Iterative Solution of the Helmholtz 
and PML Equation

Thesis submitted for the degree of 
[Doctor of Philosophy]

by

Ido Singer

Under the supervision of Prof. Eli Turkel

Submitted to the Senate of Tel-Aviv University
November 2002
This work was prepared under the supervision of

Prof. Eli Turkel

This work was supported by the Roberto Nemirovski Doctoral Fellowship in Mathematics and Computer science.
Acknowledgement:

I would like to deeply thank my advisor Professor Eli Turkel for the guidance, assistance, advice and support he gave me.

I would also like to thank my family that encouraged me to fulfill this work. Special thanks to my wife Orit who put up with me at home, taking care of our newborn child while I was spending time on the computer. Thanks also to my brother Amit who is the only member in our family with whom I can consult mathematics. Not only does he listen to you, but in many cases he even contributes good ideas.
Contents

1. Abstract ix
2. Introduction x

Chapter 1. A Perfectly Matched Layer for the Helmholtz Equation 1
   1. Derivation of the Helmholtz Equation 1
   2. Derivation of the PML 3

Chapter 2. High Order Accurate Finite Difference Schemes 7
   1. Finite Difference Schemes 7
   2. Divergence Form 13
   3. Numerical Results 21

Chapter 3. The Infinite Strip Problem 29
   1. Introduction to the Strip Problem 29
   2. Constructing the PML 32
   3. Minimizing the Error 36
   4. Exploring the Numerical Error 38
   5. The Modified Equation 46
   6. Numerical Results 51

Chapter 4. Krylov Subspace Methods for the Strip Problem 67
   1. Direct Methods for Linear Systems 68
   2. Iterative Methods for Sparse Linear Systems 70
   3. Preconditioned Iterations 83
   4. Choosing the Preconditioned Matrix 88
CONTENTS

5. Choosing the Optimal Algorithm 96
6. Conclusions 103

Bibliography 107
1. Abstract

A Perfectly Matched Layer (PML) has recently become a widespread technique used for solving waveguide problems both in the time domain and in the frequency domain. We develop a PML layer which is specially designed to solve the Helmholtz equation in an infinite two-dimensional strip. The Helmholtz-PML problem is then analyzed for the parameters that give the best performance. High order finite differences schemes for the Helmholtz equation are developed up to sixth order of accuracy. We show that the use of these high-order methods for the pure Helmholtz equation coupled with a standard second order approximation in the PML layer yields the same high-order accuracy in the overall problem. The use of these schemes on the Helmholtz-PML problem results in a complex linear systems. The matrix representation of this resulting system is analyzed and the system is solved using various Krylov-subspace methods. We analyze preconditioned algorithms designed for this problem. Numerical results are presented to support the analysis.
2. Introduction

In this thesis we describe how to solve the Helmholtz equation

\[ \Delta u + k^2 u = 0 \]

in an unbounded domain. In the first chapter we derive the Helmholtz equation from the Maxwell equations in frequency space. When \( k \) is a constant, integral equation methods can be carried out \([20]\) in fully unbounded domains in order to solve the equation, using boundary element formulations to the boundary integral form of the equation. Another approach is to apply asymptotic techniques \([19]\). These two methods have their drawbacks. The use of integral equations assume a constant \( k \) and involves the inverse of a full large matrix. The approximations using the asymptotic techniques are not accurate if the wave length is not small relative to other length scales in the problem. Thus, the common way to treat this problem is by numerically solving the partial differential equation.

To numerically solve a linear partial differential equation we need to discretize the space with a grid and approximate the equation with a numerical scheme. The most common techniques for this problem are the finite element and the finite difference methods. The result of any discretization is a linear system of equations \( Ax = b \). In order to get an accurate solution we need a one mesh, which depends on \( k \) \([5]\), and that leads to a matrix with many unknowns, though the matrix is usually sparse. In this work we use only the finite difference method. While the finite element method has its advantage when dealing with irregular geometries and inhomogeneous material media, the finite difference method is a simpler technique to apply and to analyze. In this work we develop and analyze only methods for Helmholtz
2. INTRODUCTION

\textit{\textbf{Introduction}}

The solution of partial differential equations with constant coefficients and in Cartesian coordinates. For non-constant coefficients and other geometries, including spherical coordinates, several changes should be made.

In Chapter 2 we develop finite difference schemes for the Helmholtz equation. We divide the space with a uniform grid, denoting $h$ for the grid-size. The most popular finite difference approach is the standard pointwise representation which is symmetric and is of order $O(h^2)$. High order schemes with fourth order of accuracy $O(h^4)$ were developed in [29]. These schemes, which can be used for variable function $k$ in the Helmholtz equation, are based on the Pad approximation and were examined in [24]. In [24] another approach was introduced to achieve higher order accuracy where the Helmholtz equation is used to replace higher order derivatives in the truncation error by lower order derivatives which can be approximated on a nine point stencil in two dimensions. In [28] this approach is used, when the value of $k$ is constant, to achieve a scheme with sixth order of accuracy $O(h^6)$. We extend the approach in [28] and improve it to get a family of sixth order accurate schemes which are easier to use in applications. We build schemes that preserve the divergence form of the equation and apply them on the PML-Helmholtz equation in an unbounded domain in the next chapters.

