can be written in Fourier space as

$$\begin{aligned} \mathfrak{F}(\mathcal{T}_{\mathbf{n}_1} + \mathcal{S}_1)(\mathbf{u}) &= \begin{pmatrix} (2\mu + \lambda)\partial_x \hat{u}_1 + i\lambda k \hat{v}_1 \\ i\mu k \hat{u}_1 + \mu \partial_x \hat{v}_1 \end{pmatrix} + \begin{pmatrix} \widehat{\mathcal{S}}_1(1,1)\hat{u}_1 + \widehat{\mathcal{S}}_1(1,2)\hat{v}_1 \\ \widehat{\mathcal{S}}_1(2,1)\hat{u}_1 + \widehat{\mathcal{S}}_1(2,2)\hat{v}_1 \end{pmatrix} \\ &=: \mathbb{T}_{\mathbf{n}_1}(\hat{\mathbf{u}}), \\ \mathfrak{F}(\mathcal{T}_{\mathbf{n}_2} + \mathcal{S}_2)(\mathbf{u}) &= - \begin{pmatrix} (2\mu + \lambda)\partial_x \hat{u}_2 + i\lambda k \hat{v}_2 \\ i\mu k \hat{u}_2 + \mu \partial_x \hat{v}_2 \end{pmatrix} + \begin{pmatrix} \widehat{\mathcal{S}}_2(1,1)\hat{u}_2 + \widehat{\mathcal{S}}_2(1,2)\hat{v}_2 \\ \widehat{\mathcal{S}}_2(2,1)\hat{u}_2 + \widehat{\mathcal{S}}_2(2,2)\hat{v}_2 \end{pmatrix} \\ &=: \mathbb{T}_{\mathbf{n}_2}(\hat{\mathbf{u}}), \end{aligned}$$

where  $\mathbf{n}_1 = (1, 0)$  and  $\mathbf{n}_2 = -\mathbf{n}_1$ . The interface iterations from (2.4) become

$$(2.30) \quad \begin{aligned} \mathbb{T}_{\mathbf{n}_1}(\hat{\mathbf{u}}_1^n)(\delta, \cdot) &= \mathbb{T}_{\mathbf{n}_1}(\hat{\mathbf{u}}_2^{n-1})(\delta, \cdot) \Leftrightarrow A_{1,\delta} \boldsymbol{\alpha}^n = A_{2,\delta} \boldsymbol{\beta}^{n-1}, \\ \mathbb{T}_{\mathbf{n}_2}(\hat{\mathbf{u}}_2^n)(0, \cdot) &= \mathbb{T}_{\mathbf{n}_2}(\hat{\mathbf{u}}_1^{n-1})(0, \cdot) \Leftrightarrow B_2 \boldsymbol{\beta}^n = B_1 \boldsymbol{\alpha}^{n-1}, \end{aligned}$$

where

$$A_{2,\delta} = \begin{bmatrix} \frac{\widehat{\mathcal{S}}_1(1,1) - 2\lambda_1 C_s^2 \rho}{e^{\lambda_1 \delta}} - i \frac{\lambda_1 \widehat{\mathcal{S}}_1(1,2)}{k e^{\lambda_1 \delta}} & \frac{\widehat{\mathcal{S}}_1(1,2)}{e^{\lambda_2 \delta}} - i \frac{2\rho k^2 C_s^2 - \lambda_2 \widehat{\mathcal{S}}_1(1,1) - \rho\omega^2}{k e^{\lambda_2 \delta}} \\ \frac{\widehat{\mathcal{S}}_1(2,1)}{e^{\lambda_1 \delta}} + i \frac{2\rho k^2 C_s^2 - \lambda_1 \widehat{\mathcal{S}}_1(2,2) - \rho\omega^2}{k e^{\lambda_1 \delta}} & \frac{\widehat{\mathcal{S}}_1(2,2) - 2C_s^2 \rho \lambda_2}{e^{\lambda_2 \delta}} + i \frac{\lambda_2 \widehat{\mathcal{S}}_1(2,1)}{k e^{\lambda_2 \delta}} \end{bmatrix}$$

and

$$B_1 = \begin{bmatrix} \widehat{\mathcal{S}}_2(1,1) - 2\lambda_1 \rho C_s^2 + i \frac{\lambda_1 \widehat{\mathcal{S}}_2(1,2)}{k} & \widehat{\mathcal{S}}_2(1,2) + i \frac{2k^2 \rho C_s^2 - \lambda_2 \widehat{\mathcal{S}}_2(1,1) - \rho\omega^2}{k} \\ \widehat{\mathcal{S}}_2(2,1) - i \frac{2k^2 C_s^2 \rho - \lambda_1 \widehat{\mathcal{S}}_2(2,2) - \rho\omega^2}{k} & \widehat{\mathcal{S}}_2(2,2) - 2C_s^2 \rho \lambda_2 - i \frac{\lambda_2 \widehat{\mathcal{S}}_2(2,1)}{k} \end{bmatrix}.$$

Now, to obtain the transparent boundary conditions, it suffices to write the iteration after a Fourier transform in the  $y$  direction and to choose the operators  $\mathcal{S}_j$  such that the right hand side vanishes. We note that if we replace  $(\widehat{\mathcal{S}}_1, \widehat{\mathcal{S}}_2)$  by (2.29), we get

$$A_{2,\delta} = B_1 = 0 \quad \Rightarrow \quad \boldsymbol{\alpha}^n = \boldsymbol{\beta}^n = 0,$$

thus the new convergence factor vanishes identically and the algorithm converges in two iterations independently of any initial guess and overlap  $\delta$ .  $\square$

To use the optimal choice of transmissions operators, we need to back transform the TBC from (2.29) into the physical domain. Unfortunately,  $\mathcal{S}_j$  are non local operators because of the inverse transform with square roots terms at the interfaces [HTJ88] therefore they cannot be used efficiently in practice.

We want now to build local approximations for the optimal transmissions conditions which will lead to a new class of Schwarz algorithms. In order to do this, we can approximate the symbols of the optimal transmission conditions from (2.29) by polynomial symbols in  $ik$  which correspond to local operators.

We will explain in the following how to build these local operators, but to start with, we will evaluate the convergence factor in the case where more general (not necessarily transparent) transmission operators  $\widehat{\mathcal{S}}_{1,2}$  are used.

**Lemma 2.1.** *For a given initial guess  $(\mathbf{u}_1^0 \in (L^2(\Omega_1)^2), (\mathbf{u}_2^0 \in (L^2(\Omega_2)^2)$ , the general Schwarz algorithm with overlap (2.4) has the following convergence factor for each Fourier mode*

$$(2.31) \quad \rho_{opt}(k, \omega, C_p, C_s, \delta) = (\max\{|r_+|, |r_-|\})^{\frac{1}{2}}, \quad r_{\pm} = \frac{X^2}{2} + Y \pm \frac{1}{2} \sqrt{X^2(X^2 + 4Y)},$$

with

$$(2.32) \quad X = e^{-\lambda_1 \delta} b_{11} - e^{-\lambda_2 \delta} b_{22}, \quad Y = \frac{b_{11} b_{22} - b_{12} b_{21}}{e^{\lambda_1 \delta} e^{\lambda_2 \delta}}, \quad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} := B_2^{-1} B_1,$$

where

$$(2.33) \quad B_1 = \begin{bmatrix} \widehat{\mathcal{S}}_2(1,1) - 2\lambda_1 \rho C_s^2 - i \frac{\lambda_1 \widehat{\mathcal{S}}_2(1,2)}{k} & \widehat{\mathcal{S}}_2(1,2) + i \frac{2k^2 \rho C_s^2 - \lambda_2 \widehat{\mathcal{S}}_2(1,1) - \rho \omega^2}{k} \\ \widehat{\mathcal{S}}_2(2,1) - i \frac{2k^2 C_s^2 \rho - \lambda_1 \widehat{\mathcal{S}}_2(2,2) - \rho \omega^2}{k} & \widehat{\mathcal{S}}_2(2,2) + 2C_s^2 \rho \lambda_2 + i \frac{\lambda_2 \widehat{\mathcal{S}}_2(2,1)}{k} \end{bmatrix},$$

$$(2.34) \quad B_2 = \begin{bmatrix} \widehat{\mathcal{S}}_2(1,1) + 2\lambda_1 \rho C_s^2 + i \frac{\lambda_1 \widehat{\mathcal{S}}_2(1,2)}{k} & \widehat{\mathcal{S}}_2(1,2) + i \frac{2k^2 \rho C_s^2 + \lambda_2 \widehat{\mathcal{S}}_2(1,1) - \rho \omega^2}{k} \\ \widehat{\mathcal{S}}_2(2,1) - i \frac{2k^2 C_s^2 \rho + \lambda_1 \widehat{\mathcal{S}}_2(2,2) - \rho \omega^2}{k} & \widehat{\mathcal{S}}_2(2,2) - 2C_s^2 \rho \lambda_2 - i \frac{\lambda_2 \widehat{\mathcal{S}}_2(2,1)}{k} \end{bmatrix},$$

and  $\lambda_{1,2} \in \mathbb{C}$  are given by (2.9).

*Proof.* We use the local solutions in the Fourier space already computed in (2.15), plug them in the new transmission conditions from algorithm(2.4) and write them in the Fourier space like in (2.30). We get the two half-iteration matrices like in (2.16), denoted by  $A, B \in \mathcal{M}(\mathbb{C})$  and that can be computed as follows

$$(2.35) \quad B = B_2^{-1} B_1 =: \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \quad A = A_1^{-1} A_2 = \begin{bmatrix} e^{-2\lambda_1 \delta} b_{11} & -e^{-(\lambda_1 + \lambda_2) \delta} b_{12} \\ -e^{-(\lambda_1 + \lambda_2) \delta} b_{21} & e^{-2\lambda_2 \delta} b_{22} \end{bmatrix},$$

with  $B_1, B_2$  given in (2.33) and (2.34).

This leads to the two spectrally equivalent iteration matrices  $AB$  and  $BA$ , whose common spectral radius gives the square of the convergence factor  $\rho_{opt}(k, \omega, C_p, C_s, \delta)$

$$\boldsymbol{\alpha}^{n+1} = AB\boldsymbol{\alpha}^{n-1}.$$

of the algorithm (2.31) as seen in the previous double iteration.  $\square$

The following corollary will further simplify the computation of the convergence factor

**Remark 2.1.** *For the general Schwarz algorithm applied on the Navier equations, if at least one anti-diagonal term of at least one of the half-iteration matrix is equal to zero, the eigenvalues of the iteration operator are only determined by the diagonal of*

the half-iteration matrices.

Indeed, if we start with the generic form of our two half-iteration matrices

$$(2.36) \quad M_1 = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \quad M_2 = \begin{bmatrix} e^{-2\lambda_1\delta} b_{11} & -e^{-(\lambda_1+\lambda_2)} b_{12} \\ -e^{-(\lambda_1+\lambda_2)} b_{21} & e^{-2\lambda_2\delta} b_{22} \end{bmatrix}, \quad \lambda_1, \lambda_2 \in \mathbb{C},$$

if  $b_{12} = 0$  and/or  $b_{21} = 0$ , a quick computation gives

$$(2.37) \quad r_- = e^{-2\lambda_1\delta} b_{11}^2, \quad r_+ = e^{-2\lambda_2\delta} b_{22}^2.$$

and as a consequence, the convergence factor  $\rho = \max(|r_+|, |r_-|)$ .

## 2.3 Absorbing boundary conditions

The simplest optimised Schwarz algorithm can be obtained by approximating  $\mathcal{S}_j$  in the transmission conditions using a low frequency expansion in the Fourier variable  $k$  of the optimal choice given in Theorem 2.2. This leads to the so called Taylor transmission conditions (TTC)

$$(2.38) \quad \begin{aligned} \widehat{\mathcal{S}}_1(1,1) &= i\rho\omega C_p + i\rho\frac{C_p^2}{2\omega}(C_p - 2C_s)k^2 + \mathcal{O}(k^4), \\ \widehat{\mathcal{S}}_1(1,2) &= -i\rho(C_p - 2C_s)C_s k + \mathcal{O}(k^3), \\ \widehat{\mathcal{S}}_1(2,1) &= i\rho(C_p - 2C_s)C_s k + \mathcal{O}(k^3), \\ \widehat{\mathcal{S}}_1(2,2) &= i\rho\omega C_s + i\rho\frac{C_s^2}{2\omega}(C_s - 2C_p)k^2 + \mathcal{O}(k^4), \end{aligned}$$

and  $\widehat{\mathcal{S}}_2$  with the same relation to  $\widehat{\mathcal{S}}_1$  as for the optimal choice in Theorem 2.2.

A zeroth order approximation would thus be

$$(2.39) \quad \begin{aligned} \widehat{\mathcal{S}}_1^{T_0}(1,1) &= i\rho\omega C_p, \\ \widehat{\mathcal{S}}_1^{T_0}(1,2) &= 0, \\ \widehat{\mathcal{S}}_1^{T_0}(2,1) &= 0, \\ \widehat{\mathcal{S}}_1^{T_0}(2,2) &= i\rho\omega C_s. \end{aligned}$$

which was also obtained as an absorbing boundary condition using a different argument as seen in [HMCK04] or explained in detail in the introduction. These absorbing boundary conditions happen to be exact for a combination of plane waves, therefore

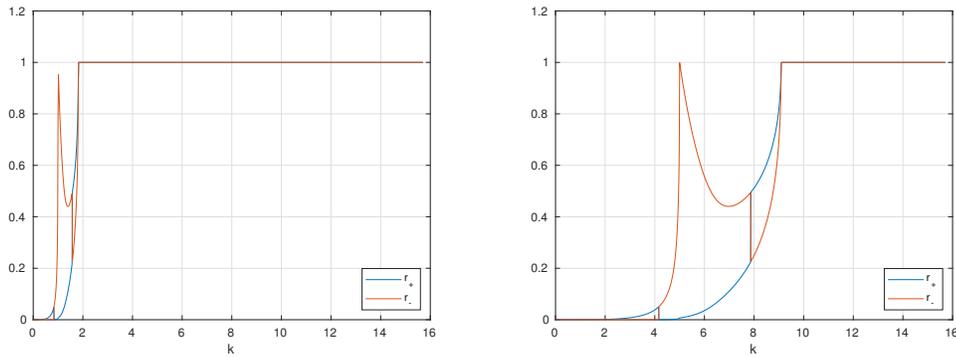


Figure 2.5: Modulus of the eigenvalues of the iteration matrix for the Schwarz method with absorbing TC without overlap and  $C_p = 1$ ,  $C_s = \frac{1}{2}$  and  $\rho = 1$ . Left: ( $\omega = 1$ ). Right: ( $\omega = 5$ ).

they have a physical sense for this particular problem.

We can also introduce an improved zeroth order optimised TTC where we would fix  $k = k_e$  in (2.38) and ignore the higher order terms,

$$\begin{aligned}
 \widehat{\mathcal{S}}_1^{T_{0,e}}(1, 1) &= i\rho\omega C_p + i\rho\frac{C_p^2}{2\omega}(C_p - 2C_s)k_e^2, \\
 \widehat{\mathcal{S}}_1^{T_{0,e}}(1, 2) &= -i\rho(C_p - 2C_s)C_s k_e, \\
 \widehat{\mathcal{S}}_1^{T_{0,e}}(2, 1) &= i\rho(C_p - 2C_s)C_s k_e, \\
 \widehat{\mathcal{S}}_1^{T_{0,e}}(2, 2) &= i\rho\omega C_s + i\rho\frac{C_s^2}{2\omega}(C_s - 2C_p)k_e^2.
 \end{aligned}
 \tag{2.40}$$

There are also more general second order optimised TTC ( $\widehat{\mathcal{S}}_j^{T_2}$ ) such as

$$\begin{aligned}
 \widehat{\mathcal{S}}_1^{T_2}(1, 1) &= i\rho\omega C_p + i\rho\frac{C_p^2}{2\omega}(C_p - 2C_s)k^2, \\
 \widehat{\mathcal{S}}_1^{T_2}(1, 2) &= -i\rho(C_p - 2C_s)C_s k, \\
 \widehat{\mathcal{S}}_1^{T_2}(2, 1) &= i\rho(C_p - 2C_s)C_s k, \\
 \widehat{\mathcal{S}}_1^{T_2}(2, 2) &= i\rho\omega C_s + i\rho\frac{C_s^2}{2\omega}(C_s - 2C_p)k^2.
 \end{aligned}
 \tag{2.41}$$

The use of absorbing (or Robin) boundary conditions for scalar equations as well as for wave type equations as Helmholtz leads to an improvement in the convergence factor and in the latter has also the advantage of making the local problem systematically well

posed. Therefore we expect that their use in a Schwarz algorithm for the time-harmonic elastic waves will also have a positive effect by greatly improving its behaviour.

### Convergence analysis in the non-overlapping case

Optimised Schwarz methods can often be used without overlap, but in the case of the Navier equations the conclusions are different.

- By using just the zeroth order Taylor condition without overlap does also not lead to a convergent algorithm, as one can see in Figure 2.5 where we plotted the eigenvalues in modulus of a full iteration (Note the plots are the same, just the scaling in  $k$  changes).
- For the low frequencies, the algorithm converges however, in contrast to the classical Schwarz method, but for the high frequencies there is stagnation. Note also that the curve for different values of  $\omega$  is the same, only the scaling in  $k$  changes.
- We notice in a similar manner that even the TTC with a general parameter don't help and can even lead to an explosion for low frequencies while not improving high frequencies (see Figure 2.6 left).

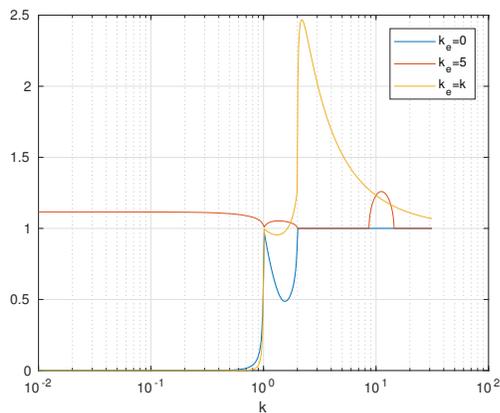


Figure 2.6: Spectrum of the iteration matrix for the Schwarz method with  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\rho = 1$ ,  $\omega = 1$  and  $\delta = 0$ . With absorbing  $\rho_{T_0}$  ( $k_e = 0$ ), general zeroth order  $\rho_{T_{0,e}}$  ( $k_e = 5$ ) and second order  $\rho_{T_2}$  ( $k_e = k$ ) TTC.

These first intuitive conclusions lead us to the following theoretical result

**Theorem 2.3** (Convergence of the non-overlapping Schwarz algorithm with TTC).  
*The Schwarz method with zero order transmission conditions (2.40) for non-overlapping*

decompositions converges for  $k \in \left(0, \frac{\omega}{C_s}\right) \setminus \left\{\frac{\omega}{C_p}\right\}$  and is divergent with the contraction factor being equal to 1 for  $k \in \left(\frac{\omega}{C_s}, \infty\right)$ .

*Proof.* Under the specific hypothesis of the theorem, the half-iteration matrix from formula (2.35) becomes

$$(2.42) \quad B = \frac{1}{D} \begin{bmatrix} -Z_1 - Z_2 - i\omega^3 \left(\lambda_1 - \lambda_2 \frac{C_p}{C_s}\right) & i\lambda_2 K \\ -i\lambda_1 K & -Z_1 - Z_2 + i\omega^3 \left(\lambda_1 - \lambda_2 \frac{C_p}{C_s}\right) \end{bmatrix} =: \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix},$$

where

$$(2.43) \quad Z_1 = C_s^3 (k^2 + \lambda_1^2)^2 + \omega^2 C_p k^2, \quad Z_2 = (4C_s^3 k^2 + C_p \omega^2) \lambda_1 \lambda_2,$$

$$(2.44) \quad K = 2k (C_p \omega^2 + 2C_s^3 (k^2 + \lambda_1^2)), \quad D = -Z_1 + Z_2 + i\omega^3 \left(\lambda_1 + \lambda_2 \frac{C_p}{C_s}\right).$$

The eigenvalues are given by

$$(2.45) \quad r_{\pm} = \frac{X^2}{2} + Y \pm \frac{1}{2} \sqrt{X^2(X^2 + 4Y)}, \quad X = b_{11} - b_{22}, \quad Y = b_{11}b_{22} - b_{12}b_{21}.$$

We define now  $\bar{\lambda}_j \in \mathbb{R}_+$ ,  $j = 1, 2$  as in (2.22). We distinguish the following cases (all the computations have been done using `Maple` and details can be found in the Appendix)

- **Case 1:** If  $k \in \left(0, \frac{\omega}{C_p}\right)$  then  $\lambda_{1,2} \in i\mathbb{R}_+$  and (2.45) gives

$$X = \frac{2\omega^3}{D} \left(\bar{\lambda}_1 - \bar{\lambda}_2 \frac{C_p}{C_s}\right), \quad Y = \frac{1}{D^2} \left((Z_1 + Z_2)^2 - \omega^6 \left(\bar{\lambda}_1 - \bar{\lambda}_2 \frac{C_p}{C_s}\right)^2 + \bar{\lambda}_1 \bar{\lambda}_2 K^2\right).$$

We can see easily that  $X^2 + 2Y > 0$  and  $X^2 + 4Y > 0$  therefore  $r_+ > |r_-| > 0$  so we just need to check the equivalent relations

$$r_+ < 1 \Leftrightarrow (X^2 + 2Y) + \sqrt{X^2(X^2 + 4Y)} < 2 \Leftrightarrow (1 - Y)^2 - X^2 > 0.$$

The latter inequality can be checked by first setting  $X = \tilde{X}/D$  and  $Y = \tilde{Y}/D^2$  which leads to the straightforward condition

$$0 < \left(1 - \tilde{Y}/D^2\right)^2 - \left(\tilde{X}/D\right)^2 \Leftrightarrow 0 < \left(D^2 - \tilde{Y}\right)^2 - D^2 \tilde{X}^2 = 16\omega^6 \frac{C_p}{C_s} \bar{\lambda}_1 \bar{\lambda}_2 C^2$$

where  $C \in \mathbb{R}^*$  is a quantity with a complicated expression depending on  $(C_p, C_s, \omega, k)$ . We conclude that in this case the algorithm is convergent.

- **Case 2:** If  $k = \frac{\omega}{C_p}$  then  $\lambda_1 = i \frac{\omega \sqrt{C_p^2 - C_s^2}}{C_s C_p}$  and  $\lambda_2 = 0$ . Then the coefficients (2.32) of the half-iteration matrix are given by

$$b_{11} = \frac{(C_p + C_s)(C_p^3 - 4C_p C_s^2 + 4C_s^3) - \sqrt{C_p^2 - C_s^2} C_p^3}{(C_p + C_s)(C_p^3 - 4C_p C_s^2 + 4C_s^3) + \sqrt{C_p^2 - C_s^2} C_p^3}, \quad b_{12} = 0, \quad b_{21} \in \mathbb{C}, \quad b_{22} = 1$$

and the eigenvalues  $r_{\pm}$  can be computed from remark (2.1)

$$r_+ = 1, \quad |r_-| = \left| \frac{(C_p + C_s)(C_p^3 - 4C_p C_s^2 + 4C_s^3) - \bar{\lambda}_1 C_p^4 C_s}{(C_p + C_s)(C_p^3 - 4C_p C_s^2 + 4C_s^3) + \bar{\lambda}_1 C_p^4 C_s} \right|^2.$$

Since  $C_p^3 - 4C_p C_s^2 + 4C_s^3 > 0 \Rightarrow |r_-| < 1$  and  $\rho_{T_0} = 1$ .

- **Case 3:** If  $k \in \left(\frac{\omega}{C_p}, \frac{\omega}{C_s}\right)$  then  $\lambda_1 \in i\mathbb{R}_+$  and  $\lambda_2 \in \mathbb{R}_+$ . The expressions of  $r_{\pm}$  become

$$r_{\pm} = \left( \frac{\omega^3 \left( \bar{\lambda}_1 + i\lambda_2 \frac{C_p}{C_s} \right) \pm \sqrt{-i\lambda_2 \bar{\lambda}_1 K^2 - (\bar{Z}_2 - iZ_1)^2}}{(-Z_1 + i\bar{Z}_2) - \omega^3 \left( \bar{\lambda}_1 - i\lambda_2 \frac{C_p}{C_s} \right)} \right)^2.$$

By computing its modulus we get

$$|r_{\pm}| = \frac{\left( \omega^3 \frac{C_p}{C_s} \lambda_2 \mp \operatorname{csgn}(\alpha) \frac{\sqrt{2}}{2} \sqrt{\sqrt{(Z_1^2 - \bar{Z}_2^2)^2 + (K^2 \lambda_2 \bar{\lambda}_1 - 2Z_1 \bar{Z}_2)^2} - Z_1^2 + \bar{Z}_2^2} \right)^2}{\left( \omega^3 \frac{C_p}{C_s} \lambda_2 + \bar{Z}_2 \right)^2 + (\omega^3 \bar{\lambda}_1 + Z_1)^2} + \frac{\left( \omega^3 \bar{\lambda}_1 \pm \frac{\sqrt{2}}{2} \sqrt{\sqrt{(Z_1^2 - \bar{Z}_2^2)^2 + (K^2 \lambda_2 \bar{\lambda}_1 - 2Z_1 \bar{Z}_2)^2} + Z_1^2 - \bar{Z}_2^2} \right)^2}{\left( \omega^3 \frac{C_p}{C_s} \lambda_2 + \bar{Z}_2 \right)^2 + (\omega^3 \bar{\lambda}_1 + Z_1)^2},$$

where

$$\alpha = (K^2 \lambda_2 \bar{\lambda}_1 - 2Z_1 \bar{Z}_2 + i(Z_1^2 - \bar{Z}_2^2)), \quad \bar{Z}_2 = (4C_s^3 k^2 + C_p \omega^2) \bar{\lambda}_1 \bar{\lambda}_2,$$

and  $\operatorname{csgn}$  is the complex sign defined as in 2.24. We can easily see that an upper

bound for the absolute values of the eigenvalues is the following quantity:

$$M := \frac{\left( \omega^3 \frac{C_p}{C_s} \lambda_2 + \frac{\sqrt{2}}{2} \sqrt{\left( (Z_1^2 - \bar{Z}_2^2)^2 + (K^2 \lambda_2 \bar{\lambda}_1 - 2Z_1 \bar{Z}_2)^2 \right)^{\frac{1}{2}} - Z_1^2 + \bar{Z}_2^2} \right)^2}{\left( \omega^3 \frac{C_p}{C_s} \lambda_2 + \bar{Z}_2 \right)^2 + (\omega^3 \bar{\lambda}_1 + Z_1)^2} + \frac{\left( \omega^3 \bar{\lambda}_1 + \frac{\sqrt{2}}{2} \sqrt{\left( (Z_1^2 - \bar{Z}_2^2)^2 + (K^2 \lambda_2 \bar{\lambda}_1 - 2Z_1 \bar{Z}_2)^2 \right)^{\frac{1}{2}} + Z_1^2 - \bar{Z}_2^2} \right)^2}{\left( \omega^3 \frac{C_p}{C_s} \lambda_2 + \bar{Z}_2 \right)^2 + (\omega^3 \bar{\lambda}_1 + Z_1)^2},$$

It is thus enough to prove that  $M < 1$ . This happens to be true as we can see

$$\begin{aligned} 0 &< \omega^3 \frac{C_p}{C_s} \lambda_2 + \frac{\sqrt{2}}{2} \sqrt{\left( (Z_1^2 - \bar{Z}_2^2)^2 + (K^2 \lambda_2 \bar{\lambda}_1 - 2Z_1 \bar{Z}_2)^2 \right)^{\frac{1}{2}} - Z_1^2 + \bar{Z}_2^2} \\ &< \omega^3 \frac{C_p}{C_s} \lambda_2 + \bar{Z}_2, \end{aligned}$$

and

$$\begin{aligned} 0 &< \omega^3 \bar{\lambda}_1 + \frac{\sqrt{2}}{2} \sqrt{\left( (Z_1^2 - \bar{Z}_2^2)^2 + (K^2 \lambda_2 \bar{\lambda}_1 - 2Z_1 \bar{Z}_2)^2 \right)^{\frac{1}{2}} + Z_1^2 - \bar{Z}_2^2} \\ &< \omega^3 \bar{\lambda}_1 + Z_1. \end{aligned}$$

In both cases we end up with the equivalent condition

$$0 < 4Z_1 \bar{Z}_2 - K^2 \lambda_2 \bar{\lambda}_1 = 4\lambda_2 \bar{\lambda}_1 \frac{C_p}{C_s} \omega^6.$$

In conclusion  $\max(|r_+|, |r_-|) \leq M < 1$  and the algorithm is convergent.

- **Case 4** If  $k = \frac{\omega}{C_s}$  then  $\lambda_1 = 0$  and  $\lambda_2 = \frac{\omega \sqrt{C_p^2 - C_s^2}}{C_s C_p} > 0$ . In this case the coefficients (2.32) of the half-iteration matrix are given by

$$b_{11} = 1, \quad b_{12} \in \mathbb{C}, \quad b_{21} = 0, \quad b_{22} = \frac{-i\sqrt{C_p^2 - C_s^2} - (C_p + C_s)}{i\sqrt{C_p^2 - C_s^2} - (C_p + C_s)}$$

and the eigenvalues  $r_{\pm}$  can be computed from remark (2.1)

$$r_+ = 1, \quad |r_-| = \left| \frac{-i\sqrt{C_p^2 - C_s^2} - (C_p + C_s)}{i\sqrt{C_p^2 - C_s^2} - (C_p + C_s)} \right|^2 = 1 \quad \Rightarrow \quad \rho_{T_0} = 1,$$

therefore the algorithm is divergent.

- **Case 5:** If  $k \in (\frac{\omega}{C_s}, \infty)$  then  $\lambda_{1,2} \in \mathbb{R}_+^*$  and (2.45) gives  $r_{\pm} = \frac{1}{D}(R \pm iI)$  where

$$(2.46) \quad \begin{cases} R = -K^2 \lambda_1 \lambda_2 - \omega^6 \left( \lambda_1 - \lambda_2 \frac{C_p}{C_s} \right)^2 + (Z_1 + Z_2)^2, \\ I = -2\omega^3 \left( \lambda_1 - \lambda_2 \frac{C_p}{C_s} \right) \sqrt{(Z_1 + Z_2)^2 - K^2 \lambda_1 \lambda_2}, \end{cases}$$

$$\Rightarrow R^2 + I^2 - |D|^2 = C(\omega, k, C_p, C_s) \left( K^2 \lambda_1 \lambda_2 - 4 \left( Z_1 Z_2 - \omega^3 \lambda_1 \lambda_2 \frac{C_p}{C_s} \right) \right) = 0,$$

where  $C(\omega, k, C_p, C_s) \in \mathbb{R}^*$  is a constant which means that

$$|r_{\pm}| = 1 \Rightarrow \rho_{T_0} = 1$$

and the algorithms is not convergent in this case.