There are many high order finite difference schemes based on nine point approximations for the Poisson equation [26] and the Helmholtz equation [10]. The approach to get the sixth-order accurate scheme by Caruthers, Steinhoe and Engels, [10] is based on a discretization of the Green’s function. In this work we show that the scheme achieved there is a special case, to order $O(h^6)$, of the family of schemes we develop.
When we are dealing with an unbounded space, we cannot model this directly on a computer. Therefore we need to truncate the unbounded domain and impose a special artificial boundary condition (ABC) instead. There are many kinds of ABC’s, designed for wave, and wave-like equations. A survey of these techniques is to be found in [27]. Two efficient methods to deal with the impedance of the far field are the monolite elements technique [9] and the Dirichlet-to-Neumann map [18].

For many years the standard boundary condition was a local absorbing condition that was a generalization of the Sommerfeld radiation condition e.g. [6], but in recent years a number of models based on the PML (Perfectly Matched Layers) scheme have become popular. These layers minimize the reflections caused by the artificial boundary.

Berenger [7],[8] was the first to introduce a PML for the time dependent Maxwell equations. Later Abarbanel and Gottlieb [1] proved that this approach is not strongly well posed and since then several other approaches generalizing the ideas of Berenger have been suggested. A survey of PML layers is to be found in [14]. Turkel and Yefet [30] showed that one can convert between the possibilities with a linear transformation. In Chapter 1 we show how to build such a layer based on the Maxwell’s equations. The solvability of the PML equation for the Helmholtz equation

$$\frac{\partial}{\partial x} \left( \frac{S_y}{S_x} \frac{\partial}{\partial x} u \right) + \frac{\partial}{\partial y} \left( \frac{S_x}{S_y} \frac{\partial}{\partial y} u \right) + k^2 S_x S_y u = 0$$

and the uniqueness of its solution was analyzed by Turkel and Tsynkov in [28]. The functions $S_x$ and $S_y$ are functions of $x$ and $y$ respectively satisfying inside the PML

$$S_x = 1 + \frac{\sigma_x}{ik}, \quad S_y = 1 + \frac{\sigma_y}{ik}$$
where $\sigma_x$ and $\sigma_y$ are the functions inside the PML that cause the solution to decay. We concentrate on a problem in an infinite strip stretching from $x = 0$ to $x = \infty$. In this strip problem $S_y = 1$. In [28] $\sigma_x$ is assumed to be a constant for the convenience of the analysis, but this choice is not suitable for numerical computations. In Chapter 3 we extend this work for a non constant decaying function

$$
\sigma_x(x) = \sigma \left( \frac{x - L_1}{L_2 - L_1} \right)^p
$$

where $L_1$, $L_2$ are the PML boundaries, $\sigma$ a positive constant and $p$ a positive integer. We search for a practical set of parameters in the PML layer based on an analysis of the error in the combined Helmholtz-PML problem. We also analyze the solvability and accuracy of the PML schemes for this problem.

We end that the error governed by the use of the PML (the absolute value of the difference between the exact solution to the Helmholtz equation and the solution of the combined problem) is given by

$$
2e^{-\frac{2\sqrt{1-(L_2-L_1)}}{p^{1/2}}}.
$$

When we use high order schemes for the Helmholtz equation coupled with a suitable $O(h^2)$ approximation inside the PML we wish that the combined scheme preserve the high order accuracy inside the interior. We analyze the solution of the numerical combined problem and prove that these schemes have this important property, if we choose suitable parameters.

The matrix in the resulting linear system, when using the high-order finite difference schemes on the combined PML-Helmholtz equation, has complex entries and is banded and sparse. In order to solve this linear system we use the $LU$ decomposition for banded systems instead of the standard $LU$ decomposition. Because the matrix is large and
sparse it is expensive in both computer time and storage. Hence, it is frequently more efficient to use iterative methods.

There exists many iterative techniques for sparse systems [22], [12]. In chapter 4 we describe a wide variety of algorithms of approximations of Krylov subspaces. We present the algorithms and search for the most efficient one for the combined problem. We show how the parameters of the decaying function in the PML and its width influence the behavior of these algorithms.

The major issue when using these techniques for the Helmholtz equation is that the rate of convergence (which we denote by the number of iterations until convergence) is very slow. To accelerate the convergence we diagonal-scale the matrix and also search for a preconditioning matrix $M^{-1}$ such that instead of solving $Ax = b$ we solve $M^{-1}Ax = M^{-1}b$ (for left preconditioning). An efficient preconditioner for the pure Helmholtz equation which is based on applying one sweep of SSOR to approximate the inverse of the Laplacian operator was found in [3] and [4]. We show that using this preconditioner on the combined problem is not as efficient as in the pure Helmholtz equation. We construct a new preconditioner based on the same idea, but with a special treatment for the behavior in the PML. This preconditioning is based on low-order schemes to the combined problem, but with a small value of $k$, and is approximated with few sweeps of SSOR and damped Jacobi algorithms. Preconditioned Krylov subspace methods for the Helmholtz equation are also popular when using the boundary element and finite element methods.