□

## Convergence analysis in the overlapping case

We investigate now if the combination of overlap and TTC can lead to a convergent optimised Schwarz algorithm. We start again with a numerical experiment in order to gain some insight.

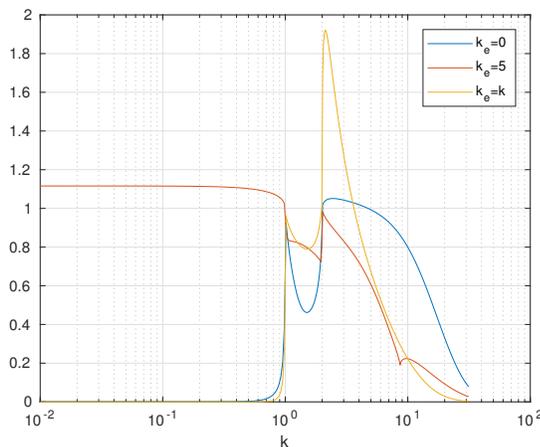


Figure 2.7: Spectrum of the iteration matrix for the optimised Taylor Schwarz method  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\rho = 1$ ,  $\omega = 5$ ,  $\delta = 0.1$ . With absorbing  $\rho_{T_0}$  ( $k_e = 0$ ), general zeroth order  $\rho_{T_{0,e}}$  ( $k_e = 5$ ) and second order  $\rho_{T_2}$  ( $k_e = k$ ) TTC.

For the same parameter choice as for Figure 2.1, we show in the right graph of Figure 2.7 the modulus of the eigenvalues of the optimized Schwarz method with overlap and the three different kind of Taylor transmission conditions we previously introduced. We can see that the absorbing transmission conditions are the best, we therefore focus on them. This leads to the following result

**Theorem 2.4** (Convergence of the overlapping Schwarz algorithm with TTC.). *For a small enough  $\delta$ , the overlapping Schwarz method with absorbing transmission conditions converges for*

$$k \in \left(0, \frac{\omega}{C_p}\right) \cup \left(\frac{\omega}{C_p}, \frac{\omega}{C_s}\right) \cup (k^*, \infty), \quad k^* (\omega, C_p, C_s, \delta) \in \left(\frac{\omega}{C_s}, \infty\right)$$

but diverges for  $k \in \left\{\frac{\omega}{C_p}\right\} \cup \left[\frac{\omega}{C_s}, k^*\right]$ . Therefore the algorithm is in general convergent except for a small interval in the neighbourhood of  $\frac{\omega}{C_s}$  and for the cut-off frequency  $\frac{\omega}{C_p}$ .

*Proof.* Under the specific hypothesis of the theorem, the half-iteration matrix from formula (2.35) is again given by

$$B = \frac{1}{D} \begin{bmatrix} -Z_1 - Z_2 - i\omega^3 \left(\lambda_1 - \lambda_2 \frac{C_p}{C_s}\right) & i\lambda_2 K \\ -i\lambda_1 K & -Z_1 - Z_2 + i\omega^3 \left(\lambda_1 - \lambda_2 \frac{C_p}{C_s}\right) \end{bmatrix} =: \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix},$$

where

$$Z_1 = C_s^3 (k^2 + \lambda_1^2)^2 + \omega^2 C_p k^2, \quad Z_2 = (4C_s^3 k^2 + C_p \omega^2) \lambda_1 \lambda_2, \\ K = 2k (C_p \omega^2 + 2C_s^3 (k^2 + \lambda_1^2)), \quad D = -Z_1 + Z_2 + i\omega^3 \left(\lambda_1 + \lambda_2 \frac{C_p}{C_s}\right).$$

The eigenvalues of the iteration matrix are

$$(2.47) \quad r_{\pm} = \frac{X^2}{2} + Y \pm \frac{1}{2} \sqrt{X^2 (X^2 + 4Y)}.$$

where

$$X = e^{-\lambda_1 \delta} b_{11} - e^{-\lambda_2 \delta} b_{22}, \quad Y = \frac{b_{11} b_{22} - b_{12} b_{21}}{e^{\lambda_1 \delta} e^{\lambda_2 \delta}}.$$

We define now  $\bar{\lambda}_j \in \mathbb{R}_+$ ,  $j = 1, 2$  as in (2.22) when  $\lambda_1$  and/or  $\lambda_2 \in i\mathbb{R}$ .

In the case when the overlap  $\delta$  is small, the series expansion of these eigenvalues is

$$(2.48) \quad r_{\pm} = (R_{1\pm} + iI_{1\pm}) + (R_{2\pm} + iI_{2\pm}) \delta + \mathcal{O}(\delta^2), \quad (R_{j\pm}, I_{j\pm}) \in \mathbb{R},$$

with the modulus is

$$|r_{\pm}|^2 = (R_{1\pm}^2 + I_{1\pm}^2) + 2\delta (R_{1\pm}R_{2\pm} + I_{1\pm}I_{2\pm}) + \mathcal{O}(\delta^2).$$

Again we need to distinguish several cases

- **Case 1:** If  $k \in \left(0, \frac{\omega}{C_p}\right)$  then  $\lambda_{1,2} \in i\mathbb{R}_+$  and  $I_{1\pm} = R_{2\pm} = 0$  for both eigenvalues. Therefore the series expansion (2.48) becomes

$$r_{\pm} = R_{1\pm} + iI_{2\pm}\delta + \mathcal{O}(\delta^2) \quad \Rightarrow \quad |r_{\pm}|^2 = R_{1\pm}^2 + \mathcal{O}(\delta^2),$$

where

$$R_{1\pm} = \frac{\omega^6 \left(\bar{\lambda}_1 - \bar{\lambda}_2 \frac{C_p}{C_s}\right)^2 + (Z_1 + Z_2)^2 + 4k^2 \bar{\lambda}_1 \bar{\lambda}_2 (4C_s^3 k^2 + C_p \omega^2 - 2C_s \omega^2)^2}{\left(Z_1 - Z_2 + \omega^3 \left(\bar{\lambda}_1 + \bar{\lambda}_2 \frac{C_p}{C_s}\right)\right)^2} \\ \pm 2\omega^3 \left(\bar{\lambda}_1 - \bar{\lambda}_2 \frac{C_p}{C_s}\right) \frac{\sqrt{(Z_1 + Z_2)^2 + 4k^2 \bar{\lambda}_1 \bar{\lambda}_2 (4C_s^3 k^2 + C_p \omega^2 - 2C_s \omega^2)^2}}{\left(Z_1 - Z_2 + \omega^3 \left(\bar{\lambda}_1 + \bar{\lambda}_2 \frac{C_p}{C_s}\right)\right)^2}.$$

After simplifications this gives exactly the same convergence factor as in the non-overlapping case which we have proven that it is less than one. Therefore the algorithm is convergent in this case.

- **Case 2:** If  $k = \frac{\omega}{C_p}$  then  $\lambda_1 = i \frac{\omega \sqrt{C_p^2 - C_s^2}}{C_s C_p}$  and  $\lambda_2 = 0$ .

The coefficients (2.32) of the half-iteration matrix are given by

$$b_{11} = \frac{(C_p + C_s)(C_p^3 - 4C_p C_s^2 + 4C_s^3) - \sqrt{C_p^2 - C_s^2} C_p^3}{(C_p + C_s)(C_p^3 - 4C_p C_s^2 + 4C_s^3) + \sqrt{C_p^2 - C_s^2} C_p^3}, \quad b_{12} = 0, \quad b_{21} \in \mathbb{C}, \quad b_{22} = 1.$$

The eigenvalues  $r_{\pm}$  can be computed from remark 2.1

$$(2.49) \quad r_+ = 1, \quad |r_-| = \left| e^{-2i\bar{\lambda}_1 \delta} \frac{(C_p + C_s)(C_p^3 - 4C_p C_s^2 + 4C_s^3) - \bar{\lambda}_1 C_p^4 C_s}{(C_p + C_s)(C_p^3 - 4C_p C_s^2 + 4C_s^3) + \bar{\lambda}_1 C_p^4 C_s} \right|^2$$

Since  $C_p^3 - 4C_p C_s^2 + 4C_s^3 > 0 \Rightarrow |r_-| < 1$  and  $\rho_{T_0} = 1$  which means the algorithm is divergent in this case.

- **Case 3:** If  $k \in \left(\frac{\omega}{C_p}, \frac{\omega}{C_s}\right)$ , then  $\lambda_1 \in i\mathbb{R}_+$  and  $\lambda_2 \in \mathbb{R}_+$ . The series expansion

(2.48) becomes

$$|r_{\pm}|^2 = (R_{1\pm}^2 + I_{1\pm}^2) + \mathcal{O}(\delta)$$

We notice the terms  $(R_{1\pm} + iI_{1\pm})$  are the same as in the non-overlapping case and we already know that  $(R_{1\pm}^2 + I_{1\pm}^2) < 1$ . We can conclude that it's convergent.

- **Case 4:** If  $k = \frac{\omega}{C_s}$  then  $\lambda_1 = 0$ ,  $\lambda_2 = \frac{\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p} > 0$ . The coefficients (2.32) of the half-iteration matrix are given by

$$b_{11} = 1, \quad b_{12} \in \mathbb{C}, \quad b_{21} = 0, \quad b_{22} = \frac{-i\sqrt{C_p^2 - C_s^2} - (C_p + C_s)}{i\sqrt{C_p^2 - C_s^2} - (C_p + C_s)}$$

and the eigenvalues  $r_{\pm}$  of the iteration matrix given by remark 2.1

(2.50)

$$r_+ = 1, \quad |r_-| = e^{-2\lambda_2\delta} \left| \frac{-i\sqrt{C_p^2 - C_s^2} - (C_p + C_s)}{i\sqrt{C_p^2 - C_s^2} - (C_p + C_s)} \right|^2 = e^{-2\lambda_2\delta} < 1 \quad \Rightarrow \rho_{T_0} = 1.$$

Again the algorithm is divergent.

- **Case 5:** If  $k \in \left(\frac{\omega}{C_s}, \infty\right)$ , then  $\lambda_{1,2} \in \mathbb{R}_+^*$  and the eigenvalues are given by (2.47). We then use series expansion (2.48) on  $r_{\pm}$  and obtain

$$R_{1\pm} + iI_{1\pm} = \frac{1}{D}(R \pm iI)$$

where the values  $(R, I, D)$  are given in (2.46) and (2.44). Hence  $R_{1\pm}^2 + I_{1\pm}^2 = 1$  and

$$\begin{aligned} R_{1+}R_{2+} + I_{1+}I_{2+} &= -\frac{\left(\lambda_2\lambda_1 K^2 - \omega^6 \left(\lambda_1 - \lambda_2 \frac{C_p}{C_s}\right)^2 - (Z_1 + Z_2)^2\right)^2}{|D|\sqrt{(Z_1 + Z_2)^2 - \lambda_2\lambda_1 K^2}} \\ &\quad \times \left(\sqrt{(Z_1 + Z_2)^2 - \lambda_2\lambda_1 K^2}(\lambda_1 + \lambda_2) - (\lambda_1 - \lambda_2)(Z_1 + Z_2)\right) < 0 \end{aligned}$$

since  $(\lambda_1 - \lambda_2) < 0 < (Z_1 + Z_2)$ .

Since the first eigenvalue is less than one

$$r_+ \approx 1 + R_{1+}R_{2+} + I_{1+}I_{2+} < 1,$$

we will focus now on  $r_- \approx 1 + R_{1-}R_{2-} + I_{1-}I_{2-} =: F(k)$

$$(2.51) \quad F(k) = - \frac{\left( \lambda_2 \lambda_1 K^2 - \omega^6 \left( \lambda_1 - \lambda_2 \frac{C_p}{C_s} \right)^2 - (Z_1 + Z_2)^2 \right)^2}{|D| \sqrt{(Z_1 + Z_2)^2 - \lambda_2 \lambda_1 K^2}} \times \left( \underbrace{\sqrt{(Z_1 + Z_2)^2 - \lambda_2 \lambda_1 K^2} (\lambda_1 + \lambda_2) + (\lambda_1 - \lambda_2)(Z_1 + Z_2)}_{g(k)} \right).$$

We know that  $g(k) \in \mathbb{R}$  as we have seen previously. Our aim is to show in which conditions  $g(k) < 0$  which is equivalent to  $r_- > 1$ . We see that

$$g(k) < 0 \Leftrightarrow 2(Z_1 + Z_2) - K(\lambda_1 + \lambda_2) < 0.$$

We will study the sign of  $g$  in a neighbourhood of  $\frac{\omega}{C_s}$  and for this reason we set  $k = \frac{\omega}{C_s} + \varepsilon$ , and then develop  $g$  as series for small  $\varepsilon$

$$g = \frac{2\omega^4}{C_p C_s^2} \left( (C_s + C_p) C_p - (C_p + 2C_s) \sqrt{C_p^2 - C_s^2} \right) - \frac{2}{C_p} \sqrt{\frac{2\omega^7}{C_s^3}} \left( C_p (C_p + 2C_s) - (C_p + 4C_s) \sqrt{C_p^2 - C_s^2} \right) \sqrt{\varepsilon} + \mathcal{O}(\varepsilon).$$

For sufficiently small values of  $\varepsilon$  (that is for  $k$  very close to  $\frac{\omega}{C_s}$ ), the leading term of this series being negative we have  $r_- > 1$ . On the other side, because of the overlap  $\lim_{k \rightarrow \infty} \rho_{T_0}(k, \omega, C_p, C_s, \delta) = 0$  and by continuity  $\exists (k^*, \bar{k})$  such that

$$\exists k^* > \bar{k} > \frac{\omega}{C_s} \quad \text{such as} \quad \begin{cases} \rho_{T_0}(k^*, \omega, C_p, C_s, \delta) = 1, \\ \rho_{T_0}|_{k > k^*} < 1, \\ \max_{k > \frac{\omega}{C_s}} |\rho_{T_0}| =: \bar{k}, \end{cases}$$

as one can see on Figure 2.7.

□

We need to find the  $k^*$  for which  $\rho_{T_0} = 1$  as we can see in Figure 2.7. Since an analytical formula is impossible to obtain for  $k^*$ , we will derive a numerical estimate of it. We use the ansatz  $k^* = C_* \delta^\beta$  and we find  $\beta$  by fitting the numerical values obtained for different values of  $\delta$ . After that we need to find the constant  $C_*$  by developing the

formula of the convergence factor as a series depending on  $\delta$ . Note that these results would require a more rigorous mathematical proof as they are derived by using purely numerical arguments. A summary of the procedure is given below:

- **Case 1.** If  $\sqrt{2}C_s < C_p < \frac{\sqrt{5}+1}{2}C_s$ , we find numerically  $k^* \approx C_*$  where  $C_*$  is a constant. We know that  $r_+ < r_-$ , therefore asymptotically  $r_- = 1$  for a given  $k^*$  if the second term in the following series is null

$$r_- = 1 + (R_{1-}R_{2-} + I_{1-}I_{2-})\delta + \mathcal{O}(\delta^2).$$

Therefore  $k^*$  verifies

$$(R_{1-}R_{2-}+I_{1-}I_{2-})(k^*) = 0 \Leftrightarrow k^* / \begin{cases} \lambda_2\lambda_1K^2 - \omega^6 \left( \lambda_1 - \lambda_2 \frac{C_p}{C_s} \right)^2 - (Z_1 + Z_2)^2 = 0 \\ \text{and/or } g(k^*) = 0, \end{cases}$$

where  $g(k)$  is defined in (2.51). The only admissible solution is

$$k^*(\omega, C_p, C_s) = C_* = -\frac{(C_p - 2C_s)C_p\omega}{2\sqrt{-C_p^3C_s + 2C_p^2C_s^2 - C_s^4C_s}},$$

Note that the hypothesis on  $C_p$  and  $C_s$  insures that  $-C_p^3C_s + 2C_p^2C_s^2 - C_s^4 > 0$  and  $C^*$  is real.

- **Case 2.** If  $\frac{\sqrt{5}+1}{2}C_s < C_p$ , we find numerically  $k^* \approx \frac{C_*}{\sqrt{\delta}}$  and we plug it into (2.47). By developing the eigenvalues of the iteration matrix we get

$$\begin{cases} r_+ \approx 1 - \frac{C_*^4C_s^4(C_s^2 - 2C_p^2) + C_s^2C_p^2(C_*^4C_p^2 + 3\omega^2) + 3C_p^3\omega^2(C_s - C_p)}{3C_s^2C_*(C_p^2 - C_s^2)^2} \delta^{\frac{3}{2}}, \\ r_- \approx 1 - 4C_*\sqrt{\delta} \end{cases}$$

which means that asymptotically for a small  $\delta$ , we have that  $r_- \leq r_+$ . In order to find  $k^*$  such that  $\rho_{T_0}(k^*) = 1$  we need to solve  $r_+ \approx 1$  with respect to  $C_*$ . We get:

$$C_* = \left( \frac{\sqrt{3}C_p\omega\sqrt{C_p^2 - C_sC_p - C_s^2}}{C_s(C_p^2 - C_s^2)} \right)^{\frac{1}{2}}.$$

In conclusion

$$k^* = \left( \frac{\sqrt{3}C_p\omega\sqrt{C_p^2 - C_sC_p - C_s^2}}{C_s(C_p^2 - C_s^2)} \right)^{\frac{1}{2}} \delta^{-\frac{1}{2}}.$$

Note that  $k^* > \frac{\omega}{C_s}$  for a  $\delta$  small enough. Moreover  $C_p^2 - C_sC_p - C_s^2 > 0$  since  $\frac{\sqrt{5}+1}{2}C_s < C_p$ .

In the same way we can also find a formula for the maximum point  $\bar{k}$ . We use again the ansatz  $\bar{k} = \bar{C}\delta^\beta$  and by fitting this formula with the numerical data we get  $\beta$ . In this way, we find numerically that  $\bar{k} \approx \bar{C}$ . In order to find the constant we need to solve the equation:

$$\partial_k \rho(\bar{k}) = 0 \Leftrightarrow F'(\bar{k}) = 0,$$

we find that the only critical point verifying  $\bar{k} > \frac{\omega}{C_s}$  is

$$\frac{\sqrt{(2C_s - C_p) \left( (C_p - 2C_s) \left( C_p - \frac{C_s}{2} \right) - \frac{\sqrt{C_s} \operatorname{sg}(C_p - 2C_s)}{2} \sqrt{4C_p^3 C_s - 3C_p^2 C_s^2 - 4C_p C_s^3 + 4C_s^4} \right) \omega}}{2C_s^{\frac{3}{2}}(C_p - C_s)}.$$

We end up with two different cases according to the value of  $\operatorname{sg}(C_p - 2C_s)$ .

- **Case 1.**  $\operatorname{sg}(C_p - 2C_s) = -1 \Leftrightarrow \sqrt{2}C_s < C_p < 2C_s$ . In this case  $\bar{k} \in \mathbb{R}$  and we get

$$\bar{k} = \frac{\sqrt{2C_s - C_p}\omega}{2C_s^{\frac{3}{2}}(C_p - C_s)} \sqrt{(2C_s - C_p) \left( C_p - \frac{C_s}{2} \right) - \frac{\sqrt{C_s}}{2} \sqrt{4C_p^3 - 3C_p^2 C_s - 4C_p C_s^2 + 4C_s^3}}.$$

- **Case 2.**  $\operatorname{sg}(C_p - 2C_s) = 1 \Leftrightarrow C_p > 2C_s$ . In this case  $\bar{k} \in \mathbb{R}$  under a further constraint that is  $C_p < 4C_s$  and we get

$$\bar{k} = \frac{\sqrt{C_p - 2C_s}\omega}{2C_s^{\frac{3}{2}}(C_p - C_s)} \sqrt{\frac{\sqrt{C_s}}{2} \sqrt{4C_p^3 - 3C_p^2 C_s - 4C_p C_s^2 + 4C_s^3} - (C_p - 2C_s) \left( C_p - \frac{C_s}{2} \right)}.$$

We won't investigate further (for  $C_p > 4C_s$ ), because it doesn't seem to correspond to any physical situation.

In Theorem 2.4 we have supposed that  $\delta$  is small enough which is in most of the cases the most difficult case from the convergence point of view. As it can be seen from the general formula of the convergence factor (2.8) for a sufficiently big overlap and for intermediate and higher frequencies, the algorithm will become convergent as the exponentials will

decrease to zero. For small frequencies, the convergence factor is naturally small as it is based on Taylor approximations around 0 of (exact) transparent boundary conditions. In the following we will try to understand when the transition occurs from divergence to convergence and for which value of the overlap this is achieved. In order to do this we will first perform the following numerical experiment illustrated in Figure 2.8.

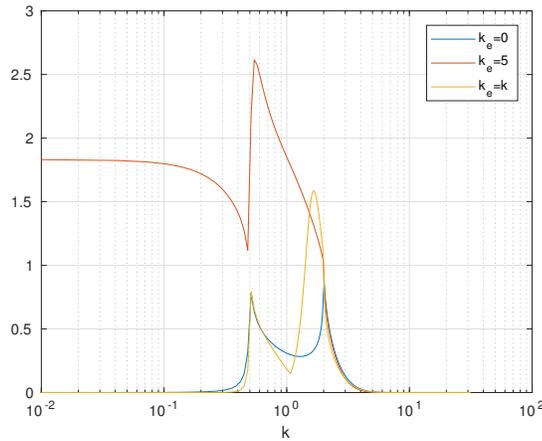


Figure 2.8: Spectrum of the iteration matrix for the optimized Taylor Schwarz method with overlap  $\delta = 1$  and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\rho = 1$  and  $\omega = 1$ . Absorbing BC  $\rho_{T_0}$  ( $k_e = 0$ ), general zeroth order  $\rho_{T_{0,e}}$  ( $k_e = 5$ ) and second order  $\rho_{T_2}$  ( $k_e = k$ ) TTC.

We can see that if the overlap is large enough, it is possible to obtain a convergent optimized Schwarz method except for the frequencies  $k = \frac{\omega}{C_p}$  and  $k = \frac{\omega}{C_s}$ . The best method seems to be that based on absorbing boundary conditions. It would thus be of great interest to estimate the value  $\delta^*(C_p, C_s, \omega)$  for which it converges as soon as the overlap  $\delta > \delta^*(C_p, C_s, \omega)$  like illustrated in Figure 2.9. Then we get

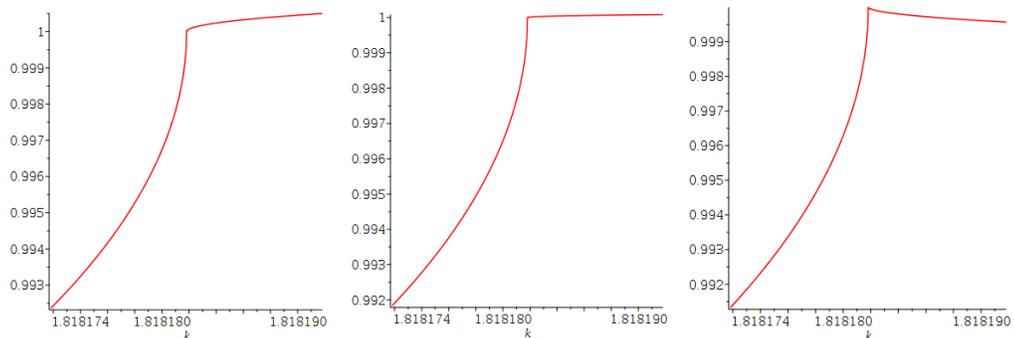


Figure 2.9: Modulus of the eigenvalues of the iteration matrix close to  $k = \frac{\omega}{C_s}$  for the optimized Schwarz method with zeroth order TTC for  $\omega = 1$ . Left:  $\delta = 0.8$  (divergence). Middle:  $\delta = 0.9$  (convergence). Right:  $\delta = 1$  (convergence).

Based on the insight given by this graphic representations, in the following lemma we will derive a formula for the value of the overlap when the transition to a convergent algorithm occurs. Note that the proof cannot rely on the series which are valid for a small  $\delta$  but we can assume that when the algorithm is convergent for a small overlap this will be the case for a sufficiently big overlap.

**Lemma 2.2.** *The overlapping Schwarz algorithm with absorbing boundary conditions will converge for  $k \in \mathbb{R}_+ \setminus \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$  if the overlap  $\delta$  is bigger than*

$$\delta^*(C_p, C_s, \omega) = \frac{C_s \sqrt{C_p^2 - C_s^2} (C_p + 2C_s)^2}{C_p \omega (C_s + C_p)} \frac{\sinh(\alpha)}{C_p \cosh(\alpha) + C_s}$$

where  $\alpha$  is the positive root of

$$\alpha C_p^2 (C_p \cosh(\alpha) + C_s) - (C_p^3 + 3C_p^2 C_s - 4C_s^3) \sinh(\alpha) = 0.$$

*Proof.* A first step into this direction comes from observing how the convergent algorithm turns into a divergent one when  $\delta$  is decreased. It seems that it is sufficient to just check the slope of the modulus of the larger eigenvalue of the iteration matrix for  $\frac{\omega}{C_s}$  coming from the right. We will distinguish again five different cases.

If  $k \in \left(0, \frac{\omega}{C_s}\right) \setminus \left\{ \frac{\omega}{C_p} \right\}$ , which corresponds to cases 1 and 3, the Schwarz algorithm with absorbing boundary conditions converges both without overlap (see theorem 2.3) and with a small overlap (see theorem 2.4). A sufficiently big overlap can only improve the behaviour of the algorithm in the mid-frequency regime and not deteriorate it in the low frequency regime, therefore it will be convergent in this case.

If  $k \in \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$ , which corresponds to cases 2 and 4, we see from (2.49) and (2.50) that the convergence factor is independent of the size of  $\delta$ .

If  $k \in \left(\frac{\omega}{C_s}, \infty\right)$  we will analyse the behaviour of the algorithm in the neighbourhood of  $\frac{\omega}{C_s}$  and see when it becomes divergent. No argument on the size of  $\delta$  is used in this case. In order to do this, we define  $\varepsilon$  such as  $k = \frac{\omega}{C_s} + \varepsilon$  and develop  $r_{\pm}$  from (2.47) in series as in (2.48) for a small  $\varepsilon$  with  $(R_{j\pm}, I_{j\pm}) \in \mathbb{R}$  and then get the modulus

$$|r_{\pm}|^2 = (R_{1\pm}^2 + I_{1\pm}^2) + 2\sqrt{\varepsilon} (R_{1\pm} R_{2\pm} + I_{1\pm} I_{2\pm}) + \mathcal{O}(\varepsilon).$$

For  $r_+$ , we have on one hand that

$$R_{1+} + iI_{1+} = -\frac{C_p^2 - 2C_s^2 - i2C_s\sqrt{C_p^2 - C_s^2}}{C_p^2 e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_p C_s}\delta}} \Rightarrow R_{1+}^2 + I_{1+}^2 = e^{-\frac{4\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} < 1,$$

and similarly for  $r_-$  we have  $R_{1-} + iI_{1-} = 1 \Rightarrow R_{1-}^2 + I_{1-}^2 = 1$ . On the other hand

$$R_{1+}R_{2+} + I_{1+}I_{2+} = -\frac{2\sqrt{2}C_s e^{-\frac{4\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} \sqrt{C_p^2 - C_s^2} (C_p + 2C_s)^2 \left( e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} - 1 \right)}{C_p \sqrt{\omega} (C_p + C_s) \left( C_p e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} + 2C_s e^{\frac{\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} + C_p \right)} < 0,$$

from which we can conclude that  $|r_+|^2 < 1$  and

$$R_{1-}R_{2-} + I_{1-}I_{2-} = -\frac{2\sqrt{2}(\omega C_s)^{-\frac{1}{2}}}{(C_p + C_s)C_p} \underbrace{\left( \delta C_p \omega (C_p + C_s) - \frac{C_s (C_p + 2C_s)^2 \sqrt{C_p^2 - C_s^2} \left( e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} - 1 \right)}{C_p e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} + 2C_s e^{\frac{\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} + C_p} \right)}_{=:f(\delta)}.$$

We study  $f$  to see if and when it becomes negative. We take

$$(2.52) \quad f'(\delta) = -\frac{2\sqrt{2}\omega e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta}}{\sqrt{C_s}C_p^2 \left( C_p e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} + 2C_s e^{\frac{\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} + C_p \right)^2} g(\delta),$$

so  $f'$  is of opposite sign w.r.t.  $g$  where

$$\begin{cases} g(\delta) = 2C_p^4 \cosh\left(\frac{2\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2} \delta\right) - 2C_p(C_p^3 + 6C_p^2 C_s - 2C_p C_s^2 - 8C_s^3) \\ \quad + 4C_s(C_p + C_s)(C_p - 2C_s)^2 \cosh\left(\frac{\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2} \delta\right), \\ g'(\delta) = \frac{4}{C_p} \omega (C_p + C_s)(C_p - 2C_s)^2 \sqrt{C_p^2 - C_s^2} \sinh\left(\frac{\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2} \delta\right) \\ \quad + \frac{4}{C_s} \omega \sqrt{C_p^2 - C_s^2} \sinh\left(\frac{2\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2} \delta\right) > 0 \quad \forall \delta, \end{cases}$$

We know that  $\cosh$  is a strictly increasing function for positive values and in our case all the parameters are real and positive. We have  $\cosh(\delta = 0) = 1$  and we denote

$$\exists! \bar{\delta} \in \mathbb{R}_+^* / \cosh\left(\frac{\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2 \bar{\delta}}\right) = 3.$$

Since we have

$$\cosh\left(\frac{\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2 \delta}\right) \leq \cosh\left(2 \frac{\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2 \delta}\right),$$

this implies

$$\begin{aligned} g(\delta) &\geq 2C_p^4 \cosh\left(\frac{\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2 \delta}\right) - 2C_p(C_p^3 + 6C_p^2 C_s - 2C_p C_s^2 - 8C_s^3) \\ &\quad + 4C_s(C_p + C_s)(C_p - 2C_s)^2 \cosh\left(\frac{\omega}{C_s C_p} \sqrt{C_p^2 - C_s^2 \delta}\right), \end{aligned}$$

then

$$g(\bar{\delta}) \geq 16C_p^2(C_p^2 - 2C_s^2) + 16C_p C_s^3 + 48C_s^4 > 0.$$

As a result  $\exists \hat{\delta}$  s.t.  $g(\hat{\delta}) = 0$  and we know that  $g$  is increasing.