Other methods that can be carried out are multigrid methods and domain decomposition methods that are beyond the scope of this research. Another approach for solving the Helmholtz equation is based
on the limiting amplitude principle. In this approach one solves the wave equation until a time harmonic steady state is achieved. This steady state is the desired solution for the Helmholtz equation. We will not pursue this approach in this work.
CHAPTER 1

A Perfectly Matched Layer for the Helmholtz Equation

The Helmholtz equation

\[ \Delta u + k^2 u = 0, \]

describes a wide variety of scattering phenomena including electromagnetic waves and acoustics. To solve this equation in an unbounded domain on a computer, one approach is to truncate the unbounded domain and introduce a boundary condition on the artificial outer surface. One artificial boundary condition is Perfectly Matched Layer (PML) which was first introduced by Berenger [7],[8] for the time-dependent Maxwell equations. In this chapter we develop the Helmholtz equation from the Maxwell equations and construct the PML for the Helmholtz equation which we will use in an unbounded domain in the next chapters.

1. Derivation of the Helmholtz Equation

We start with the Maxwell equations in MKS units which governs the propagation in time \( t \) of electromagnetic waves in vacuum:

\[ \frac{\partial \vec{B}}{\partial t} + \nabla \times \vec{E} = 0, \quad \frac{\partial \vec{D}}{\partial t} - \nabla \times \vec{H} = 0 \quad (1.1) \]

\[ \nabla \cdot \vec{D} = 0, \quad \nabla \cdot \vec{B} = 0 \]

where \( \vec{E} \) and \( \vec{H} \) are the electric and magnetic fields respectively, \( \vec{D} \) is the displacement vector, \( \vec{B} \) is the magnetic intensity. We assume here
2. A PERFECTLY MATCHED LAYER FOR THE HELMHOLTZ EQUATION

the constitutive relations in vacuum:

\[ \vec{B} = \mu_0 \vec{H}, \quad \vec{D} = \epsilon_0 \vec{E} \] (1.2)

where \( \mu_0 \) and \( \epsilon_0 \) are the vacuum permittivity and permeability constants respectively.

> From (1.2) we get

\[ \nabla \times \vec{B} = \mu_0 \nabla \times \vec{H}. \]

Substituting into (1.2)

\[ \nabla \times \vec{B} = \mu_0 \frac{\partial \vec{D}}{\partial t} \]

and differentiating in time

\[ \nabla \times \frac{\partial \vec{B}}{\partial t} = \mu_0 \frac{\partial^2 \vec{D}}{\partial t^2}. \]

Using the equation for \( \nabla \times \vec{E} \) in (1.2) we obtain

\[ \nabla \times \nabla \times \vec{E} = -\mu_0 \frac{\partial^2 \vec{D}}{\partial t^2}. \] (1.3)

Using the identity

\[ \nabla \times \nabla \times \vec{E} = \nabla \left( \nabla \cdot \vec{E} \right) - \Delta \vec{E}, \]

the relation \( \vec{D} = \epsilon_0 \vec{E} \) and the relation \( \mu_0 \epsilon_0 = \frac{1}{c^2} \), where \( c \) is the speed of light in vacuum, in (1.3) we get

\[ \Delta \vec{E} - \nabla \left( \nabla \cdot \vec{E} \right) = \frac{1}{c^2} \frac{\partial^2 \vec{E}}{\partial t^2}. \]

Since \( \nabla \cdot \vec{E} = \frac{1}{\epsilon_0} \nabla \cdot \vec{D} = 0 \) we result with the wave equation

\[ \Delta \vec{E} = \frac{1}{c^2} \frac{\partial^2 \vec{E}}{\partial t^2}. \]
In each direction the Cartesian components of the electric field $\vec{E} = (E_1, E_2, E_3)$, satisfy the scalar wave equation

$$\Delta E_j = \frac{1}{c^2} \frac{\partial^2 E_j}{\partial t^2}, \quad 1 \leq j \leq 3. \quad (1.4)$$

We look for solutions in the form

$$e^{i\omega t} A_+ (x, y, z) + e^{-i\omega t} A_- (x, y, z).$$

Inserting into (1.4) we receive the Helmholtz equation for $u = A_{\pm}$

$$\Delta u + k^2 u = 0, \quad (1.5)$$

where

$$k^2 = \frac{\omega^2}{c^2}.$$

The Helmholtz equation has many applications. Besides electromagnetic waves, other applications include acoustics and elasticity. If we Fourier-transform the wave equation (1.4) in time we get the Helmholtz equation.

$$\Delta u + k^2 (x, y, z) u = F(x, y, z).$$

In this work we analyze the homogeneous Helmholtz equation in two dimensions, and concentrate on the case where $k$ is a constant.

2. Derivation of the PML

To solve (1.5) in an unbounded domain on a computer, one approach is to truncate the unbounded domain and introduce a boundary condition on the artificial outer surface. For many years the standard boundary condition was a local absorbing condition that was a generalization of the Sommerfeld radiation condition e.g. [6], but in recent
Figure 1. The PML technique

years a number of models based on the PML (Perfectly Matched Layers) scheme have become popular. These layers minimize the reflections caused by the artificial boundary.

We begin with an analogue of section 1 and introduce a PML. The Maxwell equations (1.1) and (1.2) in the TE (transverse electric) case reduce in two dimensions to:

\[
\begin{align*}
\frac{\epsilon_0}{\epsilon_0} \frac{\partial E_x}{\partial t} &= \frac{\partial H_z}{\partial y} \\
\frac{\epsilon_0}{\epsilon_0} \frac{\partial E_y}{\partial t} &= -\frac{\partial H_z}{\partial x} \\
\frac{\mu_0}{\epsilon_0} \frac{\partial H_z}{\partial t} &= \left( \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial y} \right)
\end{align*}
\]

(1.6)

To normalize the time and space variables we set \( \epsilon_0 = \mu_0 = 1 \). We introduce electric and magnetic conductivities \( \sigma_x = \sigma_x(x), \sigma_y = \sigma_y(y) \) inside the PML as suggested by Berenger [7]. In any one direction of the PML only one of \( \sigma_x \) or \( \sigma_y \) is a positive real function and only in
2. DERIVATION OF THE PML

the corners need both of them be non-zero (see Figure 1). Equations (1.6) become

\[
\begin{align*}
\epsilon_0 \frac{\partial E_x}{\partial t} + \sigma_y E_x &= \frac{\partial (H_{zx} + H_{zy})}{\partial y} \\
\epsilon_0 \frac{\partial E_y}{\partial t} + \sigma_x E_y &= -\frac{\partial (H_{zx} + H_{zy})}{\partial x} \\
\mu_0 \frac{\partial H_{zx}}{\partial t} + \sigma_x H_{zx} &= -\frac{\partial E_y}{\partial x} \\
\mu_0 \frac{\partial H_{zy}}{\partial t} + \sigma_y H_{zy} &= \frac{\partial E_x}{\partial y},
\end{align*}
\]

where \( H_z = H_{zx} + H_{zy} \). Fourier-transforming in time (1.7) we get

\[
\begin{align*}
\tilde{E}_x (ik + \sigma_y) &= \frac{\partial}{\partial y} \left( \tilde{H}_{zx} + \tilde{H}_{zy} \right) \\
\tilde{E}_y (ik + \sigma_x) &= -\frac{\partial}{\partial x} \left( \tilde{H}_{zx} + \tilde{H}_{zy} \right) \\
\tilde{H}_{zx} (ik + \sigma_x) &= -\frac{\partial}{\partial x} \tilde{E}_y \\
\tilde{H}_{zy} (ik + \sigma_y) &= \frac{\partial}{\partial y} \tilde{E}_x.
\end{align*}
\]

Solving for \( \tilde{E}_x \)

\[
\tilde{E}_x = \frac{1}{ik + \sigma_y} \frac{\partial}{\partial y} \left( \tilde{H}_{zx} + \tilde{H}_{zy} \right) = \frac{1}{ik + \sigma_y} \frac{\partial}{\partial y} \left( \frac{1}{ik + \sigma_x} \frac{\partial}{\partial x} \tilde{E}_y + \frac{1}{ik + \sigma_y} \frac{\partial}{\partial y} \tilde{E}_x \right)
\]

and since

\[-\frac{1}{ik + \sigma_y} \frac{\partial}{\partial y} \frac{\partial}{\partial x} \tilde{E}_y = \frac{1}{ik + \sigma_x} \frac{\partial}{\partial x} \left( \frac{1}{ik + \sigma_x} \frac{\partial}{\partial x} \tilde{E}_x \right),\]

we get

\[
\tilde{E}_x = \frac{1}{ik + \sigma_y} \frac{\partial}{\partial y} \left( \frac{1}{ik + \sigma_y} \frac{\partial}{\partial y} \tilde{E}_x \right) + \frac{1}{ik + \sigma_x} \frac{\partial}{\partial x} \left( \frac{1}{ik + \sigma_x} \frac{\partial}{\partial x} \tilde{E}_x \right)
\]

(1.8)
and a similar equation for $\hat{E}_y$ and $\hat{H}_z$. De\(\phi\)ning
\[ S_x = 1 + \frac{\sigma_x}{ik}, \quad S_y = 1 + \frac{\sigma_y}{ik} \quad (1.9) \]
(1.8) becomes
\[ \hat{E}_x = \frac{1}{ik S_y \partial y} \left( \frac{1}{ik S_y \partial y} \hat{E}_x \right) + \frac{1}{ik S_x \partial x} \left( \frac{1}{ik S_x \partial x} \hat{E}_x \right). \]
Multiplying by $-k^2 S_x S_y$ and rearranging we get the same equation in the PML for the function $u$ which is one of the functions $\hat{E}_x$, $\hat{E}_y$, $\hat{H}_z$:
\[ \frac{\partial}{\partial x} \left( \frac{S_y}{S_x} \frac{\partial}{\partial x} u \right) + \frac{\partial}{\partial y} \left( \frac{S_x}{S_y} \frac{\partial}{\partial y} u \right) + k^2 S_x S_y u = 0. \quad (1.10) \]
A similar development of this equation for non TE modes can be carried out. A survey of PML layers is to be found in [14]. Turkel and Yefet [30]. If we choose $\sigma_x = \sigma_y = 0$ we get from (1.9) $S_x = S_y = 1$ and (1.10) reduces to the Helmholtz equation in two dimensions.
CHAPTER 2

High Order Accurate Finite Difference Schemes

In this chapter we analyze finite difference schemes for the inhomogeneous Helmholtz equation

$$\Delta u + k^2 (x, y, z) u = F(x, y, z). \quad (2.1)$$

We develop finite difference approximations on uniform grids, denoting $h$ for the grid-size. We start with the standard $O(h^2)$ approximation and then develop high order accuracy schemes specially designed for the Helmholtz equation, up to sixth order of accuracy.