Since  $f(0) = 0$ ,  $f$  is a strictly increasing function  $\forall \delta < \hat{\delta}$  and decreasing  $\forall \delta > \hat{\delta}$  and

$$\begin{aligned} \partial_\delta^2 f &= -\frac{4\sqrt{2\omega^3(C_p^2 - C_s^2)}(C_p + 2C_s)^2(C_p - C_s)e^{\frac{\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta}}{(\sqrt{C_s}C_p)^3 \left(C_p e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} + 2C_s e^{\frac{\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} + C_p\right)^3} \\ &\times \left[ \left(e^{\frac{4\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} - 1\right) C_p C_s + 2(2C_p^2 - C_s^2) e^{\frac{\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} \left(e^{\frac{2\omega\sqrt{C_p^2 - C_s^2}}{C_s C_p}\delta} - 1\right) \right] < 0, \end{aligned}$$

therefore  $\hat{\delta}$  is a absolut maximum for  $f$ .

Since  $\lim_{\delta \rightarrow \infty} f(\delta) \rightarrow -\infty$ , its graph will cut the x-axis only once. By solving the equation  $f(\delta) = 0$  w.r.t.  $\delta$  we get:

$$\begin{aligned} \delta^*(C_p, C_s, \omega) &= \frac{C_s \sqrt{C_p^2 - C_s^2} (C_p + 2C_s)^2}{C_p \omega (C_s + C_p)} \frac{e^{2\alpha} - 1}{C_p e^{2\alpha} + 2e^\alpha C_s + C_p} \\ &= \frac{C_s \sqrt{C_p^2 - C_s^2} (C_p + 2C_s)^2}{C_p \omega (C_s + C_p)} \frac{\sinh(\alpha)}{C_p \cosh(\alpha) + C_s}, \end{aligned}$$

where  $\alpha$  the positive root of

$$0 = [(\alpha - 1)(C_p^3 - 3C_p^2C_s + 4C_s^3)e^{2\alpha} + 2\alpha C_p^2C_s e^\alpha + (\alpha + 1)(C_p^3 + 3C_p^2C_s - 4C_s^3)]$$

$$\Leftrightarrow \alpha / \alpha C_p^2 (C_p \cosh(\alpha) + C_s) = (C_p^3 + 3C_p^2C_s - 4C_s^3) \sinh(\alpha).$$

Note that  $\alpha = 0$  is also solution but we have that  $\delta^* > 0 \Rightarrow \alpha > 0$ .  $\square$

## Numerical results

In this section we illustrate the different convergence/divergence regimes of the iterative version of the Schwarz algorithm

$$\mathbf{U}^{n+1} = \mathbf{U}^n + M^{-1}(\mathbf{F} - A\mathbf{U}^n),$$

where  $M^{-1}$  is either to RAS or the ORAS preconditioners:

$$(2.53) \quad M_{RAS}^{-1} = \sum_{i=1}^N R_i^T D_i A_i^{-1} R_i, \quad M^{-1} = \sum_{i=1}^N R_i^T D_i B_i^{-1} R_i$$

where  $B_i$  are the local matrices derived from the discretisation of boundary value problems with absorbing boundary (or Taylor transmission conditions) conditions and  $A_i = R_i A R_i^T$ .

### Two-subdomains case: iterative Schwarz with TTC

We have seen previously that the iterative Schwarz algorithm with Taylor transmission conditions can converge outside the cut-off frequencies  $\frac{\omega}{C_p}$  and  $\frac{\omega}{C_s}$  provided that the overlap is big enough. For a lower value of the overlap, the algorithm is divergent in an interval of frequencies and is dominated by a frequency  $k^*$  which is slightly bigger than  $\frac{\omega}{C_s}$ . We will illustrate these findings by some numerical experiments.

We choose again  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\rho = 1$ .

We discretise the time-harmonic Navier equations using P1 finite elements on the domain  $\Omega = (-1, 1) \times (0, 1)$ , use the two subdomains  $\Omega_1 = (-1, 2h) \times (0, 1)$  and  $\Omega_2 = (-2h, 1) \times (0, 1)$  with  $h = \frac{1}{80}$ , in other words we build a uniform decomposition into two overlapping subdomains. This time we use Dirichlet boundary conditions on the longer sides of the rectangle. The overlapping parameter is first chosen to be  $\delta = 2h$ . We show in Figure 2.10 the error in modulus at iteration 60 of the optimised

Schwarz method which uses Taylor transmission conditions for  $\omega = 5$ .

The error is computed with respect to the solution of the algebraic system obtain on the global domain  $A\mathbf{U} = \mathbf{F}$  by a direct method. Note that in more realistic test cases this reference solution is not available. In those cases the global relative residual will be a measure of the convergence

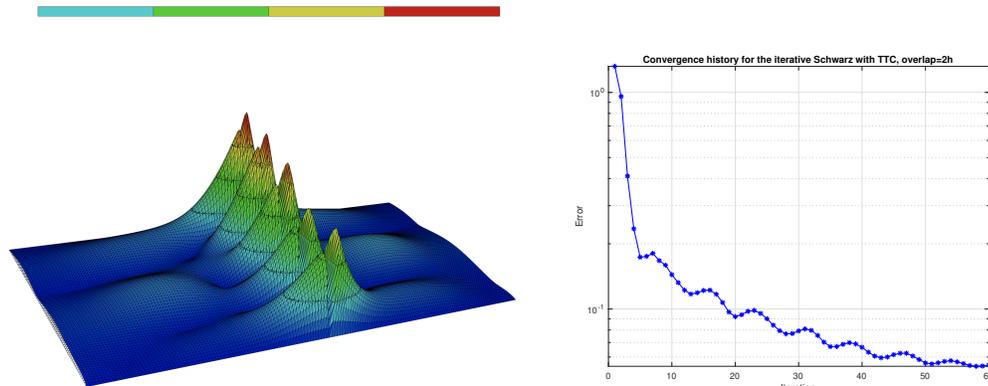


Figure 2.10: Error in modulus at iteration 60 of the iterative Schwarz method with TTC and convergence history ( $\omega = 5$ ,  $\delta = 2h$ ).

We see that the iterative method is not converging (the error after 60 iterations stagnates around the value of  $5.5e - 2$ ). This time the interval on which the method is diverging is  $\left[\frac{\omega}{C_s}, k^*\right] = [10, k^*]$ . We can see that the error has 5 bumps along the interface which corresponds well to the mode  $|\sin(ky)|$  along the interface for  $k = 5\pi \approx 15$ . This seems to be the fastest diverging mode whose existence the analysis can prove.

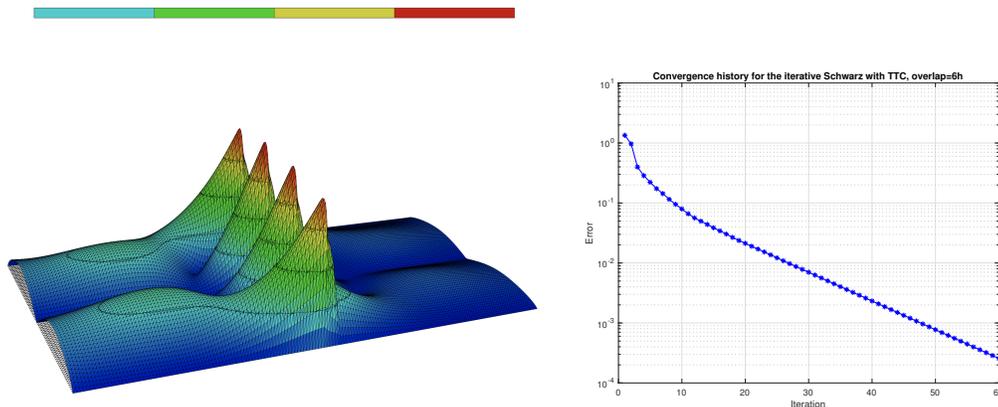


Figure 2.11: Error in modulus at iteration 60 of the iterative Schwarz method with TTC and convergence history ( $\omega = 5$ ,  $\delta = 6h$ ).

If we increase the overlap the method will converge provided that the "cut-off" fre-

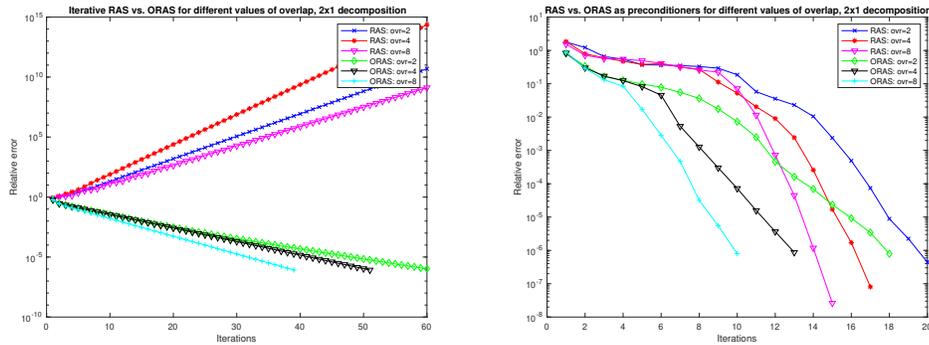


Figure 2.12: Convergence history for RAS and ORAS as solvers (left) and preconditioners (right) for  $\omega = 5$ ,  $2 \times 1$  subdomains different values of  $\delta$ .

quencies  $\frac{\omega}{C_p}$  and  $\frac{\omega}{C_s}$  can't be represented on the mesh, which seems to be the case here (see Figure 2.11). A slowly converging mode can be seen again and it corresponds to a mode  $|\sin(ky)|$  along the interface, except that the error has now decreased to  $2.5e - 4$  (for a global residual of  $9.4e - 6$ ).

### Schwarz method as solver and as a preconditioner

We simulate the wave propagation through a computational domain which is given by the unit square  $[0, 1]^2$  with Robin boundary conditions on a part of the boundary  $(\mathcal{T}^{(\mathbf{n})} - i\sigma_{\mathbf{n}}) \mathbf{u} = \mathbf{g}$ , with the source term  $\mathbf{g}$  chosen such that the exact solution is a plane wave  $\mathbf{u}^{inc}$  consisting both P- and S-waves  $\mathbf{u}^{inc} = \mathbf{d} e^{i\kappa_p \mathbf{x} \cdot \mathbf{d}} + \mathbf{d}^\perp e^{i\kappa_s \mathbf{x} \cdot \mathbf{d}}$ ,  $\mathbf{d} = (\cos(\frac{\pi}{3}), \cos(\frac{\pi}{3}))^T$ . Note that in the two-dimensional case considered here

$$(2.54) \quad \sigma_{\mathbf{n}} = \omega \rho \begin{pmatrix} c_p n_x^2 + c_s n_y^2 & (c_p - c_s) n_x n_y \\ (c_p - c_s) n_x n_y & c_p n_y^2 + c_s n_x^2 \end{pmatrix}.$$

In the first test case the physical parameters are given by  $C_p = 1$ ,  $C_s = 0.5$ ,  $\rho = 1$ ,  $\lambda = \rho(C_p^2 - 2C_s^2)$ ,  $\mu = \rho C_s^2$ ,  $\omega = 5$ . This test case does not necessarily correspond to an accurate physical situation but it produces simple but oscillatory enough solutions reflecting the difficulties related to the solving of the problem. We will test the two versions of RAS and ORAS in (2.53) on a uniform decomposition of the rectangle  $[0, 2] \times [0, 1]$  into  $2 \times 1$  subdomains having each one  $40 \times 40$  discretisation points for a total number of 6400 dof per subdomain. We will then repeat the experience by solving the global preconditioned system  $\mathbf{A}\mathbf{U} = \mathbf{F}$  by GMRES method. The behaviour of the algorithms for different values of overlap is shown in Figure 2.12.

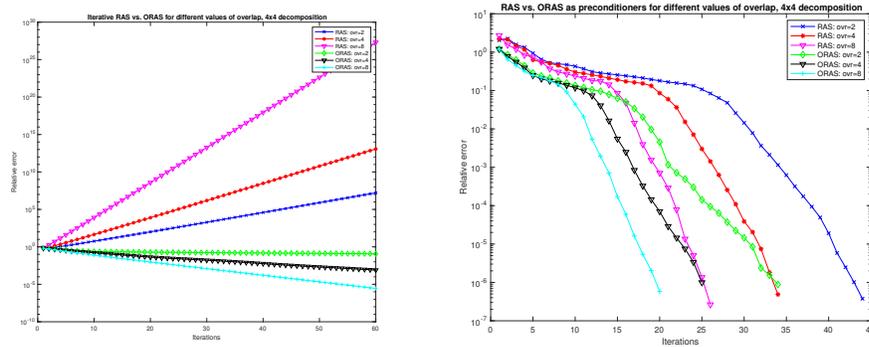


Figure 2.13: Convergence history for RAS and ORAS as solvers (left) and preconditioners (right) for  $\omega = 5$ ,  $4 \times 4$  subdomains, different values of  $\delta$ .

We repeat the experience on a uniform decomposition of the unit square  $[0, 1]^1$  into  $4 \times 4$  subdomains with the same local number of degrees of freedom. The behaviour of the algorithms for different values of overlap is shown in Figure 2.13.

As expected, in its iterative version, the ORAS algorithm outperforms RAS, the latter not being convergent for any value of the overlap. Another notable difference is that by increasing the overlap, the iterative version of the ORAS algorithm is getting better whereas the performance of RAS is getting worse. Even if in practice one won't use these iterative versions of the algorithms, this comparison provides us a very useful insight of their behaviour.

When used as preconditioners in a GMRES method, again ORAS is clearly better than RAS, the former being less dependent on the overlap as RAS. Note that for a bigger value of the overlap the difference between the two methods is reduced even if we can still notice a slightly better behaviour of the ORAS algorithm.

## Chapter 3

# Optimised Schwarz methods

In the previous chapter we have seen that classical Schwarz methods based on Dirichlet or even Robin transmission conditions are not very effective when used as iterative solvers. For this reason we would like to introduce a new class of methods, namely the *optimised Schwarz methods* by designing better transmission conditions between the subdomains.

### 3.1 State of the Art

Over the last two decades, a lot of results have been obtained about optimised algorithms based on well chosen parameters in the transmission conditions. Different types of equations from the symmetric positive definite scalar equations to indefinite systems of PDEs, have been thus analysed.

For the case of steady state symmetric problems, we can mention the self contained overview article by Gander [Gan06], the later contains a very exhaustive state of the art and the description of the techniques used to tackle these kind of algorithms. Several extensions to the advection-diffusion type problems can be found in the works of Nataf et al. [JNR98, JN00, JNR01, LMO00, Nat96, NN97].

Among numerous works on the topic, several aspects have been approached. For example, the case of problems with discontinuous coefficients in [Dub07, Fla01, GN04], the influence of the geometry on the behaviour of the algorithms [Gan11], the construction of coarse grid corrections [DGL<sup>+</sup>12, KL15, LNS15], the presence of cross points [GK12] or an accurate analysis for circular domains [GX14].

The general principle of the construction of optimised interface conditions is based on polynomial (local) approximations of the Fourier symbols of the exact or transparent boundary conditions (non-local operators). An alternative would be to use rational approximations of Padé type instead of polynomials and in the works of Antoine et al. [ABG12]. We should note that perfectly matched layers (PML) [Ber94] [CW94] are also used in domain decomposition methods see [SZB<sup>+</sup>07], [GN00] or [PELY13].

For the Helmholtz equation, which is the prototype of elliptic indefinite wave type of problems with oscillatory solution, optimised transmission conditions were developed for the first time by Desprès in [Des90] and [Des91] and later on by Chevalier in [Che98, CN98], Collino [CDJP97b] or Gander et al. [GMN02a, GHM07a].

Very similar in nature to Helmholtz equations, high-frequency time-harmonic Maxwell's equations are also very difficult to solve and the design of sophisticated iterative methods seems to be quite complex. Nevertheless such attempts to develop optimised algorithms both for the first order and the second order formulations can be found chronologically in the works of Chevalier [Che98][section 4.7], Collino [CDJP97a], Alonso [ARGG06], Lee et al [PRL10, PL10, RL10] or Dolean et al. [DLP08a, DLP08b, DGG09, EDGL12a, EDGL12b].

The implementation of sophisticated transmission conditions is generally not easy especially in the case of Discontinuous Galerkin methods, therefore a special treatment is needed [DLP08a, DLP08b, EDG<sup>+</sup>11]. Same statement holds in the case of edge element discretisations of the Maxwell's equations or even for non-conforming discretisations. A certain number examples used in computations of non trivial multi-scale electromagnetic radiation and scattering problems can be found in the works of Lee et al. [LVL05, PRL10, PL10, RL10]. First order and second order formulations of Maxwell's equations lead to different optimisation results, which have nevertheless a common ground. The presentation of such a unified framework is detailed in [DGL<sup>+</sup>13, DGL<sup>+</sup>14].

As a general rule, the method of deriving optimised transmission conditions is quite general and can be in principle applied to a big variety of equations. To our knowledge the case of time-harmonic elastic waves (Navier equations) has not been studied so far and our purpose is to apply the techniques from the state of the art to Navier equations while adapting them to the specificities of the problem.

### 3.2 One parameter family of transmissions conditions

In the following we present one possible strategy of improving the interface transmission conditions. We use as a starting point the expression of the transparent Boundary Conditions (TBC) (2.29) in which we fix one frequency  $k = k_e$ . The resulting operators  $\widehat{\mathcal{S}}_j^E$  will be local, but exact for the chosen frequency:

$$\begin{aligned}
 \widehat{\mathcal{S}}_1^E(1,1) &= \rho\omega^2 \frac{\sqrt{k_e^2 - \frac{\omega^2}{C_s^2}}}{k_e^2 - \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}} \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}}, \\
 \widehat{\mathcal{S}}_1^E(1,2) &= +ik_e\rho \left( 2C_s^2 - \frac{\omega^2}{k_e^2 - \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}} \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}} \right), \\
 \widehat{\mathcal{S}}_1^E(2,1) &= -ik_e\rho \left( 2C_s^2 - \frac{\omega^2}{k_e^2 - \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}} \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}} \right), \\
 \widehat{\mathcal{S}}_1^E(1,1) &= \rho\omega^2 \frac{\sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}}{k_e^2 - \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}} \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}},
 \end{aligned}
 \tag{3.1}$$

with

$$\widehat{\mathcal{S}}_2^E(1,1) = \widehat{\mathcal{S}}_1^E(1,1), \quad \widehat{\mathcal{S}}_2^E(1,2) = -\widehat{\mathcal{S}}_1^E(1,2), \quad \widehat{\mathcal{S}}_2^E(2,1) = -\widehat{\mathcal{S}}_1^E(2,1), \quad \widehat{\mathcal{S}}_2^E(2,2) = \widehat{\mathcal{S}}_1^E(2,2).$$

We thus expect that the convergence properties in a neighbourhood of this frequency will be very good. Afterwards we can optimise the convergence factor on the whole range of frequencies with respect to  $k_e$ .

To simplify the notations we denote

$$p_1 := \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}}, \quad p_2 := \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}.$$

We name these conditions *Optimised Interface Conditions (OIC)*. We see that (3.1) gives a one parameter family of simple transmission conditions, and one can try to find the best choice for  $k_e$  to minimise the maximum of the contraction factor, excluding the frequencies  $k = \frac{\omega}{C_p}$  and  $k = \frac{\omega}{C_s}$  where the algorithm will never converge as well as a small interval around them, as it was done the case of the Helmholtz equation ([GMN02b],[GHM07b]).

**Lemma 3.1** (Convergence factor in the general case). *For a given initial guess  $\mathbf{u}_1^0 \in (L^2(\Omega_1))^2$ ,  $\mathbf{u}_2^0 \in (L^2(\Omega_2))^2$ , the Schwarz algorithm with OIC has the following convergence factor*

$$\rho_E(k, k_e, \omega, C_p, C_s, \delta) = \max\{|r_+|, |r_-|\}, \quad r_{\pm} = \frac{X^2}{2} + Y \pm \frac{1}{2} \sqrt{X^2(X^2 + 4Y)},$$

with

$$(3.2) \quad X = e^{-\lambda_1 \delta} b_{11} - e^{-\lambda_2 \delta} b_{22}, \quad Y = \frac{b_{11} b_{22} - b_{12} b_{21}}{e^{\lambda_1 \delta} e^{\lambda_2 \delta}}, \quad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} =: B,$$

where

$$(3.3) \quad b_{11} = \frac{k^2(k_e + C_2)^2 + \lambda_1 \lambda_2 (k_e + C_1)^2 - p_1 p_2 (k^2 + \lambda_1 \lambda_2) + (k_e^2 - p_1 p_2) (\lambda_1 p_2 - \lambda_2 p_1)}{k^2(k_e + C_2)^2 - \lambda_1 \lambda_2 (k_e + C_1)^2 - p_1 p_2 (k^2 - \lambda_1 \lambda_2) - (k_e^2 - p_1 p_2) (\lambda_1 p_2 + \lambda_2 p_1)},$$

$$b_{12} = \frac{-2ik((k_e + C_1)(k_e + C_2) - p_1 p_2) \lambda_2}{k^2(k_e + C_2)^2 - \lambda_1 \lambda_2 (k_e + C_1)^2 - p_1 p_2 (k^2 - \lambda_1 \lambda_2) - (k_e^2 - p_1 p_2) (\lambda_1 p_2 + \lambda_2 p_1)},$$

$$b_{21} = \frac{2ik((k_e + C_1)(k_e + C_2) - p_1 p_2) \lambda_1}{k^2(k_e + C_2)^2 - \lambda_1 \lambda_2 (k_e + C_1)^2 - p_1 p_2 (k^2 - \lambda_1 \lambda_2) - (k_e^2 - p_1 p_2) (\lambda_1 p_2 + \lambda_2 p_1)},$$

$$b_{22} = \frac{k^2(k_e + C_2)^2 + \lambda_1 \lambda_2 (k_e + C_1)^2 - p_1 p_2 (k^2 + \lambda_1 \lambda_2) - (k_e^2 - p_1 p_2) (\lambda_1 p_2 - \lambda_2 p_1)}{k^2(k_e + C_2)^2 - \lambda_1 \lambda_2 (k_e + C_1)^2 - p_1 p_2 (k^2 - \lambda_1 \lambda_2) - (k_e^2 - p_1 p_2) (\lambda_1 p_2 + \lambda_2 p_1)},$$

and

$$C_1 = 2 \frac{C_s^2}{\omega^2} (k - k_e) (k_e^2 - p_1 p_2), \quad C_2 = \frac{C_s^2}{k \omega^2} (k_e^2 - p_1 p_2) (k^2 - 2k k_e + \lambda_1^2).$$

The particular case without overlap leads to the convergence factor

$$(3.4) \quad |r_{\pm}| = \left| \frac{(p_1 \lambda_2 - \lambda_1 p_2) (k_e^2 - p_1 p_2) \pm \sqrt{(Z_1 - Z_2)^2 - 4 \lambda_1 \lambda_2 p_1 p_2 (k_e^2 - p_1 p_2)^2}}{(p_1 \lambda_2 + \lambda_1 p_2) (k_e^2 - p_1 p_2) + Z_2 - Z_1} \right|^2,$$

where

$$Z_1 = k^2 \left( (k_e + C_2)^2 - p_1 p_2 \right), \quad Z_2 = \lambda_1 \lambda_2 \left( (k_e + C_1)^2 - p_1 p_2 \right).$$

*Proof.* We use the general result on Schwarz method with general transmission conditions from Lemma 2.1 in which we insert the new boundary conditions (3.1). We get

the two following matrices in the interface iterations

$$B_1 = \frac{\rho\omega^2}{k(k_e^2 - p_1p_2)} \begin{bmatrix} kp_1 - k_e\lambda_1 - \lambda_1C_1 & i(k(k_e + C_2) - \lambda_2p_1) \\ -i(k(k_e + C_2) - \lambda_1p_2) & kp_2 - k_e\lambda_2 - \lambda_2C_1 \end{bmatrix}$$

and

$$B_2 = \frac{\rho\omega^2}{k(k_e^2 - p_1p_2)} \begin{bmatrix} kp_1 + k_e\lambda_1 + \lambda_1C_1 & i(k(k_e + C_2) + \lambda_2p_1) \\ -i(k(k_e + C_2) + \lambda_1p_2) & kp_2 + k_e\lambda_2 + \lambda_2C_1 \end{bmatrix}.$$

After some computations we obtain the half-iteration matrix  $B = B_2^{-1}B_1$  involved in (3.2) and (3.3) and the resulting convergence factor that will be denoted  $\rho_E$ .

In the case without overlap  $\delta = 0$ , these formulae simplify

$$X = \frac{-2(p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2)}{Z_1 - Z_2 - (p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2)},$$

$$Y = \frac{(Z_1 + Z_2)^2 - ((p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2))^2 - 4k^2((k_e + C_2)(k_e + C_1) - p_1p_2)^2\lambda_1\lambda_2}{(Z_1 - Z_2 - (p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2))^2},$$

and lead to

$$\begin{aligned} r_{\pm} &= \frac{((p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2))^2 + (Z_1 + Z_2)^2 - 4k^2((k_e + C_2)(k_e + C_1) - p_1p_2)^2\lambda_1\lambda_2}{(Z_1 - Z_2 - ((p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2))^2)} \\ &\pm \frac{2(p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2)\sqrt{(Z_1 + Z_2)^2 - 4k^2((k_e + C_2)(k_e + C_1) - p_1p_2)^2\lambda_1\lambda_2}}{(Z_1 - Z_2 - (p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2))^2} \\ &= \left( \frac{(p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2) \pm \sqrt{(Z_1 + Z_2)^2 - 4k^2((k_e + C_2)(k_e + C_1) - p_1p_2)^2\lambda_1\lambda_2}}{(p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2) + Z_2 - Z_1} \right)^2 \\ &= \left( \frac{(p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2) \pm \sqrt{(Z_1 - Z_2)^2 - 4\lambda_1\lambda_2p_1p_2(k_e^2 - p_1p_2)^2}}{(p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2) + Z_2 - Z_1} \right)^2. \end{aligned}$$

We will use one of the two last expressions according to our needs.  $\square$

From now on, when  $(p_1, p_2) \in \mathbb{C}$  (3.1), we define  $(\bar{p}_1, \bar{p}_2) \in \mathbb{R}_+$  such that

$$p_1 = i\sqrt{\frac{\omega^2}{C_s^2} - k_e^2} =: i\bar{p}_1, \quad p_2 = i\sqrt{\frac{\omega^2}{C_p^2} - k_e^2} =: i\bar{p}_2.$$

As in the previous chapter  $\bar{\lambda}_{1,2}$  are defined in a similar manner. We will see that the convergence properties of the algorithm will change depending on the value of  $k_e$  and on the presence of the overlap.

**Theorem 3.1** (Convergence of the non-overlapping Schwarz method with OIC:  $k_e < \frac{\omega}{C_p}$ ). *The non-overlapping Schwarz method with OIC such that  $k_e < \frac{\omega}{C_p}$  converges for  $k \in (0, \frac{\omega}{C_s}) \setminus \{\frac{\omega}{C_p}\}$  and diverges for  $k \in \{\frac{\omega}{C_p}\} \cup [\frac{\omega}{C_s}, \infty)$ .*

*Proof.* We start with a graphical illustration of the theorem in the Figure 3.1.

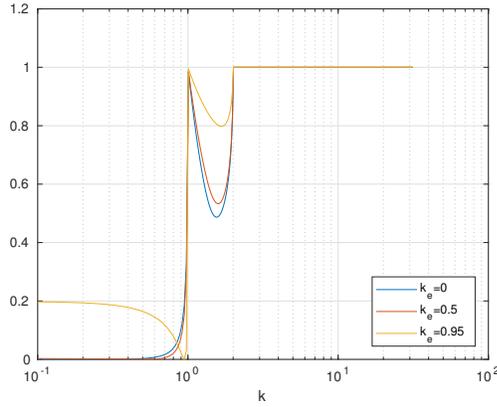


Figure 3.1:  $k_e < \frac{\omega}{C_p}$ : the spectrum of the iteration matrix for the non-overlapping Schwarz method with OIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ .

When  $k_e < \frac{\omega}{C_p}$  we have that  $p_{1,2} \in i\mathbb{R}_+$  and we distinguish the following cases

- **Case 1:** If  $k \in (0, \frac{\omega}{C_p})$  then  $\lambda_{1,2} \in i\mathbb{R}_+$  and we have

$$\sqrt{|r_{\pm}|} = \left| \frac{(-\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2)(k_e^2 + \bar{p}_1 \bar{p}_2) \pm \sqrt{(Z_1 - Z_2)^2 - 4\bar{\lambda}_1 \bar{\lambda}_2 \bar{p}_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2)^2}}{Z_2 - Z_1 - (\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2)(k_e^2 + \bar{p}_1 \bar{p}_2)} \right|,$$

$$< \begin{cases} \left| \frac{(-\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2)(k_e^2 + \bar{p}_1 \bar{p}_2) + \sqrt{(Z_1 - Z_2)^2}}{(\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2)(k_e^2 + \bar{p}_1 \bar{p}_2) - Z_2 + Z_1} \right| & \text{if } \bar{\lambda}_1 \bar{p}_2 > \bar{p}_1 \bar{\lambda}_2, \\ \left| \frac{(-\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2)(k_e^2 + \bar{p}_1 \bar{p}_2) - \sqrt{(Z_1 - Z_2)^2}}{(\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2)(k_e^2 + \bar{p}_1 \bar{p}_2) - Z_2 + Z_1} \right| & \text{if } \bar{\lambda}_1 \bar{p}_2 < \bar{p}_1 \bar{\lambda}_2, \end{cases}$$

which leads to

$$\sqrt{|r_{\pm}|} < \begin{cases} \frac{(\bar{\lambda}_1 \bar{p}_2 - \bar{p}_1 \bar{\lambda}_2) (k_e^2 + \bar{p}_1 \bar{p}_2) + k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right) + \bar{\lambda}_1 \bar{\lambda}_2 \left( (k_e + C_1)^2 + \bar{p}_1 \bar{p}_2 \right)}{(\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2) (k_e^2 + \bar{p}_1 \bar{p}_2) + k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right) + \bar{\lambda}_1 \bar{\lambda}_2 \left( (k_e + C_1)^2 + \bar{p}_1 \bar{p}_2 \right)}, \\ \frac{(\bar{p}_1 \bar{\lambda}_2 - \bar{\lambda}_1 \bar{p}_2) (k_e^2 + \bar{p}_1 \bar{p}_2) + k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right) + \bar{\lambda}_1 \bar{\lambda}_2 \left( (k_e + C_1)^2 + \bar{p}_1 \bar{p}_2 \right)}{(\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2) (k_e^2 + \bar{p}_1 \bar{p}_2) + k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right) + \bar{\lambda}_1 \bar{\lambda}_2 \left( (k_e + C_1)^2 + \bar{p}_1 \bar{p}_2 \right)}. \end{cases}$$

In the two different cases, the right hand side terms are lower than one since  $(\bar{\lambda}_1 \bar{p}_2 - \bar{p}_1 \bar{\lambda}_2)$ ,  $(\bar{p}_1 \bar{\lambda}_2 - \bar{\lambda}_1 \bar{p}_2)$  both being smaller than  $(\bar{p}_1 \bar{\lambda}_2 + \bar{\lambda}_1 \bar{p}_2)$ .