1. Finite Difference Schemes

In two dimensions the Helmholtz equation (2.1) becomes:

$$u_{xx} + u_{yy} + k(x, y)^2 u = F(x, y) \quad (2.2)$$

Let $\phi_{i,j}$ be the numerical approximation to $u(x_i, y_j)$, where $F_{i,j} = F(x_i, y_j)$ is a known function, see Figure 1. We wish to have symmetric stencil in both directions $x$ and $y$. A scheme having these properties has the form

$$A_0 \phi_{i,j} + A_s \vartheta_s + A_c \vartheta_c = B_0 F_{i,j} + B_s \Gamma_s + B_c \Gamma_c \quad (2.3)$$

where

$$\vartheta_s = \phi_{i,j+1} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i-1,j}$$
Figure 1. The 9 point approximation at each gridpoint \((x_i, y_j)\)

is the sum of the values of the mid-side points and
\[
\vartheta_c = \phi_{i+1,j+1} + \phi_{i+1,j-1} + \phi_{i-1,j-1} + \phi_{i-1,j+1}
\]
is the sum of the values at the corner points. Similarly for \(F\) we have
\[
\Gamma_s = F_{i,j+1} + F_{i+1,j} + F_{i,j-1} + F_{i-1,j}
\]
and
\[
\Gamma_c = F_{i+1,j+1} + F_{i+1,j-1} + F_{i-1,j-1} + F_{i-1,j+1}.
\]

1.1. Pointwise representation. Expanding the standard approximation for the second derivative
\[
D_{xx} \phi = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{h^2}
\]
in a Taylor series, we get for every sufficiently smooth \(u\)
\[
D_{xx} u = u_{xx} + \frac{h^2}{12} u_{xxxx} + \frac{h^4}{360} u_{xxxxxx} + O(h^6).
\]
Adding a similar approximation for \( u_{yy} \) we get

\[
(D_{xx} + D_{yy}) u = u_{xx} + u_{yy} + \frac{h^2}{12} (u_{xxxx} + u_{yyyy}) + \frac{h^4}{360} (u_{xxxxx} + u_{yyyyy}) + O \left( h^6 \right).
\]

Hence, we get the representation

\[
(D_{xx} + D_{yy}) u = u_{xx} + u_{yy} + O \left( h^2 \right)
\]

and for the Helmholtz equation

\[
(D_{xx} + D_{yy}) u = F - k^2 u + O \left( h^2 \right).
\]

Using our notation,

\[
(D_{xx} + D_{yy}) u = \frac{\partial_s - 4\phi_{i,j}}{h^2}.
\]

Multiplying by \( h^2 \) leads to the standard point pointwise representation, (2.3), (see [24, 346])

\[
A_0 = -4 + (kh)^2, \quad A_s = 1, \quad A_c = 0
\]

\[
B_0 = h^2, \quad B_s = B_c = 0.
\]

1.2. Fourth order accurate scheme. Differentiating (2.2) twice with respect to \( x \) and \( y \) we get

\[
u_{xxxy} + u_{xyyy} = F_{xx} - (k^2 u)_{xx}.
\]

\[
u_{yyyy} + u_{xyyy} = F_{yy} - (k^2 u)_{yy}.
\]

Adding these two equations we get

\[
u_{xxxy} + u_{yyyy} = F_{xx} + F_{yy} - (k^2 u)_{xx} + (k^2 u)_{yy} - 2u_{xyyy}.
\]
2. HIGH ORDER ACCURATE FINITE DIFFERENCE SCHEMES

Inserting into (2.4) and neglecting all \(O(h^6)\) terms we conclude

\[
(D_{xx} + D_{yy})u = F - k^2u + \frac{h^2}{12} \left( F_{xx} + F_{yy} - \left( (k^2u)_{xx} + (k^2u)_{yy} \right) - 2u_{xxyy} \right) + O(h^4).
\]

In order to preserve the \(O(h^4)\) approximation, we can use an \(O(h^2)\) approximation for \(F_{xx} + F_{yy} - \left( (k^2u)_{xx} + (k^2u)_{yy} \right) - 2u_{xxyy} \). We choose

\[
F_{xx} + F_{yy} \sim (D_{xx} + D_{yy}) F
\]

and

\[
u_{xxyy} \sim D_{xx}D_{yy}u.
\]

One can choose to directly difference \((k^2u)_{xx} + (k^2u)_{yy}\) with second order accuracy. Instead we assume that \(k\) is constant. Then

\[
(k^2u)_{xx} + (k^2u)_{yy} = k^2(u_{xx} + u_{yy}) = k^2(F - k^2u)
\]

and so (2.9) becomes

\[
\left( D_{xx} + D_{yy} + \frac{h^2}{6} D_{xx}D_{yy} \right) u + \left( 1 - \frac{(kh)^2}{12} \right) k^2u + \frac{h^2}{12} (D_{xx} + D_{yy}) F + O(h^4).
\]

Using (2.5) and

\[
D_{xx}D_{yy}u = \frac{\partial^2 - 2\sigma_s + 4\phi_{i,j}}{h^4}
\]

in (2.10) we get an approximation which is similar to the Equation Based (EB) approximation in [24, 347]

\[
A_0 = \frac{10}{3} + (kh)^2 \left( 1 - \frac{(kh)^2}{12} \right), \quad A_s = \frac{2}{3}, \quad A_c = \frac{1}{6} \quad (2.11)
\]

\[
B_0 = \left( \frac{2}{3} - \frac{k^2}{12} \right) h^2, \quad B_s = \frac{h^2}{12}, \quad B_c = 0.
\]
1.3. Sixth order accurate scheme. To achieve sixth order accurate schemes we have to assume $k$ is a constant, since we need more derivatives of the Helmholtz equation. We also need a fourth order accurate approximation for $D_{xx} + D_{yy}$. Using this assumption and differentiating (2.7) twice with respect to $x$ and $y$ we get

$$u_{xxxxxx} + u_{xxxxyy} = F_{xxxx} - k^2 u_{xxxx}$$  \hspace{1cm} (2.12)

$$u_{yyyyyy} + u_{xxyyyy} = F_{yyyy} - k^2 u_{yyyy}$$

and

$$u_{xxxxyy} + u_{xxyyyy} = F_{xxyy} - k^2 u_{xxyy}.$$  \hspace{1cm} (2.13)

Using (2.8, 2.12, 2.13) we get

$$u_{xxxxxx} + u_{yyyyyy} = F_{xxxx} + F_{yyyy} - F_{xxyy}$$
$$- k^2 \left( F_{xx} + F_{yy} - k^2 (u_{xx} + u_{yy}) - 3u_{xxyy} \right).$$

Inserting into (2.4) we get

$$(D_{xx} + D_{yy}) u = F - k^2 u$$
$$+ \frac{h^2}{12} \left( F_{xx} + F_{yy} - k^2 (u_{xx} + u_{yy}) - 2u_{xxyy} \right)$$
$$+ \frac{h^4}{360} \left( F_{xxxx} + F_{yyyy} - F_{xxyy} - k^2 \left( F_{xx} + F_{yy} - k^2 (u_{xx} + u_{yy}) - 3u_{xxyy} \right) \right) + O(h^6).$$

Using the Taylor series for $D_{xx}D_{yy}$ we end

$$D_{xx}D_{yy}u = u_{xxyy} + \frac{h^2}{12} (u_{xxxxxx} + u_{xxyyyy}) + O(h^4).$$
So we get

\[
\left( D_{xx} + D_{yy} + \frac{h^2}{6} D_{xx}D_{yy} \right) u = \quad (2.14)
\]

\[
F - k^2 u + \frac{h^2}{12} \left( F_{xx} + F_{yy} - k^2 (u_{xx} + u_{yy}) \right) + \frac{h^4}{360} \left( F_{xxxx} + F_{yyyy} + 4F_{xxyy} - k^2 \left( F_{xx} + F_{yy} - k^2 (u_{xx} + u_{yy}) + 2u_{xxyy} \right) \right) + O(h^6).
\]

Using \( D_{xx}D_{yy}u \) as an \( O(h^2) \) approximation for \( u_{xxyy} \) and \( F - k^2 u \) instead of \( u_{xx} + u_{yy} \) and rearranging we get

\[
\left( D_{xx} + D_{yy} + \frac{h^2}{6} \left( 1 + \frac{(kh)^2}{30} \right) D_{xx}D_{yy} \right) u \\
+ \left( 1 - \frac{(kh)^2}{12} \left( 1 - \frac{(kh)^2}{30} \right) \right) k^2 u \\
= F \left( 1 - \frac{(kh)^2}{12} \left( 1 - \frac{(kh)^2}{30} \right) \right) + \frac{h^2}{12} \left( 1 - \frac{(kh)^2}{30} \right) (F_{xx} + F_{yy}) \\
+ \frac{h^4}{360} (F_{xxxx} + 4F_{xxyy} + F_{yyyy}) + O(h^6).
\]

For arbitrary \( F \) we need a fourth order accuracy approximation for \( F_{xx} + F_{yy} \) and a second order accuracy approximation for \( F_{xxxx} + 4F_{xxyy} + F_{yyyy} \). This is easily done, but requires more than nine points and we cannot put the approximation in the form (2.3). However, we can use more than nine points because these approximations only affect the vector \( b \) in the resulting linear problem \( Ax = b \). In many problems \( F = 0 \), and we can write in our notation

\[
A_0 = -\frac{10}{3} + (kh)^2 \left( \frac{46}{45} - \frac{(kh)^2}{12} + \frac{(kh)^4}{360} \right) \quad (2.15)
\]

\[
A_i = \frac{2}{3} \frac{(kh)^2}{90}, \quad A_e = \frac{1}{6} \frac{(kh)^2}{180}.
\]