- **Case 2:** If  $k = \frac{\omega}{C_p}$  then  $\lambda_1 \in i\mathbb{R}_+$  and  $\lambda_2 = 0$ , therefore we have

$$\begin{aligned} \sqrt{|r_{\pm}|} &= \left| \frac{\bar{p}_2 \bar{\lambda}_1 (k_e^2 + \bar{p}_1 \bar{p}_2) \pm \sqrt{k^4 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right)^2}}{\bar{p}_2 \bar{\lambda}_1 (k_e^2 + \bar{p}_1 \bar{p}_2) + k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right)} \right|, \\ &= \begin{cases} \left| \frac{\bar{p}_2 \bar{\lambda}_1 (k_e^2 + \bar{p}_1 \bar{p}_2) + k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right)}{\bar{p}_2 \bar{\lambda}_1 (k_e^2 + \bar{p}_1 \bar{p}_2) + k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right)} \right| = 1, \\ \left| \frac{\bar{p}_2 \bar{\lambda}_1 (k_e^2 + \bar{p}_1 \bar{p}_2) - k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right)}{\bar{p}_2 \bar{\lambda}_1 (k_e^2 + \bar{p}_1 \bar{p}_2) + k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right)} \right| < 1. \end{cases} \end{aligned}$$

- **Case 3:** If  $k \in \left( \frac{\omega}{C_s}, \frac{\omega}{C_p} \right)$  then  $\lambda_1 \in i\mathbb{R}_+$  and we have

$$\begin{aligned} \sqrt{|r_{\pm}|} &= \frac{1}{\left| (i\bar{p}_1 \lambda_2 - \bar{\lambda}_1 \bar{p}_2) (k_e^2 + \bar{p}_1 \bar{p}_2) + Z_2 - Z_1 \right|} \times \left| (i\bar{p}_1 \lambda_2 + \bar{\lambda}_1 \bar{p}_2) (k_e^2 + \bar{p}_1 \bar{p}_2) \right. \\ &\quad \left. \pm \sqrt{(Z_1 + Z_2)^2 - 4ik^2 \bar{\lambda}_1 \lambda_2 \left( (k_e + C_1)(k_e + C_2) + \bar{p}_1 \bar{p}_2 \right)^2} \right| \\ &< \left| \frac{(i\bar{p}_1 \lambda_2 + \bar{\lambda}_1 \bar{p}_2) (k_e^2 + \bar{p}_1 \bar{p}_2) + \sqrt{(Z_1 + Z_2)^2}}{(i\bar{p}_1 \lambda_2 - \bar{\lambda}_1 \bar{p}_2) (k_e^2 + \bar{p}_1 \bar{p}_2) + Z_2 - Z_1} \right| = 1, \end{aligned}$$

since  $\Im \left( (Z_1 + Z_2)^2 \right) = k^2 \left( (k_e + C_2)^2 + \bar{p}_1 \bar{p}_2 \right) \bar{\lambda}_1 \lambda_2 \left( (k_e + C_1)^2 + \bar{p}_1 \bar{p}_2 \right) > 0$ .

- **Case 4:** If  $k = \frac{\omega}{C_s}$  then  $\lambda_1 = 0$ , so we have

$$\sqrt{|r_{\pm}|} = \left| \frac{i\bar{p}_1\lambda_2(k_e^2 + \bar{p}_1\bar{p}_2) \pm \sqrt{k^4((k_e + C_2)^2 + \bar{p}_1\bar{p}_2)^2}}{-i\bar{p}_1\lambda_2(k_e^2 + \bar{p}_1\bar{p}_2) + k^2((k_e + C_2)^2 + \bar{p}_1\bar{p}_2)} \right| = 1.$$

- **Case 5:** If  $k = \left(\frac{\omega}{C_s}, \infty\right)$  then we have

$$\begin{aligned} \sqrt{|r_{\pm}|} &= \left| \frac{i(\bar{p}_1\lambda_2 - \lambda_1\bar{p}_2)(k_e^2 + \bar{p}_1\bar{p}_2) \pm \sqrt{(Z_1 - Z_2)^2 + 4\lambda_1\lambda_2\bar{p}_1\bar{p}_2(k_e^2 + \bar{p}_1\bar{p}_2)^2}}{i(\bar{p}_1\lambda_2 + \lambda_1\bar{p}_2)(k_e^2 + \bar{p}_1\bar{p}_2) + Z_2 - Z_1} \right| \\ &= \left| \frac{(\bar{p}_1\lambda_2 - \lambda_1\bar{p}_2)^2(k_e^2 + \bar{p}_1\bar{p}_2)^2 + (Z_1 - Z_2)^2 + 4\lambda_1\lambda_2\bar{p}_1\bar{p}_2(k_e^2 + \bar{p}_1\bar{p}_2)^2}{(\bar{p}_1\lambda_2 + \lambda_1\bar{p}_2)^2(k_e^2 + \bar{p}_1\bar{p}_2)^2 + (Z_2 - Z_1)^2} \right| = 1. \end{aligned}$$

since  $Z_1, Z_2 \in \mathbb{R}$ . □

**Theorem 3.2** (Convergence of the Overlapping Schwarz method with OIC:  $k_e < \frac{\omega}{C_p}$ ). *The overlapping Schwarz method with a small overlap  $\delta$  and with OIC such that  $k_e < \frac{\omega}{C_p}$  converges for all values  $k \in \left[0, \frac{\omega}{C_p}\right) \cup \left(\frac{\omega}{C_p}, \frac{\omega}{C_s}\right) \cup (k^*, \infty)$  but diverges for the following  $k \in \left\{\frac{\omega}{C_p}\right\} \cup \left[\frac{\omega}{C_s}, k^*\right]$  where  $k^*(\omega, C_p, C_s, \delta) \in \left(\frac{\omega}{C_s}, \infty\right)$ .*

*Proof.* We start with an illustration of the theorem in Figure 3.2.

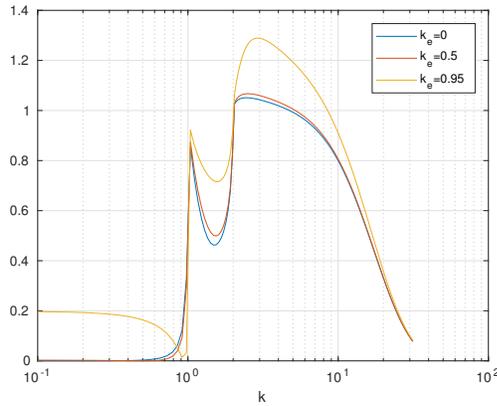


Figure 3.2: For  $k_e < \frac{\omega}{C_p}$ , spectrum of the iteration matrix from the overlapping Schwarz method with OIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ .

Again we have  $p_{1,2} \in i\mathbb{R}_+$ . In the case of a small  $\delta$ , we can use a series expansion as in

(2.48) of the modulus of the eigenvalues of the iteration matrix,

$$|r_{\pm}|^2 = (R_{1\pm}^2 + I_{1\pm}^2) + 2\delta(R_{1\pm}R_{2\pm} + I_{1\pm}I_{2\pm}) + \mathcal{O}(\delta^2).$$

We can distinguish similar cases as previously:

- **Case 1 and 3:** If  $k \in \left(0, \frac{\omega}{C_s}\right) \setminus \left\{\frac{\omega}{C_p}\right\}$ , we proved in the previous theorem that the convergence factor in the case without overlap, which corresponds to  $(R_{1\pm}^2 + I_{1\pm}^2)$ , is strictly lower than one. Therefore for a small enough  $\delta$ , the conclusion still holds

$$|r_{\pm}|^2 = \underbrace{R_{1\pm}^2 + I_{1\pm}^2}_{<1} + \mathcal{O}(\delta) < 1.$$

- **Case 2:** If  $k = \frac{\omega}{C_p}$  then  $\lambda_1 \in i\mathbb{R}_+$ ,  $\lambda_2 = 0$ ,  $Z_2 = 0$ ,  $b_{11} = 1$ ,  $b_{12} = 0$ , and

$$\begin{aligned} \sqrt{|r_{\pm}|} &= \left| \frac{(Z_1 + \bar{\lambda}_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2))^2 e^{-2i\bar{\lambda}_1 \delta} + (Z_1 - \bar{\lambda}_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2))^2}{2 (Z_1 + \bar{\lambda}_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2))^2} \right. \\ &\quad \left. \pm \frac{\sqrt{(Z_1 + \bar{\lambda}_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2))^2 e^{-2i\bar{\lambda}_1 \delta} - (Z_1 - \bar{\lambda}_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2))^2}}{2 (Z_1 + \bar{\lambda}_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2))^2} \right| \\ &= \begin{cases} |e^{-2i\bar{\lambda}_1 \delta}| = 1, \\ \left| \frac{Z_1 - \lambda_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2)}{Z_1 + \lambda_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2)} \right| = \left| \frac{k^2((k_e + C_2)^2 + \bar{p}_1 \bar{p}_2) - \lambda_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2)}{k^2((k_e + C_2)^2 + \bar{p}_1 \bar{p}_2) + \lambda_1 \bar{p}_2 (k_e^2 + \bar{p}_1 \bar{p}_2)} \right| < 1. \end{cases} \end{aligned}$$

- **Case 4:** If  $k = \frac{\omega}{C_s}$  then  $\lambda_1 = 0$  and  $Z_2 = 0$ ,  $b_{21} = 0$ ,  $b_{22} = 1$ . Then

$$\begin{aligned} \sqrt{|r_{\pm}|} &= \left| \frac{(Z_1 - i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2))^2 e^{-2\lambda_2 \delta} + (Z_1 + i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2))^2}{2 (Z_1 - i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2))^2} \right. \\ &\quad \left. \pm \frac{(Z_1 - i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2)) e^{-2\lambda_2 \delta} - (Z_1 + i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2))}{2 (Z_1 - i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2))^2} \right. \\ &\quad \left. \times \frac{\sqrt{((Z_1 - i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2)) e^{-2\lambda_2 \delta} + Z_1 + i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2))^2}}{2 (Z_1 - i\lambda_2 \bar{p}_1 (k_e^2 + \bar{p}_1 \bar{p}_2))^2} \right| \end{aligned}$$

and we end up this time with

$$\sqrt{|r_{\pm}|} = \begin{cases} |e^{-2\lambda_2\delta}| < 1, \\ \left| \frac{Z_1 + i\lambda_2\bar{p}_1(k_e^2 + \bar{p}_1\bar{p}_2)}{Z_1 - i\lambda_2\bar{p}_1(k_e^2 + \bar{p}_1\bar{p}_2)} \right| = \left| \frac{k^2((k_e + C_2)^2 + \bar{p}_1\bar{p}_2) + i\lambda_2\bar{p}_1(k_e^2 + \bar{p}_1\bar{p}_2)}{k^2((k_e + C_2)^2 + \bar{p}_1\bar{p}_2) - i\lambda_2\bar{p}_1(k_e^2 + \bar{p}_1\bar{p}_2)} \right| = 1. \end{cases}$$

• **Case 5:** If  $k \in \left(\frac{\omega}{C_s}, \infty\right)$ , then we write the series expansion (2.48) of  $r_{\pm}$  and obtain  $R_{1\pm}^2 + I_{1\pm}^2 = 1$  for the main term (as seen in the previous theorem) and

$$\begin{aligned} R_{1\pm}R_{2\pm} + I_{1\pm}I_{2\pm} = & \\ & - \frac{\left((\lambda_2\bar{p}_1 - \lambda_1\bar{p}_2)^2(k_e^2 + \bar{p}_1\bar{p}_2)^2 + (Z_1 + Z_2)^2 - \lambda_2\lambda_14k^2((k_e + C_2)(k_e + C_1) + \bar{p}_1\bar{p}_2)^2\right)^2}{\mathcal{D}\sqrt{(Z_1 + Z_2)^2 - \lambda_2\lambda_14k^2((k_e + C_2)(k_e + C_1) + \bar{p}_1\bar{p}_2)^2}} \\ & \times \underbrace{\left(\sqrt{(Z_1 - Z_2)^2 + 4\lambda_2\lambda_1\bar{p}_1\bar{p}_2(k_e^2 + \bar{p}_1\bar{p}_2)^2}(\lambda_1 + \lambda_2) \pm \underbrace{(\lambda_1 - \lambda_2)}_{<0} \underbrace{(Z_1 + Z_2)}_{>0}\right)}_{g(k)} \end{aligned}$$

where

$$\mathcal{D} := \left| Z_1 - Z_2 - i(\lambda_2\bar{p}_1 + \lambda_1\bar{p}_2)(k_e^2 + \bar{p}_1\bar{p}_2) \right|,$$

therefore  $R_{1-}R_{2-} + I_{1-}I_{2-} < 0$  and  $r_- \approx 1 + R_{1-}R_{2-} + I_{1-}I_{2-} < 1$ .

We now study the sign of  $g(k)$  in the case of  $R_{1+}R_{2+} + I_{1+}I_{2+}$  and one can show by asymptotic arguments that  $g(k) < 0$  in a neighbourhood of  $\frac{\omega}{C_s}$  which means that  $R_{1+}R_{2+} + I_{1+}I_{2+} > 0$  and  $r_+ \approx 1 + R_{1+}R_{2+} + I_{1+}I_{2+} > 1$ .

Then by using the fact that the overlap makes the convergence factor vanish at infinity,

$$\lim_{k \rightarrow \infty} \rho_E(k, \omega, C_p, C_s, \delta) = 0$$

we can conclude there will be a small interval on which the algorithm is not convergent.  $\square$

We summarise the previous result as follows

**Remark 3.1.** We notice that  $\rho_E \xrightarrow{k_e \rightarrow 0} \rho_{T_0}$  ( $\Leftrightarrow \hat{S}_j^E \xrightarrow{k_e \rightarrow 0} \hat{S}_j^{T_0}$ ), and the curves show that for almost all  $k \in \mathbb{R}_+$ ,  $\rho_E \leq \rho_H$  the Schwarz method with zeroth order TTC is uniformly better than that with OIC for  $k_e < \frac{\omega}{C_p}$ .

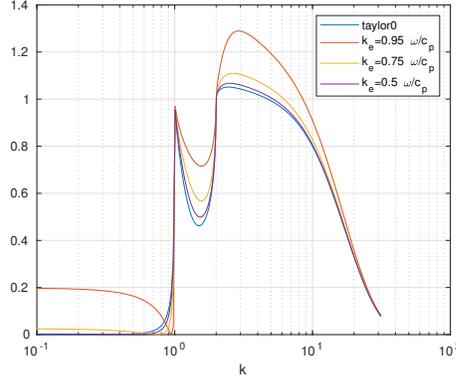


Figure 3.3: Comparison of the overlapping Schwarz method with zeroth order TTC and OIC for  $k_e < \frac{\omega}{C_p}$  with  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ ,  $\delta = \frac{1}{10}$ .

We move on now to the case where  $k_e = \frac{\omega}{C_p}$  or  $k_e = \frac{\omega}{C_s}$ .

**Theorem 3.3** (Convergence of the non-overlapping Schwarz method with OIC:  $k_e = \frac{\omega}{C_p}$  or  $k_e = \frac{\omega}{C_s}$ ). *The non-overlapping Schwarz method with OIC such that  $k_e \in \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$  is always divergent except when  $k \in \left( \frac{\omega}{C_p}, \frac{\omega}{C_s} \right)$*

*Proof.* Under the specific hypothesis of the theorem we have

$$k_e = \frac{\omega}{C_p} =: k_1 \Leftrightarrow [p_2 = 0, p_1 = i\bar{p}_1] \quad \& \quad k_e = \frac{\omega}{C_s} =: k_2 \Leftrightarrow [p_1 = 0, p_2 \in \mathbb{R}_+^*],$$

therefore (3.4) gives us in that first case  $r_{\pm}^{k_1}$  (and similarly  $r_{\pm}^{k_2}$ ) as

$$\sqrt{r_{\pm}^{k_1}} = \begin{cases} \left| \frac{i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2 + k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 - \lambda_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2}{-k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2 + i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2} \right| =: r_{+}^{k_1}, \\ \left| \frac{i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2 - k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2}{-k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2 + i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2} \right| =: r_{-}^{k_1} = 1, \end{cases}$$

$$\sqrt{r_{\pm}^{k_2}} = \begin{cases} \left| \frac{-\frac{\omega^2}{C_s^2} p_2 \lambda_1 + k^2 \left( \frac{\omega}{C_s} + C_2^{k_2} \right)^2 - \lambda_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2}{-k^2 \left( \frac{\omega}{C_s} + C_2^{k_2} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2 + \frac{\omega^2}{C_s^2} p_2 \lambda_1} \right| =: r_{+}^{k_2} = 1, \\ \left| \frac{-\frac{\omega^2}{C_s^2} p_2 \lambda_1 - k^2 \left( \frac{\omega}{C_s} + C_2^{k_2} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2}{-k^2 \left( \frac{\omega}{C_s} + C_2^{k_2} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2 + \frac{\omega^2}{C_s^2} p_2 \lambda_1} \right| =: r_{-}^{k_2}, \end{cases}$$

where

$$C_1^{k_1} = 2 \left( \frac{C_s}{C_p} \right)^2 \left( k - \frac{\omega}{C_p} \right), \quad C_2^{k_1} = \left( \frac{C_s}{\sqrt{k}C_p} \right)^2 \left( k^2 - \frac{2k\omega}{C_p} + \lambda_1^2 \right),$$

and

$$C_1^{k_2} = 2 \left( k - \frac{\omega}{C_s} \right), \quad C_2^{k_2} = \frac{1}{k} \left( k^2 - \frac{2k\omega}{C_s} + \lambda_1^2 \right).$$

We notice  $(C_1^{k_1}, C_2^{k_1}, C_1^{k_2}, C_2^{k_2})$  will remain real and distinguish again several cases to study  $r_+^{k_1}$  and  $r_-^{k_2}$  according the values of  $k$  as follows

- If  $k \in \left( 0, \frac{\omega}{C_p} \right)$  then  $\lambda_{1,2} \in i\mathbb{R}_+$ . So we get

$$\begin{cases} r_+^{k_1} = \left| \frac{-\frac{\omega^2}{C_p^2} \bar{p}_1 \bar{\lambda}_2 + k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 + \bar{\lambda}_1 \bar{\lambda}_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2}{-k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 - \bar{\lambda}_1 \bar{\lambda}_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2 - \frac{\omega^2}{C_p^2} \bar{p}_1 \bar{\lambda}_2} \right| < 1, \\ r_-^{k_2} = \left| \frac{-i \frac{\omega^2}{C_s^2} p_2 \bar{\lambda}_1 - k^2 \left( \frac{\omega}{C_s} + C_2^{k_2} \right)^2 - \bar{\lambda}_1 \bar{\lambda}_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2}{-k^2 \left( \frac{\omega}{C_s} + C_2^{k_2} \right)^2 - \bar{\lambda}_1 \bar{\lambda}_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2 + i \frac{\omega^2}{C_s^2} p_2 \bar{\lambda}_1} \right| = 1. \end{cases}$$

- If  $k = \frac{\omega}{C_p}$  then  $\lambda_1 \in i\mathbb{R}_+$  and  $C_1^{k_1} = \lambda_2 = 0$  and we get

$$\lim_{k \rightarrow k_1^-} |r_+^{k_1}| = \lim_{k \rightarrow k_1^+} |r_+^{k_1}| = 0 \quad \text{and} \quad r_-^{k_2} = \left| \frac{-i \frac{\omega^6}{C_s^2} p_2 \bar{\lambda}_1 - k^2 \left( \frac{\omega^3}{C_s} + \bar{C}_2^{k_2} \right)^2}{-k^2 \left( \frac{\omega^3}{C_s} + \bar{C}_2^{k_2} \right)^2 + i \frac{\omega^6}{C_s^2} p_2 \bar{\lambda}_1} \right| = 1.$$

- If  $k \in \left( \frac{\omega}{C_p}, \frac{\omega}{C_s} \right)$  then  $\lambda_1 \in i\mathbb{R}_+$ . So we get

$$\begin{cases} r_+^{k_1} = \left| \frac{i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2 + k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 - i \bar{\lambda}_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2}{-k^2 \left( \frac{\omega}{C_p} + \bar{C}_2^{k_1} \right)^2 + i \bar{\lambda}_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2 + i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2} \right| =: \left| \frac{n_1}{d_1} \right| < 1, \\ r_-^{k_2} = \left| \frac{-i \frac{\omega^2}{C_s^2} p_2 \bar{\lambda}_1 - k^2 \left( \frac{\omega}{C_s} + \bar{C}_2^{k_2} \right)^2 + i \bar{\lambda}_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2}{-k^2 \left( \frac{\omega}{C_s} + \bar{C}_2^{k_2} \right)^2 + i \bar{\lambda}_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2 + i \frac{\omega^2}{C_s^2} p_2 \bar{\lambda}_1} \right| =: \left| \frac{n_2}{d_2} \right| < 1, \end{cases}$$

because all variables being real and positive, we have the relations

$$\Re(n_1) = -\Re(d_1), \quad \Re(n_2) = \Re(d_2),$$

and

$$\left\{ \begin{array}{l} \Im(n_1) = \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2 - \bar{\lambda}_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2 < \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2 + \bar{\lambda}_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2 \\ \hspace{15em} = \Im(d_1), \\ \Im(n_2) = -\frac{\omega^2}{C_s^2} p_2 \bar{\lambda}_1 + \bar{\lambda}_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2 < \frac{\omega^2}{C_s^2} p_2 \bar{\lambda}_1 + \bar{\lambda}_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2 \\ \hspace{15em} = \Im(d_2). \end{array} \right.$$

- If  $k = \frac{\omega}{C_s}$  then  $\lambda_2 \in \mathbb{R}_+$  and  $\lambda_1 = 0$ , so we get

$$r_+^{k_1} = \left| \frac{i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2 + k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2}{-k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 + i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2} \right| = 1 \quad \text{and} \quad \lim_{k \rightarrow k_2^-} |r_-^{k_2}| = \lim_{k \rightarrow k_2^+} |r_-^{k_2}| = 0.$$

- If  $k \in \left( \frac{\omega}{C_s}, \infty \right)$  then  $\lambda_{1,2} \in \mathbb{R}_+$  and we get

$$\left\{ \begin{array}{l} r_+^{k_1} = \left| \frac{i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2 + k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 - \lambda_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2}{-k^2 \left( \frac{\omega}{C_p} + C_2^{k_1} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_p} + C_1^{k_1} \right)^2 + i \frac{\omega^2}{C_p^2} \bar{p}_1 \lambda_2} \right| = 1, \\ r_-^{k_2} = \left| \frac{-\frac{\omega^2}{C_s^2} p_2 \lambda_1 - k^2 \left( \frac{\omega}{C_s} + C_2^{k_2} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2}{-k^2 \left( \frac{\omega}{C_s} + C_2^{k_2} \right)^2 + \lambda_1 \lambda_2 \left( \frac{\omega}{C_s} + C_1^{k_2} \right)^2 + \frac{\omega^2}{C_s^2} p_2 \lambda_1} \right| < 1. \end{array} \right.$$

□

**Remark 3.2.** In overlapping case, when  $k_e \in \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$  the behaviour of the algorithm is illustrated in Figure 3.4.

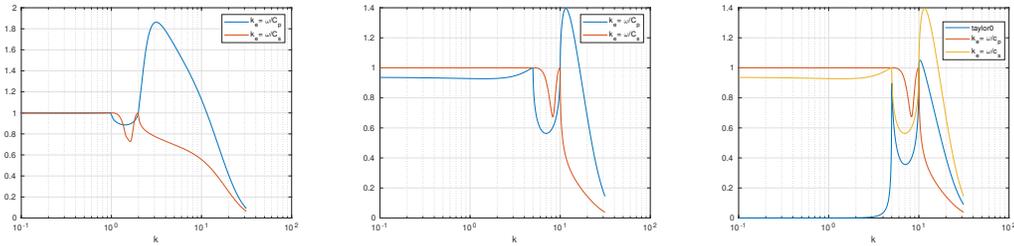


Figure 3.4:  $k_e \in \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$ : spectrum of the iteration matrix from the overlapping Schwarz method with OIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\delta = \frac{1}{10}$ . Left:  $\omega = 1$ . Middle:  $\omega = 5$ , Right: comparison with TTC.

Moreover, one can see that the Schwarz algorithm with zeroth order TTC is better.

We investigate now the case  $k_e \in \left(\frac{\omega}{C_s}, \infty\right)$  and we get the result

**Theorem 3.4** (Convergence of the non-overlapping Schwarz method with OIC:  $k_e > \frac{\omega}{C_s}$ ).  
The non-overlapping Schwarz method with OIC such as  $k_e > \frac{\omega}{C_s}$  diverges for  $k \in \left(0, \frac{\omega}{C_s}\right]$  and converges for  $k \in \left(\frac{\omega}{C_s}, \infty\right)$ .

*Proof.* We start with a illustration of the theorem in the Figure 3.5 and distinguish as usual five cases.

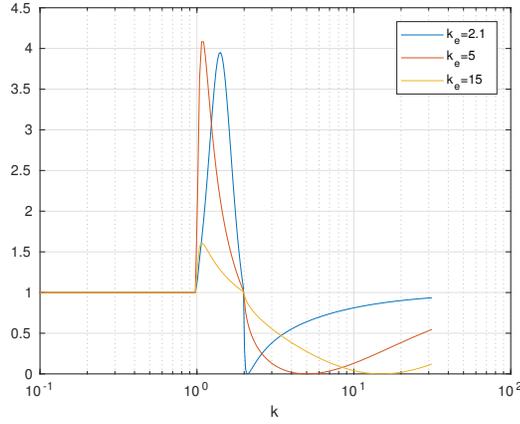


Figure 3.5:  $k_e > \frac{\omega}{C_p}$ : the spectrum of the iteration matrix from the non-overlapping Schwarz method with OIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ .

- **Case 1:** If  $k \in \left(0, \frac{\omega}{C_p}\right)$  then  $\lambda_{1,2} \in i\mathbb{R}_+$  and  $Z_1, Z_2 \in \mathbb{R}$  and (3.4) gives

$$\sqrt{|r_{\pm}|} = \left| \frac{i(p_1\bar{\lambda}_2 - \bar{\lambda}_1 p_2)(k_e^2 - p_1 p_2) \pm \sqrt{(Z_1 - Z_2)^2 + 4\bar{\lambda}_2 \bar{\lambda}_1 p_1 p_2 (k_e^2 - p_1 p_2)^2}}{i(p_1\bar{\lambda}_2 + \bar{\lambda}_1 p_2)(k_e^2 - p_1 p_2) + Z_2 - Z_1} \right| = 1.$$

- **Case 2:** If  $k = \frac{\omega}{C_p}$  then  $\lambda_1 \in i\mathbb{R}_+$  and  $\lambda_2 = 0$ ,  $Z_2 = 0$ ,  $C_2 \in \mathbb{R}$  therefore (3.4) gives

$$\sqrt{|r_{\pm}|} = \left| \frac{-ip_2 \bar{\lambda}_1 (k_e^2 - p_1 p_2) \pm \sqrt{k^4 \left( (k_e + C_2)^2 - p_1 p_2 \right)^2}}{ip_2 \bar{\lambda}_1 (k_e^2 - p_1 p_2) - k^2 \left( (k_e + C_2)^2 - p_1 p_2 \right)} \right| = 1.$$

- **Case 3:** If  $k \in \left(\frac{\omega}{C_p}, \frac{\omega}{C_s}\right)$  then  $\lambda_1 \in i\mathbb{R}_+$ , and we get from (3.4)

$$\sqrt{|r_{\pm}|} = \left| \frac{(p_1\lambda_2 - i\bar{\lambda}_1 p_2)(k_e^2 - p_1 p_2) \pm \sqrt{(Z_1 - Z_2)^2 - 4i\lambda_2 \bar{\lambda}_1 p_1 p_2 (k_e^2 - p_1 p_2)^2}}{(p_1\lambda_2 + i\bar{\lambda}_1 p_2)(k_e^2 - p_1 p_2) + Z_2 - Z_1} \right|,$$

and let us denote

$$r_+^* := \left| \frac{(p_1\lambda_2 - i\bar{\lambda}_1 p_2)(k_e^2 - p_1 p_2) + \sqrt{(Z_1 - Z_2)^2}}{(p_1\lambda_2 + i\bar{\lambda}_1 p_2)(k_e^2 - p_1 p_2) + Z_2 - Z_1} \right|.$$

We notice that in the numerator of  $r_+^*$ , the first term has a positive real part and negative imaginary part. Moreover under the square root we have

$$\Im((Z_1 - Z_2)^2) = -2\bar{\lambda}_1 \lambda_2 k^2 \underbrace{\left((k_e + C_2)^2 - p_1 p_2\right)}_{<0} \underbrace{\left((k_e + C_1)^2 - p_1 p_2\right)}_{<0} < 0,$$

therefore no matter the value of  $\Re((Z_1 - Z_2)^2)$ , we have  $r_+^* < \sqrt{|r_+|}$ . We want to know if  $|r_+^*| > 1$ . We notice that  $\Re(Z_1 - Z_2) = k^2 \left((k_e + C_2)^2 - p_1 p_2\right) < 0$  therefore

$$r_+^* = \left| \frac{(p_1\lambda_2 - i\bar{\lambda}_1 p_2)(k_e^2 - p_1 p_2) - (Z_1 - Z_2)}{(p_1\lambda_2 + i\bar{\lambda}_1 p_2)(k_e^2 - p_1 p_2) + Z_2 - Z_1} \right| =: \left| \frac{n_+}{d_+} \right|.$$

We get  $\Re(n_+ - d_+) = 0$  and  $\Im(Z_2 - Z_1) = \bar{\lambda}_1 \lambda_2 \left((k_e + C_1)^2 - p_1 p_2\right) < 0$  and  $|\Im(n_+)| > |\Im(d_+)|$ . We can conclude that  $|r_+^*| > 1$ , which means there is always at least one eigenvalue whose lower bound is bigger than one.

- **Case 4:** If  $k = \frac{\omega}{C_s}$  then  $\lambda_1 = 0$ ,  $Z_2 = 0$  and  $C_2 \in \mathbb{R}$ , we have from (3.4)

$$\begin{aligned} \sqrt{|r_{\pm}|} &= \left| \frac{p_1\lambda_2(k_e^2 - p_1 p_2) \pm \frac{\omega^2}{C_s^2} \sqrt{\left((k_e + C_2)^2 - p_1 p_2\right)^2}}{p_1\lambda_2(k_e^2 - p_1 p_2) - \frac{\omega^2}{C_s^2} \left((k_e + C_2)^2 - p_1 p_2\right)} \right| \\ &= \begin{cases} \left| \frac{p_1\lambda_2(k_e^2 - p_1 p_2) + \frac{\omega^2}{C_s^2} \left((k_e + C_2)^2 - p_1 p_2\right)}{p_1\lambda_2(k_e^2 - p_1 p_2) - \frac{\omega^2}{C_s^2} \left((k_e + C_2)^2 - p_1 p_2\right)} \right| = \sqrt{|r_+|}, \\ \left| \frac{p_1\lambda_2(k_e^2 - p_1 p_2) - \frac{\omega^2}{C_s^2} \left((k_e + C_2)^2 - p_1 p_2\right)}{p_1\lambda_2(k_e^2 - p_1 p_2) - \frac{\omega^2}{C_s^2} \left((k_e + C_2)^2 - p_1 p_2\right)} \right| = 1. \end{cases} \end{aligned}$$

We denote

$$\left(\tilde{Z}_1\right)_{k_e, C_p}(C_s) = (k_e + C_2)^2 - p_1 p_2 \in \mathcal{C}^1 \left[0, \frac{C_p}{\sqrt{2}}\right].$$

A quick computation shows  $\#C_s^* / \left(\tilde{Z}_1\right)_{k_e, C_p}(C_s^*) = 0$ , and we see numerically there is at least one negative value, so  $\tilde{Z}_1 < 0$  and then  $\sqrt{|r_+|} < 1$ .

- **Case 5:** If  $k \in \left(\frac{\omega}{C_s}, \infty\right)$  then we get from (3.4)

$$\sqrt{|r_{\pm}|} = \left| \frac{(p_1 \lambda_2 - \lambda_1 p_2) (k_e^2 - p_1 p_2) \pm \sqrt{(Z_1 - Z_2)^2 - 4 \lambda_2 \lambda_1 p_1 p_2 (k_e^2 - p_1 p_2)^2}}{(p_1 \lambda_2 + \lambda_1 p_2) (k_e^2 - p_1 p_2) + Z_2 - Z_1} \right|.$$

We have  $Z_1, Z_2 \in \mathbb{R}$  and suppose  $Z_1 < Z_2$ , (which seems to be the case according to the figure 3.6).

Since we cannot prove analytically that  $Z_2 > Z_1$ , we sketched in Figure 3.6 and Figure 3.7, the difference  $Z_2 - Z_1$  for  $k, k_e \in \left(\frac{\omega}{C_s} = k_{\min}, \frac{\pi}{h} = k_{\max}\right)$ , which represent the set of all the possible  $k, k_e$  we could have in our case, and we take  $h = 1/10$ . We do it in two different cases including the one with oscillatory solutions ( $C_p = 5, C_s = 1, \omega = 10$ ), and we will significantly increase the number of grid points  $\frac{k_{\max} - k_{\min}}{N}$  on both axes  $k$  and  $k_e$  simultaneously by taking  $N = 100$  then  $N = 1000$  in order to have an accurate enough representation of this difference.

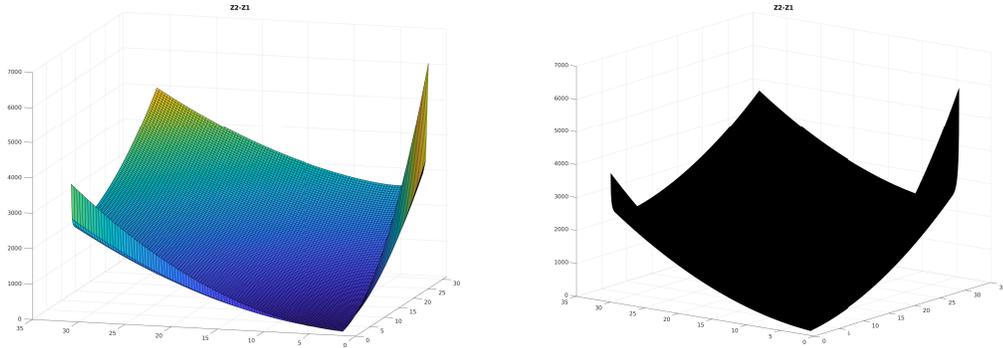


Figure 3.6:  $Z_2 - Z_1$  with  $C_p = 1, C_s = \frac{1}{2}, \omega = 1$ . Left:  $N=100$ . Right:  $N=1000$ .

and we notice that is always bigger than or equal to 0.

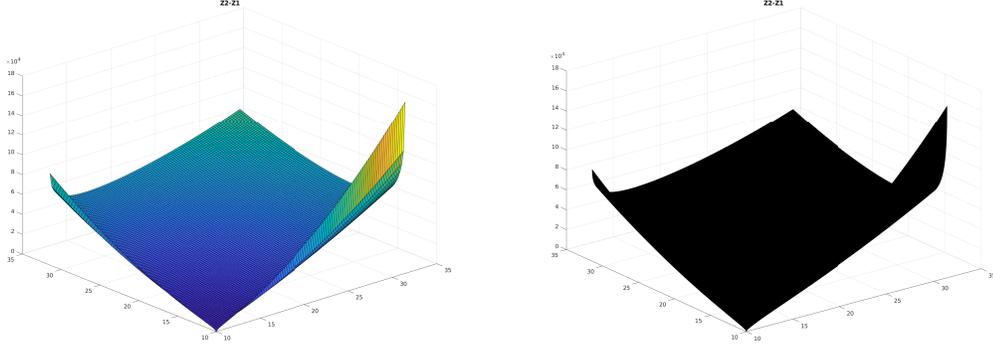


Figure 3.7:  $Z_2 - Z_1$  with  $C_p = 5$ ,  $C_s = 1$ ,  $\omega = 10$ . Left:  $N=100$ . Right:  $N=1000$ .

If  $(Z_1 - Z_2)^2 < 4\lambda_2\lambda_1p_1p_2(k_e^2 - p_1p_2)^2$ , we obtain

$$\begin{aligned} |r_{\pm}| &= \left| \frac{((p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2))^2 - (Z_1 - Z_2)^2 + 4\lambda_2\lambda_1p_1p_2(k_e^2 - p_1p_2)^2}{((p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2) + Z_2 - Z_1)^2} \right|^2 \\ &= \left| \frac{[(p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2)]^2 - [Z_2 - Z_1]^2}{[(p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2)] + [Z_2 - Z_1]^2} \right|^2 < 1, \end{aligned}$$

since  $Z_2 > Z_1$ .

If not we have

$$\begin{aligned} |r_{\pm}| &= \left| \frac{(p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2) \pm \sqrt{(Z_1 - Z_2)^2 - 4\lambda_2\lambda_1p_1p_2(k_e^2 - p_1p_2)^2}}{(p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2) + Z_2 - Z_1} \right|^2 \\ &\leq \left| \frac{(p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2) \pm \sqrt{(Z_1 - Z_2)^2}}{(p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2) + Z_2 - Z_1} \right|^2 \\ &= \left| \frac{(p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2) \pm (Z_2 - Z_1)}{(p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2) + Z_2 - Z_1} \right|^2 =: (r_{\pm}^*)^2, \end{aligned}$$

where  $r_+^*$  (resp.  $r_-^*$ ) is the maximum when  $(p_1\lambda_2 > \lambda_1p_2)$  (resp.  $(p_1\lambda_2 < \lambda_1p_2)$ ),  $(k_e^2 - p_1p_2)$  being positive. In the first case, we have as well  $(p_1\lambda_2 + \lambda_1p_2) > (p_1\lambda_2 - \lambda_1p_2)$  and by supposition  $Z_2 > Z_1$ , and therefore  $|r_+^*| < 1$ . In the second case

$$|r_-^*| = \left| \frac{(p_1\lambda_2 - \lambda_1p_2)(k_e^2 - p_1p_2) - (Z_2 - Z_1)}{(p_1\lambda_2 + \lambda_1p_2)(k_e^2 - p_1p_2) + Z_2 - Z_1} \right|$$

and same as before, if  $(p_2\lambda_1 - \lambda_2p_1) < (p_1\lambda_2 + \lambda_1p_2)$  &  $Z_2 > Z_1$ , then  $|r_-^*| < 1$ .  $\square$

**Theorem 3.5** (Convergence of the overlapping Schwarz method with OIC:  $k_e > \frac{\omega}{C_s}$ ).  
 The overlapping Schwarz method with OIC such as  $k_e > \frac{\omega}{C_s}$  diverges for  $k \in \left(0, \frac{\omega}{C_s}\right]$   
 and converges for  $k \in \left(\frac{\omega}{C_s}, \infty\right)$ .

The conclusions are very similar as in the non-overlapping case and the proof is mainly based on the series with respect to  $\delta$  in which we use the results of the Theorem 3.4.

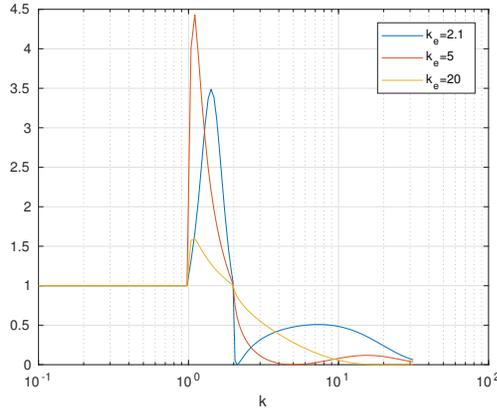


Figure 3.8:  $k_e > \frac{\omega}{C_p}$ : the spectrum of the iteration matrix from the overlapping Schwarz method with OIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ ,  $\delta = \frac{1}{10}$ .

A graphic illustration of the theorem is given in Figure 3.8.

**Remark 3.3.** Figure 3.9 shows that classical Schwarz method is better than with OIC for  $k_e > \frac{\omega}{C_s}$  for low and "middle" frequencies and less good for higher ones.

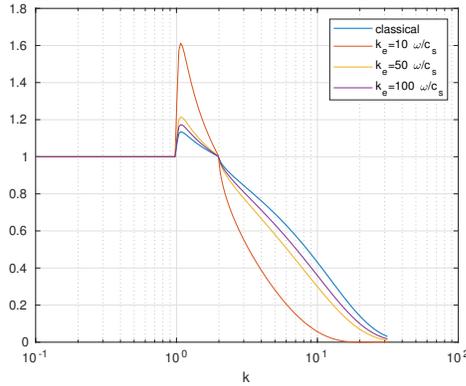


Figure 3.9: Comparison of the classical and the OIC Schwarz method for  $k_e > \frac{\omega}{C_s}$  with  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ ,  $\delta = \frac{1}{10}$ .

Indeed, we already know that  $\rho_{T_0} < 1$  for  $k \in \left[0, \frac{\omega}{C_s}\right) \setminus \left\{\frac{\omega}{C_p}\right\}$  from Theorem 2.4. If  $k \in \left(0, \frac{\omega}{C_p}\right)$ , we know that  $\rho_E = 1$ , and for  $k \in \left(\frac{\omega}{C_p}, \frac{\omega}{C_s}\right)$ , we have  $\rho_{T_0} < 1 < \rho_E$  from Theorem 3.5. Moreover, we notice that if we take the general form of the eigenvalues from Lemma 3.1

$$X = e^{-\lambda_1 \delta} b_{11} - e^{-\lambda_2 \delta} b_{22}, \quad Y = e^{-(\lambda_1 + \lambda_2) \delta} (b_{11} b_{22} - b_{12} b_{21}),$$

where  $b_{ij}$  are given by (3.3), we get

$$\begin{cases} b_{11} \xrightarrow{k_e \rightarrow \infty} \frac{(4k^2 C_s^4 + 4C_s^4 \lambda_1 \lambda_2) k_e^6}{(4k^2 C_s^4 - 4C_s^4 \lambda_1 \lambda_2) k_e^6} = \frac{k^2 + \lambda_1 \lambda_2}{k^2 - \lambda_1 \lambda_2} & \& b_{22} \xrightarrow{k_e \rightarrow \infty} \frac{k^2 + \lambda_1 \lambda_2}{k^2 - \lambda_1 \lambda_2}, \\ b_{11} b_{22} - b_{21} b_{12} \xrightarrow{k_e \rightarrow \infty} \frac{((4k^2 C_s^4 + 4C_s^4 \lambda_1 \lambda_2)^2 - 64 \lambda_2 C_s^8 k^2 \lambda_1) k_e^{12}}{(4k^2 C_s^4 - 4C_s^4 \lambda_1 \lambda_2)^2 k_e^{12}} = 1. \end{cases}$$

We end up up with

$$\begin{cases} X = e^{-\lambda_1 \delta} b_{11} - e^{-\lambda_2 \delta} b_{22} \xrightarrow{k_e \rightarrow \infty} \frac{k^2 + \lambda_1 \lambda_2}{k^2 - \lambda_1 \lambda_2} (e^{-\lambda_1 \delta} - e^{-\lambda_2 \delta}), \\ Y = e^{-(\lambda_1 + \lambda_2) \delta} (b_{11} b_{22} - b_{12} b_{21}) \xrightarrow{k_e \rightarrow \infty} e^{-(\lambda_1 + \lambda_2) \delta}, \end{cases}$$

which is the classical Schwarz method, see (2.8), therefore  $\rho_E \xrightarrow{k_e \rightarrow \infty} \rho_{cla}$ .

We have seen so far that zeroth order TTC gives the best approximation of the non-local terms in the transmissions conditions. Schwarz algorithm with TTC is known to be good for low frequencies but we need to improve the behaviour for high frequencies.

The most interesting case seems to be  $\frac{\omega}{C_p} < k_e < \frac{\omega}{C_s}$  as we can see in Figure 3.10 for the overlapping algorithm.

Unfortunately, we cannot perform an accurate analysis as the expression of the convergence factor is very complicated in this case. Nevertheless, we can find the optimal value  $k_e$ , that is the one giving the best possible algorithm, numerically by minimising the maximum of the convergence factor for all frequencies excluding only two small intervals around the two frequencies  $k_1$  and  $k_2$  for which the algorithm is always divergent:

$$(3.5) \quad \min_{k_e \in \mathbb{R}_*^+} \left( \max_{\substack{k \in (k_{\min}, k_1^-) \cup (k_1^+, k_2^-) \\ \cup (k_2^+, k_{\max})}} |\rho_E(k, k_e, \omega, C_p, C_s, \delta, L)| \right)$$

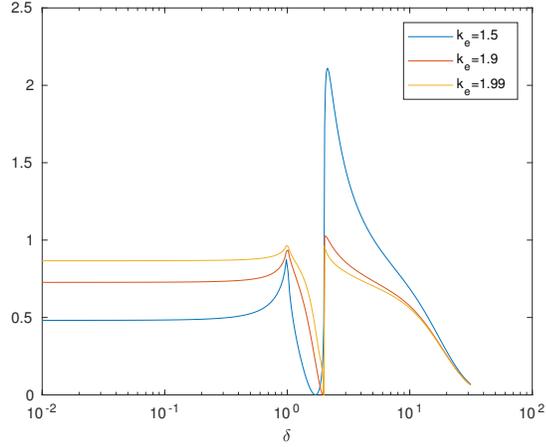


Figure 3.10:  $\frac{\omega}{C_p} < k_e < \frac{\omega}{C_s}$ : the spectrum of the iteration matrix for the overlapping Schwarz method with OIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ ,  $\delta = \frac{1}{10}$

where

$$k_j^\pm := k_j \pm \Delta k, \quad k_1 := \frac{\omega}{C_p}, \quad k_2 := \frac{\omega}{C_s}, \quad \Delta k := \frac{\pi}{L}.$$

We leave these two frequencies to the Krylov method and treat all the others by optimisation. Here we denoted by  $L$  the strip of height of the rectangular domain  $\Omega$  and by  $h$  the grid spacing. Note that the largest frequency supported by the numerical grid is  $k_{max} = \frac{\pi}{h}$  and the lowest frequency would be  $k_{min} = \frac{\pi}{L}$  for a rectangular domain of width  $L$ .

Let us give an example of such a numerical optimisation performed with the help of Matlab (see Appendix for details). Here are the values of the parameters for this example:

$$L = 10\pi, \quad \Delta k := \frac{1}{10}, \quad \delta = 0.1.$$

The optimal value obtained numerically is  $k_e = k_e^* = 1.9890$  as shown in Figure 3.11.

**Remark 3.4.** *The properties of this optimisation algorithm can be summarised as follows:*

- When  $\delta$  goes to zero, the minimised convergence factor attains its maximum at  $k = k_1 + \Delta k$  and/or  $k = k_2 + \Delta k$  and/or  $k = k_1 - \Delta k$ , which are balanced when the convergence factor is minimised (Figure 3.11).
- By varying the overlap size we notice that the optimal choice for  $k_e$  for  $\delta$  small has three different asymptotic regimes. It is interesting to note that even with this transition in  $k_e$  from the  $O(\delta)$  to the  $O(\delta^2)$  regime, the convergence factor does

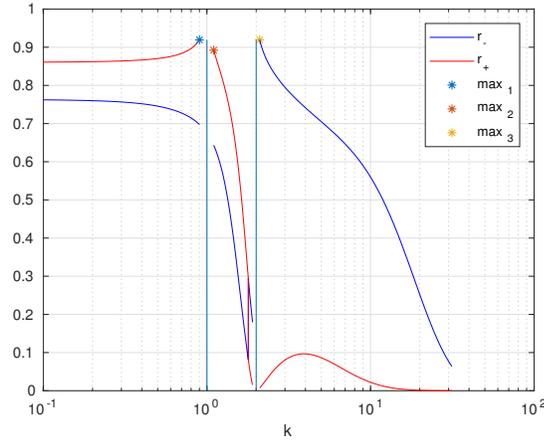


Figure 3.11: Optimized convergence factor, i.e. the maximum modulus of the eigenvalues of the iteration matrix for the overlapping Schwarz method with OIC for  $\frac{\omega}{C_p} < k_e < \frac{\omega}{C_s}$  and  $\omega = 1$ ,  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\delta = \frac{1}{10}$ .

not have such a transition, and behaves asymptotically as:

$$R \sim 1 - C_R(\omega, \mu, C_s, C_p, \rho) \delta.$$

These facts are illustrated in Figure 3.12.

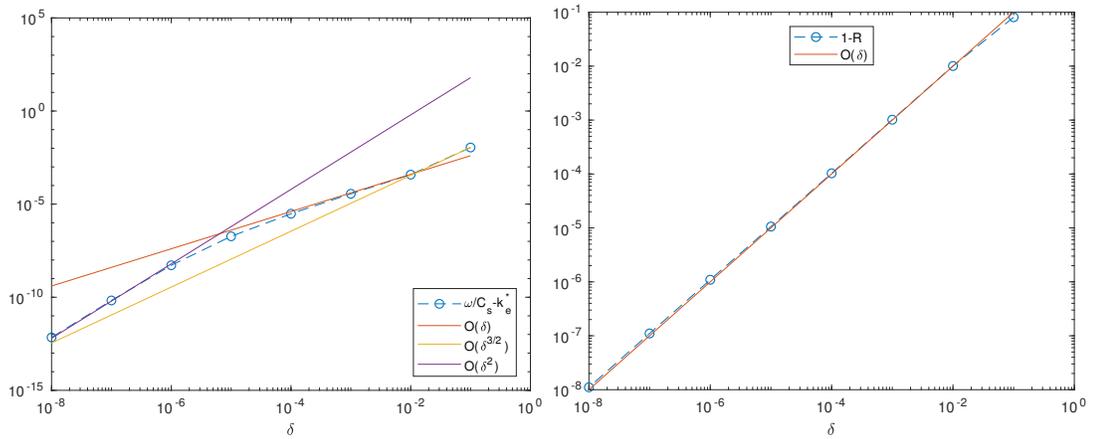


Figure 3.12: Left: Value of  $\frac{\omega}{C_s} - k_e^*$  obtained by minimising the convergence factor numerically compared to the asymptotic behaviour when the overlap  $\delta$  becomes small. Right: asymptotic behaviour of the convergence factor.

### 3.3 Higher order conditions

Another possible strategy to approximate transparent transmission operators if we keep the local polynomial terms in their expressions and approximate only the non-local ones is the following:

$$\begin{aligned}
 \widehat{\mathcal{S}}_1^H(1,1) &= \rho\omega^2 \frac{\sqrt{k_e^2 - \frac{\omega^2}{C_s^2}}}{k^2 - \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}} \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}}, \\
 \widehat{\mathcal{S}}_1^H(1,2) &= ik\rho \left( 2C_s^2 - \frac{\omega^2}{k^2 - \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}} \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}} \right), \\
 \widehat{\mathcal{S}}_1^H(2,1) &= -ik\rho \left( 2C_s^2 - \frac{\omega^2}{k^2 - \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}} \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}} \right), \\
 \widehat{\mathcal{S}}_1^H(2,2) &= \rho\omega^2 \frac{\sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}}{k^2 - \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}} \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}},
 \end{aligned} \tag{3.6}$$

with  $\widehat{\mathcal{S}}_2^H$  being defined from  $\widehat{\mathcal{S}}_1^H$  in the same way as for the optimal choice from the Theorem 2.2. We name these conditions *Higher Optimised Interface Conditions (HOIC)*, and as previously one can try to find the best choice for  $k_e$  giving the best possible algorithm, that is, which minimises the maximum of the contraction factor, excluding the frequencies  $k = \frac{\omega}{C_p}$  and  $k = \frac{\omega}{C_s}$  where the algorithm will never converge as well as a small interval around them.

We distinguish different cases as previously.

**Theorem 3.6** (Convergence of the overlapping Schwarz method with HOIC:  $k_e < \frac{\omega}{C_p}$ ).

For a given initial guess  $\mathbf{u}_1^0 \in (L^2(\Omega_1))^2$ ,  $\mathbf{u}_2^0 \in (L^2(\Omega_2))^2$ , the overlapping Schwarz method with HOIC such that  $k_e \in \left[0, \frac{\omega}{C_p}\right)$  has the following convergence factor

$$\rho_H(k, k_e, \omega, C_p, C_s, \delta) = \max\{|r_{\pm}|\}, \quad r_- = \left(\frac{p_1 - \lambda_1}{p_1 + \lambda_1} e^{-\lambda_1 \delta}\right)^2, \quad r_+ = \left(\frac{p_2 - \lambda_2}{p_2 + \lambda_2} e^{-\lambda_2 \delta}\right)^2, \tag{3.7}$$

and converges for  $k \in \mathbb{R}_+ \setminus \left\{\frac{\omega}{C_p}, \frac{\omega}{C_s}\right\}$ .

*Proof.* We start by plotting the convergence factor in the Figure 3.13. We use the general result on optimised Schwarz method from the Lemma 2.1 and insert our new

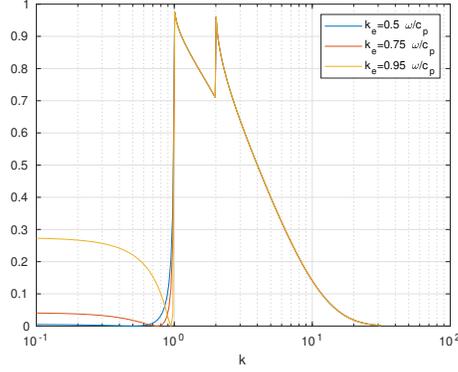


Figure 3.13:  $k_e < \frac{\omega}{C_p}$ : the spectrum of the iteration matrix from the overlapping Schwarz method with HOIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ ,  $\delta = \frac{1}{10}$ .

boundary operators (3.6). We get two interface iteration matrices

$$B_1 = \frac{\rho\omega^2}{(k^2 - p_1p_2)} \begin{bmatrix} p_1 - \lambda_1 & i\frac{p_1}{k}(p_2 - \lambda_2) \\ -i\frac{p_1}{k}(p_1 - \lambda_1) & p_2 - \lambda_2 \end{bmatrix},$$

and

$$B_2 = \frac{\rho\omega^2}{(k^2 - p_1p_2)} \begin{bmatrix} p_1 + \lambda_1 & i\frac{p_1}{k}(p_2 + \lambda_2) \\ -i\frac{p_1}{k}(p_1 + \lambda_1) & p_2 + \lambda_2 \end{bmatrix},$$

then we obtain the only half-iteration matrix we need

$$B = B_2^{-1}B_1 = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} \frac{p_1 - \lambda_1}{p_1 + \lambda_1} & 0 \\ 0 & \frac{p_2 - \lambda_2}{p_2 + \lambda_2} \end{bmatrix},$$

where

$$p_1 = \sqrt{k_e^2 - \frac{\omega^2}{C_s^2}}, \quad p_2 = \sqrt{k_e^2 - \frac{\omega^2}{C_p^2}}, \quad \lambda_1 = \sqrt{k^2 - \frac{\omega^2}{C_s^2}}, \quad \lambda_2 = \sqrt{k^2 - \frac{\omega^2}{C_p^2}}.$$

According to the remark 2.1, we end up with the convergence factor as in (3.7). Then, for  $k_e < \frac{\omega}{C_p}$  we have  $p_1, p_2 \in \mathbb{C}$  and then

$$|r_-| = \left| \frac{i\bar{p}_1 - \lambda_1}{i\bar{p}_1 + \lambda_1} \right|^2 \left| e^{-2\lambda_1\delta} \right|, \quad |r_+| = \left| \frac{i\bar{p}_2 - \lambda_2}{i\bar{p}_2 + \lambda_2} \right|^2 \left| e^{-2\lambda_2\delta} \right|$$

where  $\bar{p}_{1,2}$  are defined previously. We distinguish the following 5 cases:

- **Case**  $k > \frac{\omega}{C_s}$ . Here we have  $\lambda_1, \lambda_2 \in \mathbb{R}_+^*$  and

$$(3.8) \quad |r_-| = \left| \frac{i\bar{p}_1 - \lambda_1}{i\bar{p}_1 + \lambda_1} \right|^2 \left| e^{-2\lambda_1\delta} \right| = e^{-2\lambda_1\delta} < 1, \quad |r_+| = \left| \frac{i\bar{p}_2 - \lambda_2}{i\bar{p}_2 + \lambda_2} \right|^2 \left| e^{-2\lambda_2\delta} \right| = e^{-2\lambda_2\delta} < 1.$$

- **Case**  $k = \frac{\omega}{C_s}$ . In this case  $\lambda_1 = 0$  and

$$(3.9) \quad |r_+| = e^{-2\lambda_2\delta} < 1, \quad |r_-| = 1.$$

- **Case**  $\frac{\omega}{C_p} < k < \frac{\omega}{C_s}$ . In this case  $\lambda_1 \in \mathbb{C}^*$  and

$$(3.10) \quad |r_-| = \left| \frac{\bar{p}_1 - \bar{\lambda}_1}{\bar{p}_1 + \bar{\lambda}_1} \right|^2 \left| e^{-2i\bar{\lambda}_1\delta} \right| = \left( \frac{\bar{p}_1 - \bar{\lambda}_1}{\bar{p}_1 + \bar{\lambda}_1} \right)^2 < 1, \quad |r_+| = e^{-2\lambda_2\delta} < 1.$$

- **Case**  $k = \frac{\omega}{C_p}$ . Here  $\lambda_2 = 0, \lambda_1 \in \mathbb{C}^*$  and

$$(3.11) \quad |r_-| = \left( \frac{\bar{p}_1 - \bar{\lambda}_1}{\bar{p}_1 + \bar{\lambda}_1} \right)^2 < 1, \quad |r_+| = 1.$$

- **Case**  $k < \frac{\omega}{C_p}$ . In this case  $\lambda_1, \lambda_2 \in \mathbb{C}_+^*$  and therefore

$$|r_-| = \left( \frac{\bar{p}_1 - \bar{\lambda}_1}{\bar{p}_1 + \bar{\lambda}_1} \right)^2 < 1, \quad |r_+| = \left( \frac{\bar{p}_2 - \bar{\lambda}_2}{\bar{p}_2 + \bar{\lambda}_2} \right)^2 < 1.$$

□

**Remark 3.5.** *There are several cases (as seen in Figure 3.14) where the algorithm is not convergent as follows:*

- *If  $k_e \geq \frac{\omega}{C_p}$  both the non-overlapping and overlapping algorithms are non-convergent. Indeed  $p_2 \in \mathbb{R}_+$  and for  $k < \frac{\omega}{C_p}$  we get  $\lambda_2 = i\bar{\lambda}_2$  therefore*

$$|r_+| = \left| \frac{p_2 - i\bar{\lambda}_2}{p_2 + i\bar{\lambda}_2} \right|^2 \left| e^{-2i\bar{\lambda}_2\delta} \right| = 1.$$

- *If  $k_e < \frac{\omega}{C_p}$  the non-overlapping algorithm is not converging. In this case  $p_2 \in i\mathbb{R}_+$ , for  $k \geq \frac{\omega}{C_p}$  we get  $\lambda_2 \in \mathbb{R}_+$  therefore*

$$(3.12) \quad |r_+| = \left| \frac{i\bar{p}_2 - \lambda_2}{i\bar{p}_2 + \lambda_2} \right|^2 = 1.$$

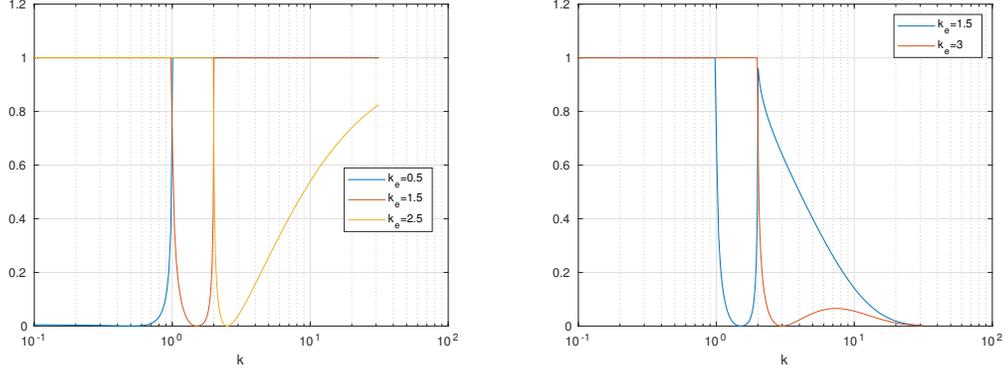


Figure 3.14: Spectrum of the iteration matrix for the Schwarz method with HOIC with  $C_p = 1$ ,  $C_s = \frac{1}{2}$ . Left:  $\delta = 0$ . Right:  $\delta = \frac{1}{10}$ .