We call this approximation EB-6.
2. DIVERGENCE FORM

2. Divergence Form

The last two schemes developed have a difficulty. When we want to use the schemes in an unbounded domain, with a PML like the one we developed in Chapter 1, we need to solve, inside the PML, the variable coefficient equation (1.10). In the more general case we need to solve

\[
\frac{\partial}{\partial x} (Au_x) + \frac{\partial}{\partial y} (Bu_y) + Cu = F. \tag{2.16}
\]

where \(A, B, C\) are functions of \(x, y\). We have not found a formula which keeps the self-adjoint form and is also fourth order accurate for non-constant \(A, B\) and \(C\). In the PML layer \(A, B, C\) are variable but in the interior of the domain \(A = B = 1, C = k^2\), so we recover the Helmholtz equation. We want to use in the interior a symmetric stencil in the form (2.3), and automatically switch to a second order accurate stencil in the PML layer. We also want that when using high-order schemes inside the interior and a second order accurate scheme in the PML, then the accuracy of the solution in the interior remains high order. We prove in Chapter 3 that the schemes developed in this section fulfill this property.

2.1. Second order divergence form. We start with the standard second order three point symmetric approximation

\[
D_x (Au_x)_j = \frac{A_{i+\frac{1}{2},j} (u_{i+1,j} - u_{i,j}) - A_{i-\frac{1}{2},j} (u_{i,j} - u_{i-1,j})}{h^2}.
\]

Using the Taylor expansion we get

\[
D_x (Au_x)_j = \frac{\partial}{\partial x} (Au_x) + \frac{h^2}{24} \left( \frac{\partial^3}{\partial x^3} (Au_x) + \frac{\partial}{\partial x} (Au_{xxx}) \right) + O(h^4) \tag{2.17}
\]
We can construct a more general divergence free form by averaging this in the $y$ direction. So we take

$$\alpha D_x (Au_x)_j + \frac{1-\alpha}{2} (D_x (Au_x)_{j+1} + D_x (Au_x)_{j-1})$$

and averaging in the $y$ direction yields

$$\begin{align*}
[Au_x]_x &= \alpha D_x (Au_x)_j + \frac{1-\alpha}{2} \left( D_x (Au_x)_{j+1} + D_x (Au_x)_{j-1} \right) \\
&= \frac{\partial}{\partial x} (Au_x) + \nu h^2 + O \left( h^4 \right),
\end{align*}$$

where

$$\nu = \frac{1}{24} \left( \frac{\partial^3}{\partial x^3} (Au_x) + \frac{\partial}{\partial x} (Au_{xxx}) + 12 (1-\alpha) \frac{\partial^2}{\partial y^2} \frac{\partial}{\partial x} (Au_x) \right).$$

A similar formula for $[Bu_y]_y$ is valid for $D_y (Bu_y)$. The approximation to $Cu$ can be a general nine point formula

$$[Cu] = (1-4\beta_s - 4\beta_c) C_{i,j} u_{i,j} +$$

$$\begin{align*}
\beta_s & \left( C_{i+1,j} u_{i+1,j} + C_{i-1,j} u_{i-1,j} \right) + \\
\beta_c & \left( C_{i+1,j+1} u_{i+1,j+1} + C_{i-1,j-1} u_{i-1,j-1} \right)
\end{align*}$$

Using the Taylor expansion we obtain

$$[Cu] = Cu + h^2 (\beta_s + 2\beta_c) \left( (Cu)_{xx} + (Cu)_{yy} \right) + O \left( h^4 \right).$$

Thus the approximation $[Au_x]_x + [Bu_y]_y + [Cu]$ for the left hand side of (2.16) is guaranteed to be $O \left( h^2 \right)$ for all values of $\alpha, \beta_s, \beta_c$, having

$$[Au_x]_x + [Bu_y]_y + [Cu] = \frac{\partial}{\partial x} (Au_x) + \frac{\partial}{\partial y} (Bu_y) + Cu + O h^2 + O \left( h^4 \right).$$
2. DIVERGENCE FORM

where $\Theta_{h^2}$, the error up to order $h^2$, satisfies

$$
\Theta_{h^2} = \frac{1}{24} \left( \frac{\partial^3}{\partial x^3} (Au_x) + \frac{\partial}{\partial x} (Au_{xxx}) + 12 (1 - \alpha) \frac{\partial^2}{\partial y^2} \frac{\partial}{\partial x} (Au_x) \right) + \\
\frac{1}{24} \left( \frac{\partial^3}{\partial y^3} (Bu_y) + \frac{\partial}{\partial y} (Bu_{yyy}) + 12 (1 - \alpha) \frac{\partial^2}{\partial x^2} \frac{\partial}{\partial y} (Bu_y) \right) + \\
(\beta_s + 2\beta_c) \left( (Cu)_{xx} + (Cu)_{yy} \right)
$$

(2.20)

When $A = B = 1$, $C = k^2$ we get

$$
A_0 = -4\alpha + (1 - 4\beta_s - 4\beta_c) (kh)^2
$$

(2.21)

$$
A_s = 2\alpha - 1 + \beta_s (kh)^2, \quad A_c = 1 - \alpha + \beta_c (kh)^2.
$$

Choosing $\alpha = 1, \beta_s = 0, \beta_c = 0$ recovers the pointwise representation (2.6). We wish to construct higher order approximations that use this limited subset of coefficients. Unfortunately this cannot be done for neither (2.11) nor (2.15).