According to the results above, an optimal parameter, if it exists must lie in the interval  $[0, \frac{\omega}{C_p}]$ . Moreover, we notice from the formula (3.7) that the convergence factor of the algorithm applied to the time-harmonic equations of elastic waves can be related to that coming from the application of the optimised overlapping Schwarz method to Helmholtz equations. Nevertheless, as the situation is more complex here, as both eigenvalues  $r_{\pm}$  can play a role, we cannot extend directly the results presented in [GZ16]. This will be the object of future work and needs further investigation.

### 3.3.1 Two-sided version of higher order conditions

The natural next step is to choose one  $k_e$  different for each subdomain. We will analyse the convergence factor for the new algorithms.

**Lemma 3.2** (Convergence factor of the two-sided algorithm with high order conditions). *For a given initial guess  $\mathbf{u}_1^0 \in (L^2(\Omega_1))^2$ ,  $\mathbf{u}_2^0 \in (L^2(\Omega_2))^2$ , the Schwarz algorithm with two-sided HOIC has the following convergence factor*

$$\rho_H(k, k_e^{(1)}, k_e^{(2)}, \omega, C_p, C_s, \delta) = \max\{|r_+|, |r_-|\},$$

where

$$r_- = \frac{(\lambda_1 - p_1(k_e^{(1)}))(\lambda_1 - p_1(k_e^{(2)}))}{(\lambda_1 + p_1(k_e^{(1)}))(\lambda_1 + p_1(k_e^{(2)}))} e^{-2\lambda_1\delta},$$

and

$$r_+ = \frac{(\lambda_2 - p_2(k_e^{(1)}))(\lambda_2 - p_2(k_e^{(2)}))}{(\lambda_2 + p_2(k_e^{(1)}))(\lambda_2 + p_2(k_e^{(2)}))} e^{-2\lambda_2\delta}.$$

*Proof.* We use as a starting point the TBC where we insert now  $(k_e^{(1)}, k_e^{(2)})$  and get

$$\widehat{S}_j^H = \begin{bmatrix} \rho\omega^2 \frac{p_1(k_e^{(j)})}{k^2 - p_1(k_e^{(j)})p_2(k_e^{(j)})} & ik\rho \left( 2C_s^2 - \frac{\omega^2}{k^2 - p_1(k_e^{(j)})p_2(k_e^{(j)})} \right) \\ -ik\rho \left( 2C_s^2 - \frac{\omega^2}{k^2 - p_1(k_e^{(j)})p_2(k_e^{(j)})} \right) & \rho\omega^2 \frac{p_2(k_e^{(j)})}{k^2 - p_1(k_e^{(j)})p_2(k_e^{(j)})} \end{bmatrix}, \quad j = 1, 2,$$

our new interface operators, which lead us to the two half-iteration matrices

$$B = \begin{bmatrix} \frac{p_1(k_e^{(2)}) - \lambda_1}{p_1(k_e^{(2)}) + \lambda_1} & 0 \\ 0 & \frac{p_2(k_e^{(2)}) - \lambda_2}{p_2(k_e^{(2)}) + \lambda_2} \end{bmatrix}, \quad A = \begin{bmatrix} e^{-2\lambda_1\delta} \frac{p_1(k_e^{(1)}) - \lambda_1}{p_1(k_e^{(1)}) + \lambda_1} & 0 \\ 0 & e^{-2\lambda_2\delta} \frac{p_2(k_e^{(1)}) - \lambda_2}{p_2(k_e^{(1)}) + \lambda_2} \end{bmatrix},$$

and the iteration matrix  $M = AB$  gives the eigenvalues and the convergence factor.  $\square$

We have this first result

**Theorem 3.7** (Convergence of the overlapping Schwarz method with two-sided HOIC).  
*The overlapping Schwarz method with two-sided HOIC converges for  $k \in \mathbb{R}_+ \setminus \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$  and  $\forall \delta > 0$  except when  $k_e^{(1)} > \frac{\omega}{C_p}$  and  $k_e^{(2)} > \frac{\omega}{C_p}$ .*

*Proof.* We start by drawing the convergence factor of the algorithm for different values of  $k_e^{(1,2)}$  (see Figure 3.15).

If  $k_e^{(1)} < \frac{\omega}{C_p}$  we get

$$r_- = \frac{(\lambda_1 - i\bar{p}_1(k_e^{(1)}))(\lambda_1 - p_1(k_e^{(2)}))}{(\lambda_1 + i\bar{p}_1(k_e^{(1)}))(\lambda_1 + p_1(k_e^{(2)}))} e^{-2\lambda_1\delta}$$

and

$$r_+ = \frac{(\lambda_2 - i\bar{p}_2(k_e^{(1)}))(\lambda_2 - p_2(k_e^{(2)}))}{(\lambda_2 + i\bar{p}_2(k_e^{(1)}))(\lambda_2 + p_2(k_e^{(2)}))} e^{-2\lambda_2\delta}.$$

We distinguish three cases according to the values of  $k_e^{(2)}$ .

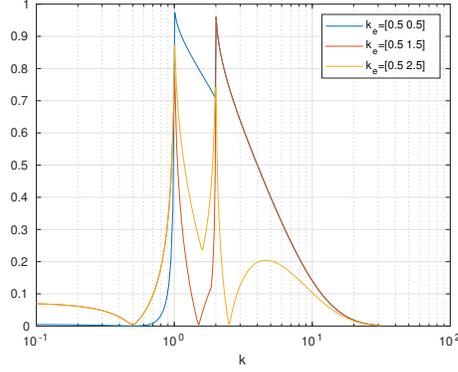


Figure 3.15: Spectrum of the iteration matrix for the overlapping Schwarz method with two-sided HOIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ ,  $\delta = \frac{1}{10}$ .

**Case 1:**  $k_e^{(2)} < \frac{\omega}{C_p}$ . We distinguish four cases according to the values of  $k$

- **Case**  $k < \frac{\omega}{C_p}$ .

$$\begin{aligned}
 |r_-| &= \left| \frac{\left( i\bar{\lambda}_1 - i\bar{p}_1 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_1 - i\bar{p}_1 \left( k_e^{(2)} \right) \right)}{\left( i\bar{\lambda}_1 + i\bar{p}_1 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_1 + i\bar{p}_1 \left( k_e^{(2)} \right) \right)} e^{-2i\bar{\lambda}_1\delta} \right| \\
 &= \left| \frac{\left( \bar{\lambda}_1 - \bar{p}_1 \left( k_e^{(1)} \right) \right) \left( \bar{\lambda}_1 - \bar{p}_1 \left( k_e^{(2)} \right) \right)}{\left( \bar{\lambda}_1 + \bar{p}_1 \left( k_e^{(1)} \right) \right) \left( \bar{\lambda}_1 + \bar{p}_1 \left( k_e^{(2)} \right) \right)} \right| < 1, \\
 |r_+| &= \left| \frac{\left( i\bar{\lambda}_2 - i\bar{p}_2 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_2 - i\bar{p}_2 \left( k_e^{(2)} \right) \right)}{\left( i\bar{\lambda}_2 + i\bar{p}_2 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_2 + i\bar{p}_2 \left( k_e^{(2)} \right) \right)} e^{-2i\bar{\lambda}_2\delta} \right| \\
 &= \left| \frac{\left( \bar{\lambda}_2 - \bar{p}_2 \left( k_e^{(1)} \right) \right) \left( \bar{\lambda}_2 - \bar{p}_2 \left( k_e^{(2)} \right) \right)}{\left( \bar{\lambda}_2 + \bar{p}_2 \left( k_e^{(1)} \right) \right) \left( \bar{\lambda}_2 + \bar{p}_2 \left( k_e^{(2)} \right) \right)} \right| < 1.
 \end{aligned}$$

- **Case**  $k > \frac{\omega}{C_s}$ .

$$\begin{aligned}
 |r_-| &= \left| \frac{\left( \lambda_1 - i\bar{p}_1 \left( k_e^{(1)} \right) \right) \left( \lambda_1 - i\bar{p}_1 \left( k_e^{(2)} \right) \right)}{\left( \lambda_1 + i\bar{p}_1 \left( k_e^{(1)} \right) \right) \left( \lambda_1 + i\bar{p}_1 \left( k_e^{(2)} \right) \right)} e^{-2\lambda_1\delta} \right| = e^{-2\lambda_1\delta} < 1, \\
 |r_+| &= \left| \frac{\left( \lambda_2 - i\bar{p}_2 \left( k_e^{(1)} \right) \right) \left( \lambda_2 - i\bar{p}_2 \left( k_e^{(2)} \right) \right)}{\left( \lambda_2 + i\bar{p}_2 \left( k_e^{(1)} \right) \right) \left( \lambda_2 + i\bar{p}_2 \left( k_e^{(2)} \right) \right)} e^{-2\lambda_2\delta} \right| = e^{-2\lambda_2\delta} < 1.
 \end{aligned}$$

- **Case**  $\frac{\omega}{C_p} < k < \frac{\omega}{C_s}$ . In that case,  $|r_-|$  is like in the case  $k < \frac{\omega}{C_p}$  and  $|r_+|$  is like in the case  $k > \frac{\omega}{C_s}$ , so both are lower than one.
- **Case**  $k = \frac{\omega}{C_p}$  **and**  $k = \frac{\omega}{C_s}$ . A quick computation shows  $\rho_H = 1$ .

**Case 2:**  $k_e^{(2)} > \frac{\omega}{C_s}$ . We distinguish four cases according to the values of  $k$

- **Case**  $k < \frac{\omega}{C_p}$ .

$$|r_-| = \left| \frac{\left( i\bar{\lambda}_1 - i\bar{p}_1 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_1 - p_1 \left( k_e^{(2)} \right) \right)}{\left( i\bar{\lambda}_1 + i\bar{p}_1 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_1 + p_1 \left( k_e^{(2)} \right) \right)} e^{-2i\bar{\lambda}_1\delta} \right| = \left| \frac{\bar{\lambda}_1 - \bar{p}_1 \left( k_e^{(1)} \right)}{\bar{\lambda}_1 + \bar{p}_1 \left( k_e^{(1)} \right)} \right| < 1,$$

$$|r_+| = \left| \frac{\left( i\bar{\lambda}_2 - i\bar{p}_2 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_2 - p_2 \left( k_e^{(2)} \right) \right)}{\left( i\bar{\lambda}_2 + i\bar{p}_2 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_2 + p_2 \left( k_e^{(2)} \right) \right)} e^{-2i\bar{\lambda}_2\delta} \right| = \left| \frac{\bar{\lambda}_2 - \bar{p}_2 \left( k_e^{(1)} \right)}{\bar{\lambda}_2 + \bar{p}_2 \left( k_e^{(1)} \right)} \right| < 1.$$

- **Case**  $k > \frac{\omega}{C_s}$ .

$$|r_-| = \left| \frac{\left( \lambda_1 - i\bar{p}_1 \left( k_e^{(1)} \right) \right) \left( \lambda_1 - p_1 \left( k_e^{(2)} \right) \right)}{\left( \lambda_1 + i\bar{p}_1 \left( k_e^{(1)} \right) \right) \left( \lambda_1 + p_1 \left( k_e^{(2)} \right) \right)} e^{-2\lambda_1\delta} \right| = \left| \frac{\left( \lambda_1 - p_1 \left( k_e^{(2)} \right) \right)}{\left( \lambda_1 + p_1 \left( k_e^{(2)} \right) \right)} e^{-2\lambda_1\delta} \right| < 1,$$

$$|r_+| = \left| \frac{\left( \lambda_2 - i\bar{p}_2 \left( k_e^{(1)} \right) \right) \left( \lambda_2 - p_2 \left( k_e^{(2)} \right) \right)}{\left( \lambda_2 + p_2 \left( k_e^{(1)} \right) \right) \left( \lambda_2 + p_2 \left( k_e^{(2)} \right) \right)} e^{-2\lambda_2\delta} \right| = \left| \frac{\left( \lambda_2 - p_2 \left( k_e^{(2)} \right) \right)}{\left( \lambda_2 + p_2 \left( k_e^{(2)} \right) \right)} e^{-2\lambda_2\delta} \right| < 1.$$

- **Case**  $\frac{\omega}{C_p} < k < \frac{\omega}{C_s}$ . In that case,  $|r_-|$  is like in the case  $k < \frac{\omega}{C_p}$  and  $|r_+|$  is like in the case  $k > \frac{\omega}{C_s}$ , so both are lower than one.
- **Case**  $k = \frac{\omega}{C_p}$  **and**  $k = \frac{\omega}{C_s}$ . A quick computation shows  $\rho_H = 1$ .

**Case 3:**  $\frac{\omega}{C_p} < k_e^{(2)} < \frac{\omega}{C_s}$ . One notices that in that case,  $r_-$  is like in the Case 1 and  $r_+$  is like in the Case 2, so it will be convergent as well.

The proof for  $k_e^{(2)} < \frac{\omega}{C_p}$  follows easily by symmetry.

To conclude, we consider the remaining cases not covered above, that is when both  $k_e^{(1)}$  and  $k_e^{(2)}$  are bigger than  $\frac{\omega}{C_p}$ . We see that for  $k < \frac{\omega}{C_p}$  we get

$$|r_+| = \left| \frac{\left( i\bar{\lambda}_2 - p_2 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_2 - p_2 \left( k_e^{(2)} \right) \right)}{\left( i\bar{\lambda}_2 + p_2 \left( k_e^{(1)} \right) \right) \left( i\bar{\lambda}_2 + p_2 \left( k_e^{(2)} \right) \right)} e^{-2i\bar{\lambda}_2\delta} \right| = 1.$$

which means that the global convergence factor will be always bigger or equal to 1 and the algorithm will be divergent.  $\square$

**Lemma 3.3** (Convergence of the non-overlapping Schwarz method with two-sided HOIC). *The non-overlapping Schwarz method with two-sided HOIC converges for  $k \in \mathbb{R}_+ \setminus \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$*

$$\forall k_e^{(2)} > \frac{\omega}{C_s} \quad \text{if} \quad k_e^{(1)} < \frac{\omega}{C_p} \quad \& \quad \forall k_e^{(1)} > \frac{\omega}{C_s} \quad \text{if} \quad k_e^{(2)} < \frac{\omega}{C_p}.$$

*Proof.* The proof follows the lines of the theorem 3.7 where we put  $\delta = 0$ .  $\square$

The next step would be again to extend the results or establish a link with Helmholtz equations as in [GZ16].

### 3.3.2 General higher order conditions

There is another way to approximate the nonlocal  $\lambda_j$ ,  $j = 1, 2$ , by well-chosen constants

$$\lambda_1^{app} = \alpha_1 + i\beta_1, \quad \lambda_2^{app} = \alpha_2 + i\beta_2 \quad \text{on } \Omega_1,$$

and

$$\tilde{\lambda}_1^{app} = \tilde{\alpha}_1 + i\tilde{\beta}_1, \quad \tilde{\lambda}_2^{app} = \tilde{\alpha}_2 + i\tilde{\beta}_2 \quad \text{on } \Omega_2,$$

and then try to find the best choice for the parameters in both domains. We name these conditions *General Higher Optimised Interface Conditions (GHOIC)*.

**Theorem 3.8.** *For a given initial guess  $\mathbf{u}_1^0 \in (L^2(\Omega_1))^2$ ,  $\mathbf{u}_2^0 \in (L^2(\Omega_2))^2$ , the Schwarz method with GHOIC converges for  $k \in \mathbb{R}_+ \setminus \left\{ \frac{\omega}{C_p}, \frac{\omega}{C_s} \right\}$ .*

*Proof.* The proof follows the lines of Lemma 3.2, where we replace

$$p_1 \left( k_e^{(1)} \right) \rightarrow \lambda_1^{app}, \quad p_2 \left( k_e^{(1)} \right) \rightarrow \lambda_2^{app} \quad \text{on } \Omega_1,$$

and

$$p_1 \left( k_e^{(2)} \right) \rightarrow \tilde{\lambda}_1^{app}, \quad p_2 \left( k_e^{(2)} \right) \rightarrow \tilde{\lambda}_2^{app} \quad \text{on } \Omega_2.$$

The new eigenvalues of the iteration matrix are in this case

$$r_- = \frac{(\lambda_1^{app} - \lambda_1) (\tilde{\lambda}_1^{app} - \lambda_1)}{(\lambda_1^{app} + \lambda_1) (\tilde{\lambda}_1^{app} + \lambda_1)} e^{-2\lambda_1\delta}, \quad r_+ = \frac{(\lambda_2^{app} - \lambda_2) (\tilde{\lambda}_2^{app} - \lambda_2)}{(\lambda_2^{app} + \lambda_2) (\tilde{\lambda}_2^{app} + \lambda_2)} e^{-2\lambda_2\delta},$$

which gives

$$|r_-| = \left| \frac{(\alpha_1 + i\beta_1 - \lambda_1) (\tilde{\alpha}_1 + i\tilde{\beta}_1 - \lambda_1)}{(\alpha_1 + i\beta_1 + \lambda_1) (\tilde{\alpha}_1 + i\tilde{\beta}_1 + \lambda_1)} e^{-\lambda_1 \delta} \right|$$

$$= \begin{cases} \frac{\sqrt{(\alpha_1 - \lambda_1)^2 + \beta_1^2} \sqrt{(\tilde{\alpha}_1 - \lambda_1)^2 + \tilde{\beta}_1^2}}{\sqrt{(\alpha_1 + \lambda_1)^2 + \beta_1^2} \sqrt{(\tilde{\alpha}_1 + \lambda_1)^2 + \tilde{\beta}_1^2}} e^{-2\lambda_1 \delta} < 1 & \text{if } k > \frac{\omega}{C_s}, \\ \frac{\sqrt{(\beta_1 - \bar{\lambda}_1)^2 + \alpha_1^2} \sqrt{(\tilde{\beta}_1 - \bar{\lambda}_1)^2 + \tilde{\alpha}_1^2}}{\sqrt{(\beta_1 + \bar{\lambda}_1)^2 + \alpha_1^2} \sqrt{(\tilde{\beta}_1 + \bar{\lambda}_1)^2 + \tilde{\alpha}_1^2}} < 1 & \text{if } k < \frac{\omega}{C_s}, \end{cases}$$

and similarly

$$|r_+| = \begin{cases} \frac{\sqrt{(\alpha_2 - \lambda_2)^2 + \beta_2^2} \sqrt{(\tilde{\alpha}_2 - \lambda_2)^2 + \tilde{\beta}_2^2}}{\sqrt{(\alpha_2 + \lambda_2)^2 + \beta_2^2} \sqrt{(\tilde{\alpha}_2 + \lambda_2)^2 + \tilde{\beta}_2^2}} e^{-2\lambda_2 \delta} < 1 & \text{if } k > \frac{\omega}{C_p}, \\ \frac{\sqrt{(\beta_2 - \bar{\lambda}_2)^2 + \alpha_2^2} \sqrt{(\tilde{\beta}_2 - \bar{\lambda}_2)^2 + \tilde{\alpha}_2^2}}{\sqrt{(\beta_2 + \bar{\lambda}_2)^2 + \alpha_2^2} \sqrt{(\tilde{\beta}_2 + \bar{\lambda}_2)^2 + \tilde{\alpha}_2^2}} < 1 & \text{if } k < \frac{\omega}{C_p}. \end{cases}$$

We can see that the convergence factor remains less than one (except on the resonance frequencies as usual), either with overlap ( $\delta > 0$ ) or not ( $\delta = 0$ ). If it is one-sided ( $\alpha_j = \tilde{\alpha}_j, \beta_j = \tilde{\beta}_j, j = 1, 2$ ), the non-overlapping and overlapping methods are both convergent.  $\square$

Then we have this specific result

**Remark 3.6.** *We could chose the parameters for the non-overlapping Schwarz method with one-sided GHOIC as*

$$q_j^* = \alpha_j^*(1 + i), j = 1, 2,$$

where the  $\alpha_j^*$  are chosen as in [GMN02b]

$$\alpha_1^* = \left( \frac{\sqrt{\frac{\omega^2}{C_s^2} - (k_2^-)^2} \sqrt{(k_{\max})^2 - \frac{\omega^2}{C_s^2}}}{2} \right)^{\frac{1}{2}}, \alpha_2^* = \left( \frac{\sqrt{\frac{\omega^2}{C_s^2} - (k_2^-)^2} \sqrt{(k_{\max})^2 - \frac{\omega^2}{C_s^2}}}{2} \right)^{\frac{1}{2}}.$$

In this case the corresponding convergence factor is

$$\rho(k, \omega, C_p, C_s, q_1^*, q_2^*) = \frac{1 - \sqrt{2} \left( \frac{k_1^2 - (k_1^-)^2}{(k_{\max}^*)^2 - k_1^2} \right)^{\frac{1}{4}} + \sqrt{\frac{k_1^2 - (k_1^-)^2}{(k_{\max}^*)^2 - k_1^2}}}{1 + \sqrt{2} \left( \frac{k_1^2 - (k_1^-)^2}{(k_{\max}^*)^2 - k_1^2} \right)^{\frac{1}{4}} + \sqrt{\frac{k_1^2 - (k_1^-)^2}{(k_{\max}^*)^2 - k_1^2}}},$$

where

$$k_j^\pm := k_j \pm \Delta k, \quad k_1 := \frac{\omega}{C_p}, \quad k_2 := \frac{\omega}{C_s}, \quad \Delta k := \frac{\pi}{L}, \quad k_{\min} = \frac{\pi}{L}, \quad k_{\max} = \frac{\pi}{h}.$$

Note that this is not a solution to the minmax problem but it could potentially improve the convergence of the algorithm.

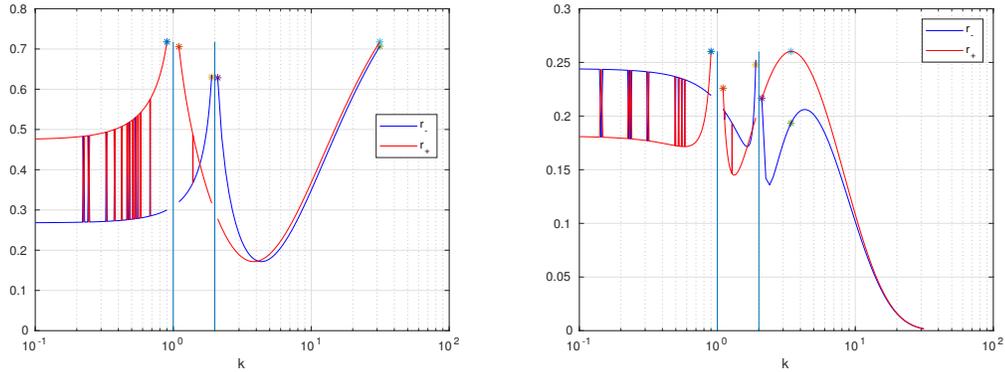


Figure 3.16: Spectrum of the iteration matrix from the Schwarz method with one-sided GHOIC and  $C_p = 1$ ,  $C_s = \frac{1}{2}$ ,  $\omega = 1$ . Left:  $\delta = 0$ . Right:  $\delta = \frac{1}{10}$ .

The solution to the min/max problem is even more complex than in the case of Helmholtz equations. As an illustration, we can solve this problem numerically by Matlab (an illustration of this can be seen in Figure 3.16) and we see that the optimal value is a result of the equioscillation of the convergence factor, but since we have to equilibrate different quantities on various intervals, the extension of the results obtained in the Helmholtz case is not straightforward.

### 3.4 Conclusions and future works

As we mentioned earlier, the classical Schwarz method does not converge without overlap. In turn, simple optimised Schwarz methods, however, can be used without overlap,

and non-overlapping Schwarz methods can be of great interest if the physical properties in the subdomains differ. These methods deserve further exploration as the preliminary results seem to be very promising.

The first obvious future work would be to find the optimal parameter  $k_e$  asymptotically as a function of the parameters of the problem for the Schwarz algorithm with or without overlap. The numerical optimisation problem solved with Matlab has shown an improvement and an asymptotic behaviour of the convergence factor that needs to be confirmed by a mathematical proof.

Secondly, the general higher order conditions, one or two-sided derived in the last part of the chapter demonstrate that there is a strong link with the Helmholtz equations. Nevertheless, since for the time-harmonic elastic waves, the expressions being far more complex and involve additional quantities, the extension of these results is not straightforward. This link needs to be understood and the techniques used in the case of Helmholtz equations generalised to the elastic waves.

Thirdly, we would like to apply these methods to more realistic physical models from geophysics, that are still challenging from a computational point of view.

## Chapter 4

# Numerical assessment of a grid coarse space for elastic waves

As we have seen in the previous chapters, solving the time-harmonic elastic wave equation is a challenging task. Despite several attempts to solve it efficiently, there doesn't seem to exist an established and robust preconditioner, whose behaviour is independent of the frequency and of the number of subdomains, in the case of general decompositions. A very robust domain decomposition method was developed recently in [GSV17a, GSV17b], where two-level domain decomposition approximations of the damped (with absorption) Helmholtz equation  $-\Delta u - (k^2 + i\varepsilon)u = f$  were used as preconditioners for the pure Helmholtz equation without absorption; in this particular case the coarse correction is based on a coarse mesh with diameter constrained by the wave number  $k$ . As a result, in the ideal case, the obtained convergence was independent of the wave number. Our purpose is to perform a preliminary numerical study where we can assess the performance of a two-level grid based preconditioner in the case of the time-harmonic elastic waves without absorption but with absorbing transmission conditions at the interfaces between domains.

### 4.1 The grid coarse space

In order to achieve *weak scalability* or the independence with respect to the number of subdomains, we need to add a coarse component to the one-level ORAS preconditioner

(1.8). The *two-level* preconditioner can be written in a generic way as follows

$$(4.1) \quad \begin{aligned} M_{2,RAS}^{-1} &= QM_{RAS}^{-1}P + ZE^{-1}Z^*, \\ M_{2,ORAS}^{-1} &= QM_{ORAS}^{-1}P + ZE^{-1}Z^*, \end{aligned}$$

where  $*$  denotes the conjugate transpose and  $M_{RAS}^{-1}$  is the one-level preconditioner given in (1.7) and  $M_{ORAS}^{-1}$  is given in (1.8).

Other ingredients:

- $Z$  is a rectangular matrix with full column rank,
- $E = Z^*AZ$  is the so-called coarse grid matrix,
- $\Xi = ZE^{-1}Z^*$  is the so-called coarse grid correction matrix.

From now on we will use  $P = I - A\Xi$  and  $Q = I - \Xi A$ , which is a *hybrid* two-level preconditioner also called the *Balancing* Neumann Neumann (BNN) preconditioner.

**Remark 4.1.** *If  $P = Q = I$ , we would get an additive two-level preconditioner.*

Preconditioner (4.1) is characterized by the choice of  $Z$ , whose columns span the *coarse space* (CS). Consequently, the definition of  $Z$  will give the nature of the new preconditioner and the columns of  $Z$  represent the basis vectors of what is called the CS.

The most natural coarse space would be one based on a coarser mesh, we subsequently call it *grid coarse space*.

Let us consider  $\mathcal{T}_{H_{\text{coarse}}}$  a simplicial mesh of the computational domain  $\Omega$  with a mesh diameter  $H_{\text{coarse}}$  and  $W_{H_{\text{coarse}}}$  the corresponding finite element space.

Let  $\mathcal{R}_0: W_h \rightarrow W_{H_{\text{coarse}}}$  be the nodal interpolation operator from the fine grid finite element space to the coarse grid finite element space and  $R_0$  the corresponding matrix. We define  $Z = R_0^T$ , then in this case  $E = Z^*AZ$  is the stiffness matrix of the problem discretised on the coarse mesh and the component  $ZE^{-1}Z^*$  of the preconditioner is called *coarse space correction*.

## 4.2 Numerical results

In this section we compare the two-level preconditioners defined in (4.1) for the some of test cases presented in Section 1.4. We will particularly focus on the highly oscillatory test case (Test case 2) and the heterogeneous one (Test case 3).

### Two-level RAS and ORAS: Test case 2

In these tests we notice that the iteration count is only slowly varying with the number of subdomains and that there is a considerable improvement with respect to the one-level method. We also notice that the ORAS preconditioner is not necessarily better than RAS, the effect of transmission conditions seems to be less obvious when a second level is added. Also we notice that the type of the decomposition has only a little influence on the iteration count.

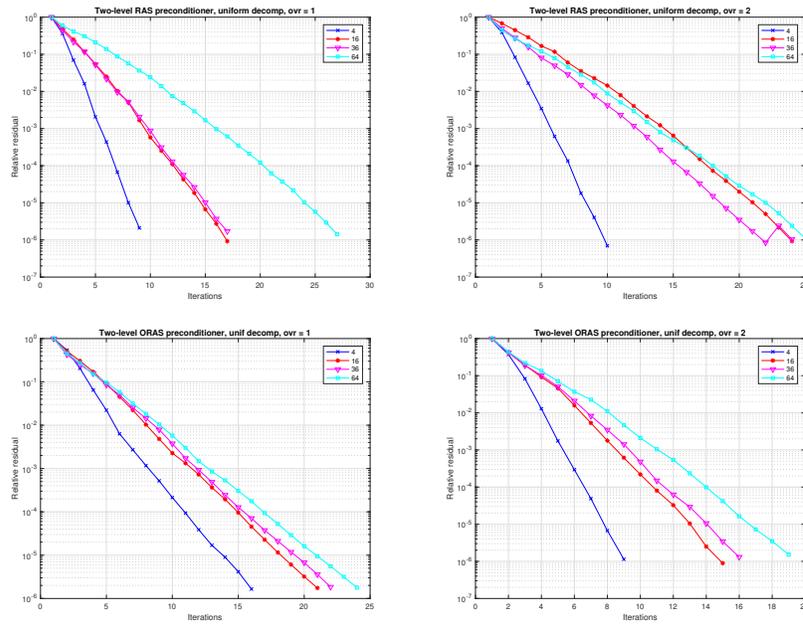


Figure 4.1: Convergence history for RAS (upper row) and ORAS (lower row) on uniform decompositions and overlap = $2h$  (left) and overlap= $4h$  (right)

This experiment is repeated on METIS decompositions (Figure 4.2) and a numerical summary of the results of the previous figures can be found in the table below.

N	Overlap = $2h$				Overlap = $4h$			
	RAS		ORAS		RAS		ORAS	
	Unif	MTS	Unif	MTS	Unif	MTS	Unif	MTS
<b>4</b>	9	11	16	18	10	10	9	10
<b>16</b>	17	18	20	21	24	19	15	15
<b>36</b>	17	20	21	25	22	20	16	18
<b>64</b>	26	18	23	28	25	23	19	19

Table 4.1: Preconditioners comparison for the test case 2

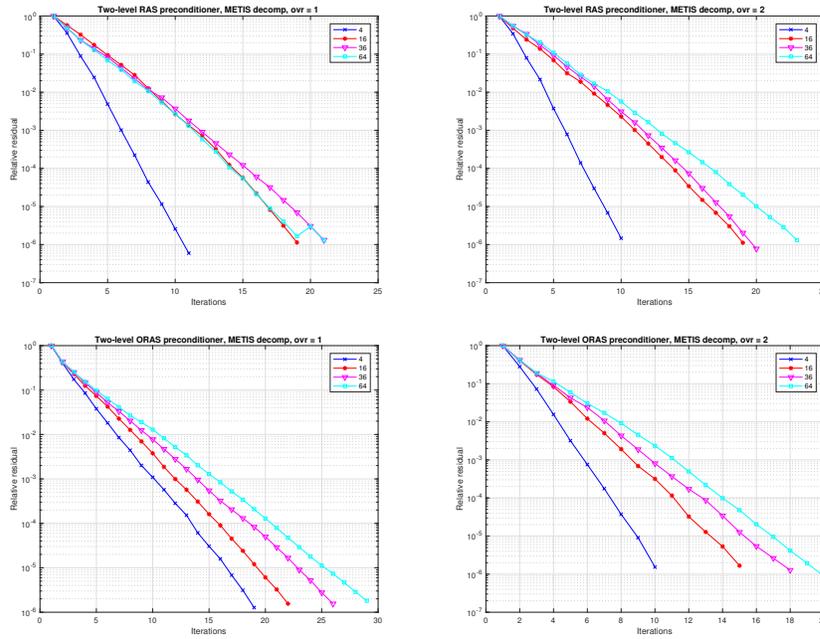


Figure 4.2: Convergence history for RAS (upper row) and ORAS (lower row) on METIS decompositions and overlap  $=2h$  (left) and overlap  $=4h$  (right)

In conclusion, these preliminary tests show that the two-level preconditioner seems to be very robust, that is the iterations vary only slightly when the number of subdomains is increased. More extensive tests are needed to conclude on the general applicability of this two-level method.

### Two-level RAS and ORAS: Test case 3

We will focus now on the heterogeneous test case defined on a disk, where the decomposition in subdomains is done by METIS.

A numerical summary can be found in the table 4.2.

N	Overlap $=2h$		Overlap $=4h$	
	RAS	ORAS	RAS	ORAS
<b>4</b>	31	33	29	28
<b>16</b>	40	45	36	37
<b>36</b>	47	75	43	54
<b>64</b>	52	128	51	97

Table 4.2: Preconditioners comparison for the heterogeneous test case 3

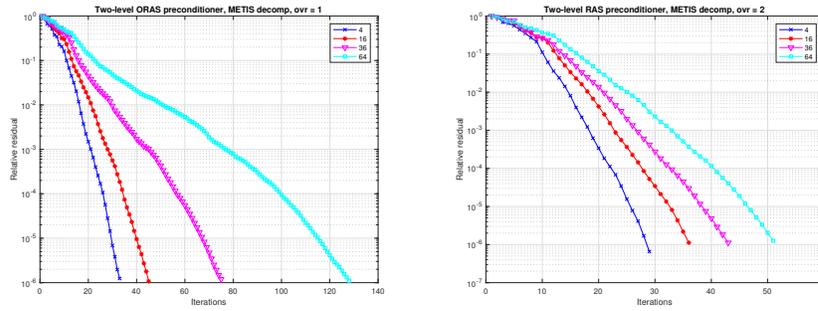


Figure 4.3: Convergence history for ORAS and RAS on METIS decompositions and overlap  $=2h$  (left) and overlap  $=4h$  (right)

Here we notice that the two-level ORAS preconditioner is less efficient whereas the two-level RAS preconditioner leads to quite a stable number of iterations where we can see only a slight increase when the number of subdomain gets bigger. At the same time, there is an important decrease in the iteration count with respect to the one-level method.

We can conclude these preliminary tests by saying that a two-level preconditioner is mandatory in achieving robustness of the domain decomposition solver but the behaviour of this method is not completely understood in the case of time-harmonic elastic waves and therefore it requires further investigation.

# Appendices

# Appendix A

## Matlab implementations

In the main script denoted by Main.m, one enters the different parameters and the choice of the methods. In Rho.m, the computation of the convergence factor based on different interface transmission conditions is implemented. More details are given in the comments inside the code.

### A.1 Main script

```
1  % INITIALIZATION-----
2  global omega cs cp rho kmin kmax delta L choice order;
3  format long;
4
5  % Parameters
6  omega=1.; cp=1.; cs=0.5; rho=1.; delta=0.1;
7  L=10.*pi; h=1./10; kmin=pi./L; kmax=pi./h; LL=0; RR=0; dk=pi./L;
8
9  % Choice of method
10 % 1 = Classical // 2 = Taylor // 3 = two-sided Taylor // 4 = one parameter family // 6 =
    Two-Sided one parameter family // 8 = Higher conditions // 10 = Two-sided higher
    conditions // 12 = General higher conditions
11 % To find the min-max, the equivalent in a discontinuous grid
12 % 5 = one parameter family // 7 = Two-sided one parameter family // 11 = Two-sided higher
    conditions // 13 = General higher conditions
13 choice=12;
14
15 % Choice of order for Taylor (zeroth, improved zeroth, second)
16 order=0;
17
18 % Choice of ke
```

```

19 if (choice==1) || (choice==2) || (choice==4) || (choice==5) || (choice==8) || (choice==9)
20     ke=0.995;
21 else if (choice==12) || (choice==13)
22     alpha=[1 1];
23 else
24     ke=[0.5 5];
25 end;
26
27 % CODE -----
28 if (choice==1) || (choice==2) || (choice==3) || (choice==4) || (choice==6) || (choice==8)
29     || (choice==10)
30     Rho(ke);
31 else if (choice==12)
32     Rho(alpha);
33 % Min-max of One parameter family
34 else if (choice==5) || (choice==9)
35     opt=optimset('TolX',1e-12,'TolFun',1e-12);
36
37 % change p if one wants asymptotic behavior
38 p=0:0;
39 KE=0;
40 for i=1:length(p)
41     % comment delta for non-overlapping methods
42     delta=0.1./(1.*10^p(i))
43     if i==1
44         [ke,R]=fminsearch('Rho',ke,opt);
45     else
46         [ke,R]=fminsearch('Rho',KE(i-1),opt);
47     end;
48     RR(i)=R; LL(i)=delta; KE(i)=ke
49 end;
50
51 if (length(p)>1) || (choice==5)
52     % Behaviour of ke^* w.r.t. the overlap
53     loglog(LL,omega/cs-KE,'-o',LL,0.04*LL.^1,'-',LL,0.35.*LL.^1.5,'-',LL,6200.*LL.^2,'-')
54     legend1 = legend('\omega/C_s-k_e^*', 'O(\delta)', 'O(\delta^{3/2})', 'O(\delta^2)');
55     set(legend1,'Location','southeast'); xlabel('\delta');
56     set(gca,'FontSize',10); print -depsc keoptexp2.eps;
57
58 % Behaviour of the convergence factor w.r.t. the overlap
59 figure;
60 loglog(LL,1-RR,'-o',LL,0.7.*LL.^1,'-')
61 legend1 = legend('1-R', 'O(\delta)');
62 set(legend1,'Location','north'); xlabel('\delta');

```

```

63     set(gca,'FontSize',10); print -depsc keoptexp3.eps;
64 end;
65
66 % Min-max of Two-Sided methods
67 else if (choice==7) || (choice==11)
68     opt=optimset('TolX',1e-12,'TolFun',1e-12);
69     % change p if one wants asymptotic behavior
70     p=0:0;
71     KE2=zeros(length(p),2);
72     for i=1:length(p)
73         % comment delta for non-overlapping methods
74         delta=0.1*1.3^p(i)
75         if i==1
76             [ke,R]=fminsearch('Rho',ke,opt);
77         else
78             [ke,R]=fminsearch('Rho',KE2(i-1,:),opt);
79         end;
80         RR(i)=R; LL(i)=delta; KE2(i,:)=ke
81     end;
82
83 % Min-max of general higher
84 else if (choice==13)
85     opt=optimset('TolX',1e-12,'TolFun',1e-12);
86     % change p if one wants asymptotic behavior
87     p=0:0;
88     ALPHA=zeros(length(p),2);
89     for i=1:length(p)
90         % comment delta for non-overlapping methods
91         delta=0.1*1.3^p(i)
92         if i==1
93             [alpha,R]=fminsearch('Rho',alpha,opt);
94         else
95             [alpha,R]=fminsearch('Rho',ALPHA(i-1,:),opt);
96         end;
97         RR(i)=R; LL(i)=delta; ALPHA(i,:)=alpha
98     end;
99 end;

```

## A.2 Rho - the computation of the convergence factor

```

1 function R=Rho(Ke)
2 global omega cs cp rho kmin kmax delta L choice order;
3
4 if (choice==3) || (choice==6) || (choice==7) || (choice==10) || (choice==11)

```

```

5   ke1=Ke(1); ke2=Ke(2);
6   else if (choice==12) || (choice==13)
7       alpha1=Ke(1); alpha2=Ke(2);
8   else
9       ke=Ke(1);
10  end;
11
12  % Grid
13  if (choice==5) || (choice==7) || (choice==9) || (choice==11) || (choice==13)
14      N=200; k1=omega./cp; k2=omega./cs; dk=pi./L;
15      kk=[kmin:(k1-dk-kmin)./N:k1-dk k1+dk:(k2-dk-(k1+dk))./N:k2-dk k2+dk:(kmax-(k2+dk))./N:
          kmax];
16  else
17      N=1000; kk=kmin:(kmax-kmin)/N:kmax;
18  end;
19
20  mu=rho*cs^2; lambda=rho*(cp^2-2*cs^2); k_bar=0; k_bar2=0; f_bar=0; temp=0; temp2=0;
21
22  % Classical
23  if (choice==1)
24      for j=1:length(kk)
25          k=kk(j);
26          lambda1=sqrt(k.^2-omega^2./cs^2); lambda2=sqrt(k.^2-omega^2./cp^2);
27          X1=(k.^2+lambda1.*lambda2)./(k.^2-lambda1.*lambda2); X2=-1i.*(2.*k.*lambda2)./(k.^2-
          lambda1.*lambda2);
28          G=[exp(-delta.*(lambda1+lambda2)).*X2^2.*lambda1./lambda2+exp(-2.*delta.*lambda1).*X1
          ^2 X1.*X2.*(-exp(-delta.*(lambda1+lambda2))+exp(-2.*delta.*lambda1))
29          X1.*X2.*lambda1./lambda2.*(exp(-delta.*(lambda1+lambda2))-exp(-2.*delta.*lambda2))
          exp(-delta.*(lambda1+lambda2)).*X2^2.*lambda1./lambda2+exp(-2.*delta.*lambda2).*X1
          ^2];
30          e=eig(G); R1(j)=e(1); R2(j)=e(2);
31      end;
32  else
33      for j=1:length(kk)
34          k=kk(j); lambda1=sqrt(k.^2-omega^2./cs^2); lambda2=sqrt(k.^2-omega^2./cp^2);
35
36          % Taylor
37          if (choice==2)
38              if (order==0)
39                  sigma111=1i*rho*omega*cp; sigma112=0; sigma121=0; sigma122=1i*rho*omega*cs;
40              else if (order==1)
41                  sigma111=1i*rho*omega*cp+1i*rho*cp^2/(2*omega)*(cp-2*cs)*ke^2;
42                  sigma112=-1i*rho*cs*(cp-2*cs)*ke; sigma121=-sigma112;
43                  sigma122=1i*rho*omega*cs+1i*rho*cs^2/(2*omega)*(cs-2*cp)*ke^2;
44              else if (order==2)

```

```

45     sigma111=li*rho*omega*cp+li*rho*cp^2/(2*omega)*(cp-2*cs)*k^2;
46     sigma112=-li*rho*cs*(cp-2*cs)*k; sigma121=-sigma112;
47     sigma122=li*rho*omega*cs+li*rho*cs^2/(2*omega)*(cs-2*cp)*k^2;
48     end;
49     sigma211=sigma111; sigma212=-sigma112; sigma221=-sigma121; sigma222=sigma122;
50
51 % Two-sided Taylor
52 else if (choice==3)
53     sigma111=li*rho*omega*cp+li*rho*cp^2/(2*omega)*(cp-2*cs)*ke1^2;
54     sigma112=-li*rho*cs*(cp-2*cs)*ke1; sigma121=-sigma112;
55     sigma122=li*rho*omega*cs+li*rho*cs^2/(2*omega)*(cs-2*cp)*ke1^2;
56     sigma211=li*rho*omega*cp+li*rho*cp^2/(2*omega)*(cp-2*cs)*ke2^2;
57     sigma212=li*rho*cs*(cp-2*cs)*ke2; sigma221=-sigma212;
58     sigma222=li*rho*omega*cs+li*rho*cs^2/(2*omega)*(cs-2*cp)*ke2^2;
59
60 % One parameter family
61 else if (choice==4) || (choice==5)
62     var1_1=sqrt(ke.^2-omega^2./cs^2); var2_1=sqrt(ke.^2-omega^2./cp^2);
63     var1_2=var1_1; var2_2=var2_1; K_1=ke; K_2=K_1;
64
65 % Two-sided one parameter family
66 else if (choice==6) || (choice==7)
67     var1_1=sqrt(ke1.^2-omega^2./cs^2); var2_1=sqrt(ke1.^2-omega^2./cp^2); K_1=ke1;
68     var1_2=sqrt(ke2.^2-omega^2./cs^2); var2_2=sqrt(ke2.^2-omega^2./cp^2); K_2=ke2;
69
70 % Higher conditions
71 else if (choice==8) || (choice==9)
72     var1_1=sqrt(ke.^2-omega^2./cs^2); var2_1=sqrt(ke.^2-omega^2./cp^2);
73     var1_2=var1_1; var2_2=var2_1; K_1=k; K_2=K_1;
74
75 % Two-sided Higher conditions
76 else if (choice==10) || (choice==11)
77     var1_1=sqrt(ke1.^2-omega^2./cs^2); var2_1=sqrt(ke1.^2-omega^2./cp^2); K_1=k;
78     var1_2=sqrt(ke2.^2-omega^2./cs^2); var2_2=sqrt(ke2.^2-omega^2./cp^2); K_2=k;
79
80 % General Higher conditions
81 else if (choice==12) || (choice==13)
82     var1_1=alpha1.*(1+1i); var2_1=alpha2.*(1+1i);
83     var1_2=var1_1; var2_2=var2_1; K_1=k; K_2=k;
84     end;
85
86 % General Formula
87 if (choice>2)
88     sigma111=rho.*omega.^2.*var1_1./(K_1.^2-var1_1.*var2_1);
89     sigma112=li.*K_1.*rho.*(2.*cs^2-omega^2./(K_1.^2-var1_1.*var2_1));

```

```

90     sigma121=-sigma112; sigma122=rho.*omega^2.*var2_1./(K_1.^2-var1_1.*var2_1);
91     sigma211=rho.*omega.^2.*var1_2./(K_2.^2-var1_2.*var2_2);
92     sigma212=-1i.*K_2.*rho.*(2.*cs^2-omega^2./(K_2.^2-var1_2.*var2_2));
93     sigma221=-sigma212; sigma222=rho.*omega^2.*var2_2./(K_2.^2-var1_2.*var2_2);
94     end;
95
96     % Matrices from Interface iterations
97     A1=[exp(lambda1*delta)*(2*lambda1*mu+sigma111+(1i*sigma112*lambda1)/k) exp(lambda2*
          delta)*(1i*k*lambda+sigma112-(1i*2*lambda2^2*mu+1i*lambda2^2*lambda+1i*sigma111*
          lambda2)/k)
98     exp(lambda1*delta)*(1i*mu*k+sigma121+(1i*lambda1^2*mu+1i*sigma122*lambda1)/k) exp(
          lambda2*delta)*(2*lambda2*mu+sigma122-(1i*sigma121*lambda2)/k)];
99     A2=[exp(-lambda1*delta)*(-2*lambda1*mu+sigma111-(1i*sigma112*lambda1)/k) exp(-
          lambda2*delta)*(1i*k*lambda+sigma112-(1i*2*lambda2^2*mu+1i*lambda2^2*lambda-1i*
          sigma111*lambda2)/k)
100    exp(-lambda1*delta)*(1i*mu*k+sigma121+(1i*lambda1^2*mu-1i*sigma122*lambda1)/k) exp(-
          lambda2*delta)*(-2*lambda2*mu+sigma122+(1i*sigma121*lambda2)/k)];
101     B1=[-2*lambda1*mu+sigma211+(1i*sigma212*lambda1)/k -1i*k*lambda+sigma212+(1i*lambda2
          ^2*lambda+1i*2*lambda2^2*mu-1i*sigma211*lambda2)/k
102     -1i*mu*k+sigma221+(-1i*lambda1^2*mu+1i*sigma222*lambda1)/k -2*lambda2*mu+sigma222-(1i
          *sigma221*lambda2)/k];
103     B2=[2*lambda1*mu+sigma211-(1i*sigma212*lambda1)/k -1i*k*lambda+sigma212+(1i*2*
          lambda2^2*mu+1i*lambda2^2*lambda+1i*sigma211*lambda2)/k
104     -1i*mu*k+sigma221+(-1i*lambda1^2*mu-1i*sigma222*lambda1)/k 2*lambda2*mu+sigma222+(1i*
          sigma221*lambda2)/k];
105
106     % Iteration matrix and its eigenvalues
107     G=inv(A1)*A2*inv(B2)*B1; e=eig(G); R1(j)=e(1); R2(j)=e(2);
108
109     % Abscisses from k^* and k bar (Lemmas 2.2 and 2.3)
110     if (choice==2)
111         f_bar_2 = max(abs(e(1)),abs(e(2)));
112         if (f_bar_2 > f_bar)
113             k_bar = k; f_bar = f_bar_2;
114         elseif (f_bar_2 >= f_bar)
115             k_bar2 = k;
116         end;
117     end;
118
119     % peak spot
120     if (choice==5) || (choice==7) || (choice==9) || (choice==11) || (choice==13)
121         if (k>omega./cs+5.*dk)
122             temp2=max(abs(e(1)),abs(e(2)));
123             if (temp2 > temp)
124                 k_temp = j; temp = temp2;

```

```

125     end;
126     end;
127     end;
128     end;
129 end;
130
131 % Values of the assumed maximums
132 if (choice==5)
133     R=max([abs(R2(N+1)) abs(R2(N+2)) abs(R1(2.*N+3))]);
134 else if (choice==7) || (choice==9) || (choice==11)
135     R=max([abs(R2(N+1)) abs(R2(N+2)) abs(R1(2.*N+1)) abs(R1(2.*N+3)) abs(R1(k_temp))]);
136 else if (choice==13)
137     R=max([abs(R2(N+1)) abs(R2(N+2)) abs(R1(2.*N+1)) abs(R1(2.*N+3)) abs(R1(k_temp)) abs(
138         R2(k_temp))]);
139 end;
140
141 % PLOTS -----
142 if (choice==1)
143     plot(kk,abs(R1),'- ',kk,abs(R2),'r- ');
144     semilogx(kk,max(abs(R1),abs(R2)),'- ');
145     title("Spectrum of Classical Schwarz");
146     legend1 = legend('classical');
147     grid on
148
149 % Spectrum of the iteration operator (Classical Schwarz)
150 figure;
151 plot(1-real(R1),imag(R1));
152 set(gca,'FontSize',10);
153 hold on
154 plot(1-real(R2),imag(R2));
155 hold on
156 t = 0:0.05:2.*pi;
157 plot(1+cos(t),sin(t));
158 legend1 = legend('r_+', 'r_-'); set(legend1,'Location','southwest');
159 title("Preconditioner test");
160 print -depsc precon.eps;
161 else if (choice==2)
162     semilogx(kk,max(abs(R1),abs(R2)),'- ');
163     %title("Spectrum of Schwarz with Taylor");
164     legend1 = legend('\rho-T');
165 else if (choice==3)
166     semilogx(kk,max(abs(R1),abs(R2)),'- ');
167     title("Spectrum of Schwarz with two-sided Taylor");
168     legend1 = legend('\rho_{T.1}');
169 else if (choice==4)

```

```

169 semilogx(kk,max(abs(R1),abs(R2)),'-');
170 title("Spectrum of Schwarz with one parameter family");
171 legend1 = legend('\rho_E');
172 else if (choice==5)
173 semilogx(kk(1:N+1),abs(R1(1:N+1)),'-b',kk(1:N+1),abs(R2(1:N+1)),'-r',kk(N+1),abs(R2(N
+1)),'*',kk(N+2),abs(R2(N+2)), '*',kk(2*N+3),abs(R1(2*N+3)), '*',kk(N+2:2*N+2),abs(
R1(N+2:2*N+2)),'-b',kk(2*N+3:end),abs(R1(2*N+3:end)),'-b',kk(N+2:2*N+2),abs(R2(N
+2:2*N+2)),'-r',kk(2*N+3:end),abs(R2(2*N+3:end)),'-r')
174 line([omega/cs omega/cs],[0 R]); line([omega/cp omega/cp],[0 R])
175 title("Spectrum of Schwarz with one parameter family");
176 legend1 = legend('r_-','r_+','max_1','max_2','max_3');
177 else if (choice==6)
178 semilogx(kk,max(abs(R1),abs(R2)),'-');
179 title("Spectrum of Schwarz with two-sided one parameter family");
180 legend1 = legend('\rho_E two-sided');
181 else if (choice==7)
182 semilogx(kk(1:N+1),abs(R1(1:N+1)),'-b',kk(1:N+1),abs(R2(1:N+1)),'-r',kk(N+1),abs(R2(N
+1)), '*',kk(N+2),abs(R2(N+2)), '*',kk(2*N+1),abs(R1(2*N+1)), '*',kk(2*N+3),abs(R1(2*
N+3)), '*',kk(k.temp),abs(R1(k.temp)), '*',kk(N+2:2*N+2),abs(R1(N+2:2*N+2)),'-b',kk
(2*N+3:end),abs(R1(2*N+3:end)),'-b',kk(N+2:2*N+2),abs(R2(N+2:2*N+2)),'-r',kk(2*N
+3:end),abs(R2(2*N+3:end)),'-r')
183 line([omega/cs omega/cs],[0 R]); line([omega/cp omega/cp],[0 R])
184 title("Spectrum of Schwarz with two-sided one parameter family");
185 legend1 = legend('r_-','r_+','max_1','max_2','max_3','max_4','max_5');
186 else if (choice==8)
187 semilogx(kk,max(abs(R1),abs(R2)),'-');
188 title("Spectrum of Schwarz with higher conditions");
189 legend1 = legend('\rho_H');
190 else if (choice==9)
191 semilogx(kk(1:N+1),abs(R1(1:N+1)),'-b',kk(1:N+1),abs(R2(1:N+1)),'-r',kk(N+1),abs(R2(N
+1)), '*',kk(N+2),abs(R2(N+2)), '*',kk(2*N+1),abs(R1(2*N+1)), '*',kk(2*N+3),abs(R1(2*
N+3)), '*',kk(k.temp),abs(R1(k.temp)), '*',kk(N+2:2*N+2),abs(R1(N+2:2*N+2)),'-b',kk
(2*N+3:end),abs(R1(2*N+3:end)),'-b',kk(N+2:2*N+2),abs(R2(N+2:2*N+2)),'-r',kk(2*N
+3:end),abs(R2(2*N+3:end)),'-r')
192 line([omega/cs omega/cs],[0 R]); line([omega/cp omega/cp],[0 R])
193 title("Spectrum of Schwarz with higher conditions");
194 legend1 = legend('r_-','r_+','max_1','max_2','max_3','max_4','max_5');
195 else if (choice==10)
196 semilogx(kk,max(abs(R1),abs(R2)),'-');
197 title("Spectrum of Schwarz with two-sided higher conditions");
198 legend1 = legend('\rho_H two-sided');
199 else if (choice==11)
200 semilogx(kk(1:N+1),abs(R1(1:N+1)),'-b',kk(1:N+1),abs(R2(1:N+1)),'-r',kk(N+1),abs(R2(N
+1)), '*',kk(N+2),abs(R2(N+2)), '*',kk(2*N+1),abs(R1(2*N+1)), '*',kk(2*N+3),abs(R1(2*
N+3)), '*',kk(k.temp),abs(R1(k.temp)), '*',kk(N+2:2*N+2),abs(R1(N+2:2*N+2)),'-b',kk

```

```

(2*N+3:end),abs(R1(2*N+3:end)), '-b',kk(N+2:2*N+2),abs(R2(N+2:2*N+2)), '-r',kk(2*N
+3:end),abs(R2(2*N+3:end)), '-r')
201 line([omega/cs omega/cs],[0 R]); line([omega/cp omega/cp],[0 R])
202 title("Spectrum of Schwarz with two-sided higher conditions");
203 legend1 = legend('r_-','r_+','max_1','max_2','max_3','max_4','max_5');
204 else if (choice==12)
205 semilogx(kk,max(abs(R1),abs(R2)), '-');
206 title("Spectrum of Schwarz with General higher conditions");
207 legend1 = legend('\rho_G');
208 else if (choice==13)
209 semilogx(kk(1:N+1),abs(R1(1:N+1)), '-b',kk(1:N+1),abs(R2(1:N+1)), '-r',kk(N+1),abs(R2(N
+1)), '* ',kk(N+2),abs(R2(N+2)), '* ',kk(2*N+1),abs(R1(2*N+1)), '* ',kk(2*N+3),abs(R1(2*
N+3)), '* ',kk(k_temp),abs(R1(k_temp)), '* ',kk(k_temp),abs(R2(k_temp)), '* ',kk(N+2:2*N
+2),abs(R1(N+2:2*N+2)), '-b',kk(2*N+3:end),abs(R1(2*N+3:end)), '-b',kk(N+2:2*N
+2),abs(R2(N+2:2*N+2)), '-r',kk(2*N+3:end),abs(R2(2*N+3:end)), '-r')
210 line([omega/cs omega/cs],[0 R]); line([omega/cp omega/cp],[0 R])
211 title("Spectrum of Schwarz with General higher conditions");
212 legend1 = legend('r_-','r_+');
213 end;
214 set(legend1,'Location','northeast'); xlabel('k');
215 grid on
216 set(gca,'FontSize',10);
217 print -depsc keoptexp.eps;
218 drawnow

```

## Appendix B

# FreeFem++ implementations

In this section we discuss the FreeFem++ implementation of the methods from Chapter 1 and show the main parts of codes we used. All the details about the choice the solver, discrete spaces, boundary conditions, are described in the exhaustive comments from the codes. We use the build-in polynomial spaces such like space of all polynomials degree one P1.

Note that the data script contains a certain number of macros, whose general syntax is

```
macro <identifier>(<parameter list>) <replacement token list> //
```

where `<parameter list>` is optional; this will make it possible to replace every subsequent occurrence of `<identifier>()` with `<replacement token list>`, by using the passed arguments if `<parameter list>` is present in the macro definition. This use of macros permits to use the same scripts for different (two or three dimensional) problems, by changing only the data script.

The main program needs the routines (of `decomp.idp` and `createPartitionVec.idp`) to create a decomposition of the domain and to build the restriction and partition of unity matrices. We do not include here the files `decomp.idp` and `createpartitionVec.idp` as they can be found in [DJN15].

The matrices from the local problems are used in the constructuon of the preconditioner for the (complex) GMRES method called to solve the problem; in particular the `GMRES-left.idp` routine requires the matrix-vector product with the problem matrix and with the preconditioner.

We start by the test cases 1 and 2.

## B.1 Data files and definitions of macros

The data file used for both is dataNavier.edp

```

1 load "metis"
2 load "medit"
3
4 string method = "ORAS"; // preconditioner RAS or ORAS
5 int nn=2, mm=1; // number of the domains in each direction
6 int npart = nn*mm; // total number of domains
7 bool withmetis = 0; // =1 (Metis decomp) =0 (uniform decomp)
8 int sizeovr=8; // size of the overlap
9 int nloc = 40; // local no of dof per domain in one direction
10 string prob="square"; // test case considered
11 mesh Th;
12 real allong;
13 // Boundary conditions:
14 // Q = -1 Dirichlet, Q = 1 Traction, Q = 0 Robin
15 real Q = 0, Qi = 0 ;
16
17 // Testcase 1
18 real Cp=1, Cs=0.5;
19 real rho = 1;
20 real lambda = rho*(Cp^2-2*Cs^2), mu = Cs^2*rho;
21 real omega = 5;
22
23 // Testcase 2
24 /*real rho = 7800; // density
25 real E = 2.*10^11, nu = 0.3; // Poisson ratio/ Young's modulus
26 real mu= E/(2*(1+nu)); // Lamé coefficients
27 real lambda = (E*nu)/((1+nu)*(1-2*nu));
28 real Cp = sqrt((lambda+2*mu)/rho), Cs = sqrt(mu/rho); // P & S-wave
29 real f = 20000, omega = 2*pi*f; // frequency, pulsation
30
31 real Kp = omega/Cp, Ks = Kp*Cp/Cs; // wavenumber of P & S-waves
32
33 // Incident wave
34 real alpha = 1., beta = 1.; // coef. incident waves
35 real xcs = pi/3;
36 real sqrt2 = sqrt(2.);
37 func uinc = alpha*cos(xcs)*exp(1i*Kp*(cos(xcs)*x+sin(xcs)*y)) +
38 beta*sin(xcs)*exp(1i*Ks*(cos(xcs)*x+sin(xcs)*y));
39 func vinc = alpha*sin(xcs)*exp(1i*Kp*(cos(xcs)*x+sin(xcs)*y)) -
40 beta*cos(xcs)*exp(1i*Ks*(cos(xcs)*x+sin(xcs)*y));
41

```

```

42 int[int] chlab=[1,1 ,2,2 ,3,1 ,4,2]; //Robin conditions for label = 2
43 macro Grad(u) [dx(u),dy(u)] // EOM
44 macro epsilon(u,v) [dx(u),dy(v),(dy(u)+dx(v))/sqrt2] // EOM
45 macro div(u,v) (dx(u)+dy(v)) // EOM
46
47 // Components of the Sigma tensor
48 macro sxx() rho*omega*(Cp*N.x^2+Cs*N.y^2) //EOM
49 macro sxy() rho*omega*(Cp-Cs)*N.x*N.y //EOM
50 macro syy() rho*omega*(Cp*N.y^2+Cs*N.x^2) //EOM
51
52 // Iterative solver parameters
53 real tol=1e-6; // tolerance for the iterative method
54 int maxit=60; // maximum number of iterations

```

We also need to define the domain decomposition data structures and the global variational formulation as shown in `defNavier.edp`

```

1 // Definition ingredients - numerical solution of Navier equations
2 // Mesh of a rectangular domain
3 if(prob == "square"){
4   allong = real(nn)/real(mm); // aspect ratio of the global domain
5   Th=square(nn*nloc,mm*nloc,[x*allong,y]);
6 }
7 func bint = (x>=0) && (x<=allong) && (y>0) && (y<1) ;
8 func brd = 1-bint;
9
10 fespace Ph(Th,P0);
11 fespace Vh(Th,[P1,P1]); // vector fem space
12 fespace Uh(Th,P1); // scalar fem space
13 Ph part; // piecewise constant function
14 int[int] lpart(Ph.ndof); // giving the decomposition
15
16 // Domain decomposition data structures
17 mesh[int] aTh(npart),aTh0(npart); // sequence of ovr. meshes
18 matrix<complex>[int] Rih(npart); // local restriction operators
19 matrix<complex>[int] Dih(npart); // partition of unity operators
20 matrix[int] Dihreal(npart),Rihreal(npart),Dihreal0(npart),Rihreal0(npart),Dihrealr(npart)
21 ;
22 int[int] Ndeg(npart),Ndeg0(npart); // number of dof for each mesh
23 real[int] AreaThi(npart),AreaThi0(npart); // area of each subdomain
24 matrix<complex>[int] aA(npart),aR(npart); // local Dirichlet/Robin matrices
25
26 // Definition of the problem to solve
27 Th=change(Th,refe=chlab);
28 Vh [intern,iintern] = [bint,bint];

```

```

28 | Vh [bord , bbord ] = [ brd , brd ];
29 |
30 | // Traction operator applied to uinc
31 | Vh<complex> [ ui , vi ]=[ uinc , vinc ];
32 | macro gu () ( 2*mu*( dx ( ui ) * N . x + dy ( ui ) * N . y ) +
33 |             lambda * div ( ui , vi ) * N . x + mu * ( dx ( vi ) - dy ( ui ) ) * N . y ) //EOM
34 | macro gv () ( 2*mu*( dx ( vi ) * N . x + dy ( vi ) * N . y ) +
35 |             lambda * div ( ui , vi ) * N . y - mu * ( dx ( vi ) - dy ( ui ) ) * N . x ) //EOM
36 |
37 | // global variational formulation
38 | Vh<complex> [ rhsglobal , rrhsglobal ] , [ uglob , uuglob ] , [ u , v ] , [ uu , vv ];
39 | macro Navier ( u , v , uu , vv ) rho * omega ^ 2 * ( u * uu + v * vv ) -
40 |             lambda * ( div ( u , v ) * div ( uu , vv ) ) - 2 . * mu * ( epsilon ( u , v ) ' * epsilon ( uu , vv ) ) // EOM
41 | varf vglobal ( [ u , v ] , [ uu , vv ] ) = int 2d ( Th ) ( Navier ( u , v , uu , vv )
42 |             + int 1d ( Th , 2 ) ( 1 i * ( 1 - Q ) / ( 1 + Q ) * ( sxx * u * uu + sxy * ( v * uu + u * vv ) + syy * v * vv )
43 |             - int 1d ( Th , 2 ) ( gu * uu + gv * vv - 1 i * ( 1 - Q ) / ( 1 + Q ) * ( sxx * ui * uu + sxy * ( vi * uu + ui * vv ) + syy * vi
44 |             * vv )
45 |             + on ( 1 , u = ui ) + on ( 1 , v = vi ) );
46 | matrix < complex > Aglobal ;

```

## B.2 RAS/ORAS

The main script file for the iterative versions of RAS and ORAS algorithms is `Solver-Navier.edp`

```

1 | /*# debutPartition #*/
2 | include ". / dataNavier . edp "
3 | include ". / defNavier . edp "
4 | // include ". / dataNavier - transmission . edp "
5 | // include ". / defNavier - transmission . edp "
6 | include ". / decomp . idp "
7 | include ". / createPartitionVec . idp "
8 | SubdomainsPartitionUnityVec ( Th , part [ ] , sizeovr , aTh , Rih real , Dih real , Ndeg , AreaThi ) ;
9 | /* Build a new partition of unity
10 | SubdomainsPartitionUnityVec ( Th , part [ ] , 1 , aTh0 , Rih real 0 , Dih real 0 , Ndeg0 , AreaThi0 ) ;
11 | for ( int i = 0 ; i < npart ; i ++ ) {
12 |     matrix Maux1 , Maux2 , Maux3 ;
13 |     Maux1 = Rih real 0 [ i ] * Rih real [ i ] ' ;
14 |     Maux2 = Dih real 0 [ i ] * Maux1 ;
15 |     Maux3 = Rih real 0 [ i ] ' * Maux2 ;
16 |     Dih real r [ i ] = Rih real [ i ] * Maux3 ;
17 | } */
18 |
19 | for ( int i = 0 ; i < npart ; i ++ ) {
20 |     Rih [ i ] = Rih real [ i ] ;

```

```

21     Dih[i] = Dihreal[i];
22     /* test the partition of unity
23     Vh<complex> [ux,uux],[vx,vvx];
24     ux[]=1.;
25     matrix Maux1, Maux2;
26     Maux1 = Dihreal[i]*Rihreal[i];
27     Maux2 = Rihreal[i]’*Maux1;
28     vx[] = Maux2*ux[];
29     plot(vx, value=1, fill=1, dim=3, wait=1);*/
30 }
31
32 /*# endPartition #*/
33 /*# debutGlobalData #*/
34 Aglobal = vglobal(Vh,Vh, solver = UMFPACK); // global matrix
35 rhsglobal[] = vglobal(0,Vh); // global rhs
36 uglob[] = Aglobal^-1*rhsglobal[];
37 plot(uglob, wait=1, fill=1, dim=3, ps="GlobalSolution");
38 /*# finGlobalData #*/
39
40 /*# debutLocalData #*/
41 for(int i = 0; i<npart; ++i){
42     mesh Thi = aTh[i];
43     fespace Vhi(Thi, [P1, P1]);
44     cout << " Domain :" << i << "/" << npart << endl;
45     if (method == "ORAS"){
46         varf valocal([u,v],[uu,vv]) = int2d(Thi)(Navier(u,v,uu,vv))
47             + int1d(Thi,2)(1i*(1-Q)/(1+Q)*(sxx*u*uu+sxy*(v*uu+u*vv)+syy*v*vv))
48             + int1d(Thi,10)(1i*(1-Qi)/(1+Qi)*(sxx*u*uu+sxy*(v*uu+u*vv)+syy*v*vv))
49             + on(1, u=ui, v=vi);
50         aR[i] = valocal(Vhi, Vhi, solver = UMFPACK);
51     }
52     if (method == "RAS"){
53         matrix<complex> temp = Aglobal*Rih[i]’;
54         aR[i] = Rih[i]*temp;
55         set(aR[i], solver = UMFPACK);
56     }
57 }
58 /*# finLocalData #*/
59 /*# debutSchwarzIter #*/
60 ofstream filei(method+"_Iter_ovr"+sizeovr+"-w"+omega+".m");
61 Vh<complex> [un,uun] = [0,0]; // initial guess
62 Vh<complex> [rn,rrn] = [rhsglobal,rrhsglobal];
63 Vh<complex> [er,eer],[dr,ddr];
64 for(int iter = 0; iter<maxit; ++iter)
65     {

```

```

66   real err = 0, res;
67   [dr, ddr] = [0,0];
68   for(int i = 0; i < npart; ++i)
69     {
70       complex[int] bi = Rih[i]*rn[]; // restriction to the local domain
71       complex[int] ui = aR[i] ^-1 * bi; // local solve
72       bi = Dih[i]*ui;
73       dr[] += Rih[i]'*bi;
74     }
75   un[] += dr[]; // build new iterate
76   rn[] = Aglobal*un[]; // computes global residual
77   rn[] = rn[] - rhsglobal[];
78   rn[] *= -1;
79       er[] = un[] - uglob[];
80       //cout << "Error = " << er[][25] << endl;
81   err = er[] .l2 / uglob[] .l2;
82   res = rn[] .l2;
83   cout << "It: " << iter << " Residual = " << res << " Relative L2 Error = " << err
84       << endl;
85   Vh [abser, abseer] = [abs(er), abs(eer)];
86   plot(abser, value=1, dim=3, fill=1, wait=1, cmm="error");
87   int j = iter+1;
88   // Store the error and the residual in Matlab/Scilab/Octave form
89   filei << method+"Iter_ovr"+sizeovr+"_w"+omega+"("+j+")=" << err << ";" << endl;
90   if(err < tol) break;
91 }
92 //medit("Error", Th, abs(er));
93 /*# finSchwarzIter #*/

```

and of the preconditioned version is Precond-GMRES-Navier.edp

```

1  /*# debutPartition #*/
2  include "./dataNavier.edp"
3  include "./defNavier.edp"
4  // include "./dataNavier-transmission.edp"
5  // include "./defNavier-transmission.edp"
6  include "./decomp.idp"
7  include "./createPartitionVec.idp"
8  SubdomainsPartitionUnityVec(Th, part[], sizeovr, aTh, Rihreal, Dihreal, Ndeg, AreaThi);
9  /* Build a new partition of unity
10 SubdomainsPartitionUnityVec(Th, part[], 1, aTh0, Rihreal0, Dihreal0, Ndeg0, AreaThi0);
11 for (int i=0; i < npart; i++) {
12   matrix Maux1, Maux2, Maux3;
13   Maux1 = Rihreal0[i]*Rihreal[i]';
14   Maux2 = Dihreal0[i]*Maux1;
15   Maux3 = Rihreal0[i]'*Maux2;

```

```

16   Dihrealr[i] = Rihreal[i]*Maux3;
17 }*/
18 for (int i=0; i<npart; i++) {
19     Rih[i] = Rihreal[i];
20     Dih[i] = Dihreal[i];
21 }
22
23 /*# endPartition #*/
24 /*# debutGlobalData #*/
25 Aglobal = vglobal(Vh,Vh,solver = UMFPACK); // global matrix
26 rhsglobal [] = vglobal(0,Vh); // global rhs
27 uglob [] = Aglobal^-1*rhsglobal [];
28 //plot(uglob,wait=1,fill=1);
29 /*# finGlobalData #*/
30
31 /*# debutLocalData #*/
32 for(int i = 0;i<npart;++i){
33     mesh Thi = aTh[i];
34     fespace Vhi(Thi,[P1,P1]);
35     cout << " Domain :" << i << "/" << npart << endl;
36     if (method == "ORAS"){
37         varf valocal([u,v],[uu,vv]) = int2d(Thi)(Navier(u,v,uu,vv))
38             + int1d(Thi,2)(1i*(1-Q)/(1+Q)*(sxx*u*uu+sxy*(v*uu+u*vv)+syy*v*vv))
39             + int1d(Thi,10)(1i*(1-Qi)/(1+Qi)*(sxx*u*uu+sxy*(v*uu+u*vv)+syy*v*vv))
40             + on(1,u=ui,v=vi);
41         aR[i] = valocal(Vhi,Vhi,solver = UMFPACK);
42     }
43     if (method == "RAS"){
44         matrix<complex> temp = Aglobal*Rih[i]';
45         aR[i] = Rih[i]*temp;
46         set(aR[i],solver = UMFPACK);
47     }
48 }
49
50 /*# finLocalData #*/
51 /*# debutGMRESsolve #*/
52 include "GMRES.idp"
53 Vh<complex> [un,uun],[cbord,ccbord];
54 Vh<complex> [sol,ssol],[er,eer]; // initial guess, final solution and error
55 un[].re = intern[];
56 cbord[].re = bord[];
57 un[] += ui[].*cbord[]; // solution verifying the Dirichle BC
58
59 sol[] = GMRES(un[],tol,maxit);
60 Vh [solre,ssolre]= [real(sol),real(ssol)];

```

```

61 | Vh [solim ,ssolim]= [imag(sol) ,imag(ssol) ];
62 | //plot(solre ,dim=3,wait=1,value =1,fill=1);
63 | er [] = sol []-uglob [];
64 | cout << "Final scaled error = " << er [].linfo/uglob [].linfo << endl;
65 | /*# finGMRESSolve #*/

```

### B.3 GMRES

The details of the implementation of these preconditioners as well as the complex version of the Krylov solver used here (GMRES with a left preconditioning) are shown in `GMRES-left.idp`

```

1 | // Preconditioned GMRES algorithm Applied to the system
2 | //  $M^{-1}Aglobal x = M^{-1}b$ 
3 | // Here Aglobal denotes the global matrix
4 | //  $M^{-1}$  is the RAS preconditioner based on domain decomposition
5 | // In order to use the GMRES routine define first the matrix-vector product
6 | /*# debutGlobalMatvec #*/
7 | func complex[int] A(complex[int] &vec)
8 | {
9 |     // Matrix vector product with the global matrix
10 |     Vh<complex> [Ax, Axx];
11 |     Ax[]= Aglobal*vec;
12 |     return Ax[];
13 | }
14 | /*# finGlobalMatvec #*/
15 | /*# debutRASPrecond #*/
16 | // and the application of the preconditioner
17 | func complex[int] PREC(complex[int] &l)
18 | {
19 |     // Application of the preconditioner
20 |     //  $M^{-1}y = \sum Ri^T Di Ai^{-1} Ri y$ 
21 |     // Ri restriction operators , Ai local matrices
22 |     Vh<complex> [s,ss] = [0,0];
23 |     for(int i=0; i<npart; ++i) {
24 |         complex[int] bi = Rih[i]*l; // restricts rhs
25 |         complex[int] ui = aR[i] ^-1 * bi; // local solves
26 |         bi = Dih[i]*ui; // partition of unity
27 |         s [] += Rih[i] '*bi; // prolongation
28 |     }
29 |     return s [];
30 | }
31 | /*# finRASPrecond #*/
32 | /*# debutGMRESSolve #*/
33 | func complex[int] GMRES(complex[int] x0, real eps, int nbiter)

```

```

34 {
35     int intmetis = withmetis;
36     ofstream filei("conv_ovrl"+sizeovr+"_"+method+"_"+npart+"_"+part+intmetis+".m")
37         ;
38     Vh<complex> [r,rr], [z,zz], [v,vv],[w,ww],[er,eer], [un,uun];
39     Vh<complex>[int] [V,VV](nbiter); // orthonormal basis
40     complex[int,int] Hn(nbiter+2,nbiter+1); // Hessenberg matrix
41     Hn = 0.;
42     complex[int,int] rot(2,nbiter+2);
43     rot = 0.;
44     complex[int] g(nbiter+1),g1(nbiter+1);
45     g = 0.; g1 = 0.;
46     r[] = A(x0);
47     r[] -= rhsglobal[];
48     r[] *= -1.0;
49
50     z[] = PREC(r[]); // z = M^{-1}(b-A*x0)
51     g[0] = z[].l2; // initial residual norm
52
53     // filei << "relres("+1+")=" << g[0] << ";" << endl;
54     V[0][]=1/g[0]*z[]; // first basis vector
55     for(int it=0; it<nbiter; it++){
56         v[] = A(V[it][]);
57         w[] = PREC(v[]); // w = M^{-1}A*V_it
58         for(int i=0; i<it+1; i++) {
59             Hn(i,it) = w[]^i*v[it][];
60             w[] -= conj(Hn(i,it))*V[i][];
61         }
62         Hn(it+1,it) = w[].l2;
63         complex aux = Hn(it+1,it);
64         for(int i=0; i<it; i++){ // QR decomposition of Hn
65             complex aa = conj(rot(0,i))*Hn(i,it)+conj(rot(1,i))*Hn(i+1,it);
66             complex bb = -rot(1,i)*Hn(i,it)+rot(0,i)*Hn(i+1,it);
67             Hn(i,it) = aa;
68             Hn(i+1,it) = bb;
69         }
70         complex sq = sqrt( conj(Hn(it,it))*Hn(it,it) + Hn(it+1,it)*Hn(it+1,it) );
71         rot(0,it) = Hn(it,it)/sq;
72         rot(1,it) = Hn(it+1,it)/sq;
73         Hn(it,it) = conj(rot(0,it))*Hn(it,it)+conj(rot(1,it))*Hn(it+1,it);
74         Hn(it+1,it) = 0.;
75         g[it+1] = -rot(1,it)*g[it];
76         g[it] = conj(rot(0,it))*g[it];
77         complex[int] y(it+1); // Reconstruct the solution

```

```

78         for(int i=it; i>=0; i--) {
79             g1[i] = g[i];
80             for(int j=i+1; j<it+1; j++){
81                 g1[i] = g1[i]-Hn(i,j)*y[j];
82             }
83             y[i]=g1[i]/Hn(i,i);
84     }
85     un[] = x0;
86     for(int i=0;i<it+1;i++){
87         un[] = un[] + conj(y[i])*V[i][];
88     }
89     er[] = un[] - uglob[];
90     real relres = abs(g[it+1]);
91     real relerr = er[] .l2/uglob[] .l2;
92     cout << "It: " << it << " Residual = " << relres << " Relative L2 Error = " <<
93         relerr << endl;
94     int j = it+2;
95     int k = j-1;
96     filei << "relres("+k+")=" << relerr/*relres*/ << ";" << endl;
97     if(relerr < eps) {
98         cout << "GMRES has converged in " + it + " iterations " << endl;
99         cout << "Relative residual = " + relres << endl;
100        break;    }
101    V[it+1][]=1/aux*w[];
102    Vh [realsol ,rrealsol], [realer ,rrealer];
103    [realsol ,rrealsol] = [real(un) ,real(uun)];
104    [realer ,rrealer] = [real(er) ,real(eer)];
105    //plot(realer , dim=3, cmm="Error at step " + it , value=1, fill=1,wait=1);
106    }
107    return un[];
108 }
109 /*# finGMRESSolve #*/

```

# Bibliography

- [ABG12] X. Antoine, Y. Boubendir, and C. Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. *Journal of Computational Physics*, 231(2):262–280, 2012.
- [ARGG06] A. Alonso-Rodriguez and L. Gerardo-Giorda. New nonoverlapping domain decomposition methods for the harmonic Maxwell system. *SIAM J. Sci. Comput.*, 28(1):102–122, 2006.
- [BDG<sup>+</sup>17] M. Bonazzoli, V. Dolean, I. G. Graham, E.A. Spence, and P.-H. Tournier. Domain Decomposition preconditioning for the high-frequency time-harmonic Maxwell equations with absorption. *Submitted, arXiv:1711.03789*, 2017.
- [Ber94] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. of Comp.Phys.*, 114:185–200, 1994.
- [CDJP97a] F. Collino, G. Delbue, P. Joly, and A. Piacentini. A new interface condition in the non-overlapping domain decomposition. *Comput. Methods Appl. Mech. Engrg.*, 148:195–207, 1997.
- [CDJP97b] F. Collino, G. Delbue, P. Joly, and A. Piacentini. A new interface condition in the non-overlapping domain decomposition for the Maxwell equations Helmholtz equation and related optimal control. *Comput. Methods Appl. Mech. Engrg.*, 148:195–207, 1997.
- [Che98] P. Chevalier. *Méthodes numériques pour les tubes hyperfréquences. Résolution par décomposition de domaine*. PhD thesis, Université Paris VI, 1998.
- [CN98] P. Chevalier and F. Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition*

- methods*, 10 (Boulder, CO, 1997), pages 400–407. Amer. Math. Soc., Providence, RI, 1998.
- [CS99] X.-Ch. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21(2):792–797 (electronic), 1999.
- [CW94] W. C. Chew and W. H. Weedon. A 3d perfectly matched medium from modified maxwell’s equations with stretched coordinates. *IEEE Trans. Microwave Opt. Technol. Lett.*, 7:599–604, 1994.
- [Des90] B. Després. Décomposition de domaine et problème de Helmholtz. *C.R. Acad. Sci. Paris*, 1(6):313–316, 1990.
- [Des91] B. Després. *Méthodes de décomposition de domaine pour les problèmes de propagation d’ondes en régimes harmoniques*. PhD thesis, Paris IX, 1991.
- [DGG09] V. Dolean, L. Gerardo Giorda, and M. J. Gander. Optimized Schwarz methods for Maxwell equations. *SIAM J. Scient. Comp.*, 31(3):2193–2213, 2009.
- [DGL<sup>+</sup>12] O. Dubois, M. J. Gander, S. Loisel, A. St-Cyr, and D. B. Szyld. The optimized Schwarz method with a coarse grid correction. *SIAM J. Sci. Comput.*, 34(1):A421–A458, 2012.
- [DGL<sup>+</sup>13] V. Dolean, M. J. Gander, S. Lanteri, J.-F. Lee, and Z. Peng. Optimized Schwarz methods for curl-curl time-harmonic Maxwell’s equations. In Jocelyne Erhel, Martin J. Gander, Laurence Halpern, Taoufik Sassi, and Olof Widlund, editors, *Proceedings of the 21st international domain decomposition conference*. Springer LNCSE, 2013.
- [DGL<sup>+</sup>14] V. Dolean, M. J. Gander, S. Lanteri, J.-F. Lee, and Z. Peng. Effective transmission conditions for domain decomposition methods applied to the time-harmonic curl-curl maxwell’s equations. *Journal of Computational Physics*, 2014.
- [DJN15] V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015. Algorithms, theory, and parallel implementation.

- [DLP08a] V. Dolean, S. Lanteri, and R. Perrussel. A domain decomposition method for solving the three-dimensional time-harmonic Maxwell equations discretized by discontinuous Galerkin methods. *J. Comput. Phys.*, 227(3):2044–2072, 2008.
- [DLP08b] V. Dolean, S. Lanteri, and R. Perrussel. Optimized Schwarz algorithms for solving time-harmonic Maxwell’s equations discretized by a discontinuous Galerkin method. *IEEE. Trans. Magn.*, 44(6):954–957, 2008.
- [Dub07] O. Dubois. *Optimized Schwarz Methods for the Advection-Diffusion Equation and for Problems with Discontinuous Coefficients*. PhD thesis, McGill University, 2007.
- [EDG<sup>+</sup>11] M. El Bouajaji, V. Dolean, M. J. Gander, S. Lanteri, and R. Perrussel. Domain decomposition methods for electromagnetic wave propagation problems in heterogeneous media and complex domains. In *Domain Decomposition Methods in Science and Engineering XIX*, volume 78(1), pages 5–16. Springer LNCSE, 2011.
- [EDGL12a] M. El Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri. Comparison of a one and two parameter family of transmission conditions for Maxwell’s equations with damping. In *Domain Decomposition Methods in Science and Engineering XX*. Springer LNCSE, 2012. accepted for publication.
- [EDGL12b] M. El Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri. Optimized Schwarz methods for the time-harmonic Maxwell equations with damping. *SIAM J. Scient. Comp.*, 34(4):2048–2071, 2012.
- [EG03] E. Efstathiou and M. J. Gander. Why restricted additive Schwarz converges faster than additive Schwarz. *BIT*, 43(suppl.):945–959, 2003.
- [EG12] O. G. Ernst and M. J. Gander. Why it is difficult to solve Helmholtz problems with classical iterative methods. In *Numerical analysis of multiscale problems*, pages 325–363. Springer, 2012.
- [Fla01] E. Flauraud. *Méthode de décomposition de domaine pour des milieux poreux faillés*. PhD thesis, Paris VI, 2001. in preparation.
- [Gan06] M. J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.

- [Gan11] M. J. Gander. On the influence of geometry on optimized Schwarz methods. *SeMA J.*, 53(1):71–78, 2011.
- [GHM07a] M. J. Gander, L. Halpern, and F. Magoulès. An optimized schwarz method with two-sided robin transmission conditions for the helmholtz equation. *Int. J. for Num. Meth. in Fluids*, 55(2):163–175, 2007.
- [GHM07b] M. J. Gander, L. Halpern, and F. Magoules. An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. *International journal for numerical methods in fluids*, 55(2):163–175, 2007.
- [GK12] M. J. Gander and F. Kwok. Best Robin parameters for optimized Schwarz methods at cross points. *SIAM J. Sci. Comput.*, 34(4):A1849–A1879, 2012.
- [GMN02a] M. J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24(1):38–60, 2002.
- [GMN02b] M. J. Gander, F. Magoules, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 24(1):38–60, 2002.
- [GN00] M. J. Gander and F. Nataf. AILU: a preconditioner based on the analytic factorization of the elliptic operator. *Numer. Linear Algebra Appl.*, 7(7-8):505–526, 2000. Preconditioning techniques for large sparse matrix problems in industrial applications (Minneapolis, MN, 1999).
- [GN04] L. Gerardo Giorda and F. Nataf. Optimized Schwarz methods for unsymmetric layered problems with strongly discontinuous and anisotropic coefficients. Technical Report 561, CMAP, CNRS UMR 7641, Ecole Polytechnique, France, 2004. submitted.
- [Gra91] K.F. Graff. Wave motion in elastic solids. Dover, New York, 1991.
- [GSV17a] I. G. Graham, E. A. Spence, and E. Vainikko. Domain decomposition preconditioning for high-frequency Helmholtz problems with absorption. *Math. Comp.*, 86(307):2089–2127, 2017.
- [GSV17b] I. G. Graham, E. A. Spence, and E. Vainikko. *Recent Results on Domain Decomposition Preconditioning for the High-Frequency Helmholtz Equation Using Absorption*, pages 3–26. Geosystems Mathematics. Springer, 2017.

- [GX14] M. J. Gander and Y. Xu. Optimized Schwarz Methods for Circular Domain Decompositions with Overlap. *SIAM J. Numer. Anal.*, 52(4):1981–2004, 2014.
- [GZ16] M. J. Gander and H. Zhang. Optimized Schwarz methods with overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 38(5):A3195–A3219, 2016.
- [Hec12] F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.
- [HMCK04] T. Huttunen, P. Monk, F. Collino, and J. P. Kaipio. The ultra-weak variational formulation for elastic wave problems. *SIAM Journal on Scientific Computing*, 25(5):1717–1742, 2004.
- [Hog] M. Hogan. Ps waves. <http://www.colorado.edu/physics/phys2900/homepages/Marianne.Hogan/waves.html>. Accessed: 2017-09-07.
- [HTJ88] T. Hagstrom, R. P. Tewarson, and A. Jazcilevich. Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems. *Appl. Math. Lett.*, 1988.
- [JN00] C. Japhet and F. Nataf. The best interface conditions for domain decomposition methods: Absorbing boundary conditions. to appear in 'Artificial Boundary Conditions, with Applications to Computational Fluid Dynamics Problems' edited by L. Tounette, Nova Science, 2000.
- [JNR98] C. Japhet, F. Nataf, and F.-X. Roux. The Optimized Order 2 Method with a coarse grid preconditioner. application to convection-diffusion problems. In P. Bjorstad, M. Espedal, and D. Keyes, editors, *Ninth International Conference on Domain Decomposition Methods in Science and Engineering*, pages 382–389. John Wiley & Sons, 1998.
- [JNR01] C. Japhet, F. Nataf, and F. Rogier. The optimized order 2 method. application to convection-diffusion problems. *Future Generation Computer Systems FUTURE*, 18(1):17–30, 2001.
- [KK98] G. Karypis and V. Kumar. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Technical report, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN, 1998.

- [KL15] A. Karangelis and S. Loisel. Condition number estimates and weak scaling for 2-level 2-Lagrange multiplier methods for general domains and cross points. *SIAM J. Sci. Comput.*, 37(2):C247–C267, 2015.
- [Lio88] P.-L. Lions. On the Schwarz alternating method. I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987)*, pages 1–42. SIAM, Philadelphia, PA, 1988.
- [Lio90] P.-L. Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.
- [LMO00] G. Lube, L. Mueller, and F.-C. Otto. A non-overlapping domain decomposition method for the advection-diffusion problem. *Computing*, 64:49–68, 2000.
- [LNS15] S. Loisel, H. Nguyen, and R. Scheichl. Optimized Schwarz and 2-Lagrange multiplier methods for multiscale elliptic PDEs. *SIAM J. Sci. Comput.*, 37(6):A2896–A2923, 2015.
- [LVL05] S.-C. Lee, M. Vouvakis, and J.-F. Lee. A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays. *J. Comput. Phys.*, 203(1):1–21, 2005.
- [Nat96] F. Nataf. Absorbing boundary conditions in block Gauss-Seidel methods for convection problems. *Math. Models Methods Appl. Sci.*, 6(4):481–502, 1996.
- [NN97] F. Nataf and F. Nier. Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. *Numerische Mathematik*, 75(3):357–77, 1997.
- [NS02] R. Nabben and D. B. Szyld. Convergence theory of restricted multiplicative Schwarz methods. *SIAM J. Numer. Anal.*, 40(6):2318–2336 (electronic) (2003), 2002.
- [PELY13] J. Poulson, B. Engquist, S. Li, and L. Ying. A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations. *SIAM J. Sci. Comput.*, 35(3):C194–C212, 2013.

- [PL10] Z. Peng and J.-F. Lee. Non-conformal domain decomposition method with second-order transmission conditions for time-harmonic electromagnetics. *J. Comput. Phys.*, 229(16):5615–5629, 2010.
- [PRL10] Z. Peng, V. Rawat, and J.-F. Lee. One way domain decomposition method with second order transmission conditions for solving electromagnetic wave problems. *J. Comput. Phys.*, 229(4):1181–1197, 2010.
- [QV99] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, 1999.
- [RL10] V. Rawat and J.-F. Lee. Nonoverlapping domain decomposition with second order transmission condition for the time-harmonic Maxwell’s equations. *SIAM J. Sci. Comput.*, 32(6):3584–3603, 2010.
- [SBG96] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.
- [SCGT07] A. St-Cyr, M. J. Gander, and S. J. Thomas. Optimized multiplicative, additive, and restricted additive Schwarz preconditioning. *SIAM J. Sci. Comput.*, 29(6):2402–2425 (electronic), 2007.
- [Sch70] H. A. Schwarz. *Über einen Grenzübergang durch alternirendes Verfahren*. Zürcher u. Furrer, 1870.
- [SS86] Y. Saad and M. H. Schultz. GMRES: a Generalized Minimal Residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.
- [SZB<sup>+</sup>07] A. Schadle, L. Zschiedrich, S. Burger, R. Klose, and F. Schmidt. Domain decomposition method for Maxwell’s equations: scattering off periodic structures. *J. Comput. Phys.*, 226(1):477–493, 2007.
- [TW05] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, 2005.