2.2. Symmetric divergence form. The approximation (2.19) is not symmetric and not self adjoint for general functions $A, B, C$. In order to achieve such approximation we replace $\frac{1}{2} \left( D_x (Au_x)_{j+1} + D_x (Au_x)_{j-1} \right)$ in (2.18) with a symmetric and self-adjoint approximation, getting a new second order accurate scheme which is in divergence form

$$
\left[ Au_x \right]_x
\begin{align*}
&= \alpha D_x (Au_x)_j + \frac{1 - \alpha}{2} \left( \frac{A_{i+1,j+1}u_{i+1,j+1} - A_{i+1,j+1}u_{i,j+1} - A_{i+1,j+1}u_{i+1,j+1} + A_{i+1,j+1}u_{i,j+1}}{h^2} \right) + \\
&= \frac{\partial}{\partial x} (Au_x) + \tilde{\nu} h^2 + O(h^4).
\end{align*}
$$

Now

$$
\tilde{\nu} = \frac{1}{24} \left[ \frac{\partial^3}{\partial x^3} (Au_x) + \frac{\partial}{\partial x} (Au_{xxx}) + 12 (1 - \alpha) \frac{\partial}{\partial x} \left( Au_{xyy} + A_y u_{xy} + \frac{1}{4} A_{yy} u_x \right) \right].
$$

(2.22)
The approximation \( \widetilde{Bu_y} \) is similar to \( \widetilde{Au_x} \), and the approximation to \( Cu \) is now a general symmetric nine point formula

\[
\widetilde{Cu} = (1 - 4\beta_s - 4\beta_c) C_{i,j} u_{i,j} + \\
\beta_s \left( C_{i+1,j} u_{i+1,j} + C_{i-1,j} u_{i-1,j} + C_{i,j+1} u_{i,j+1} + C_{i,j-1} u_{i,j-1} \right) + \\
\beta_c \left( C_{i+1,j+1} u_{i+1,j+1} + C_{i-1,j+1} u_{i-1,j+1} + C_{i+1,j-1} u_{i+1,j-1} + C_{i-1,j-1} u_{i-1,j-1} \right).
\]

Having

\[
\widetilde{Cu} = Cu + h^2 (\beta_s + 2\beta_c) \left( Cu_{xx} + Cu_{yy} + C_x u_x + C_y u_y + \frac{1}{4} (C_{xx} u + C_{yy} u) \right) + O(h^4).
\]  

Again, the approximation \( \widetilde{Au_x} \) for the left hand side of (2.16) is guaranteed to be \( O(h^2) \) for all values of \( \alpha, \beta_s, \beta_c \), i.e.

\[
\widetilde{Au_x} + \widetilde{Bu_y} + \widetilde{Cu} = \frac{\partial}{\partial x} (Au_x) + \frac{\partial}{\partial y} (Bu_y) + Cu + \Theta h^2 + O(h^4).
\]

Similar to the value of \( \Theta h^2 \) in (2.20), the value of \( \Theta h^2 \) is found by combining (2.22) and (2.23). When \( A = B = 1, C = h^2 \) we get the same values of \( A_0, A_1, \) and \( A_c \) as in (2.6). The difference occurs for non-constant values of \( A, B \) or \( C \).

We do not continue to analyze this symmetric form. Numerical results show that when using the symmetric form inside the PML coupled with high-order schemes inside the interior yields an overall accuracy of \( O(h^2) \). However, combining the non-symmetric form (2.19) in the
PML with a high order scheme in the interior, as done in the next subsections, yields high-order accuracy in the physical layer. This is proved in the next chapter. It is not clear why the non-symmetric scheme has better properties than the symmetric scheme with regard to preserving the high order accuracy in the physical domain when using a second order accurate scheme in the PML.

2.3. Fourth order divergence form. To conserve the fourth order accuracy of the scheme achieved in section 1.2, we need an $O(h^2)$ approximation for the term $F_{xx} + F_{yy} - k^2 (u_{xx} + u_{yy}) - 2u_{xxyy}$ in (2.9). Instead of using $(F - k^2 u)$ as the approximation for $(u_{xx} + u_{yy})$ we can use $\left( D_{xx} + D_{yy} + \frac{h^2}{6} D_{xx} D_{yy} \right) u$, yielding

$$
\left( D_{xx} + D_{yy} + \frac{h^2}{6} D_{xx} D_{yy} \right) \left( 1 + \frac{(kh)^2}{12} \right) u + k^2 u
$$

$$
= F + \frac{h^2}{12} (D_{xx} + D_{yy}) F + O(h^4).
$$

Rearranging, we get

$$
\left( D_{xx} + D_{yy} + \frac{h^2}{6} D_{xx} D_{yy} \right) u
$$

$$
+ k^2 \left( 1 + \frac{h^2}{12} (D_{xx} + D_{yy}) + \frac{h^4}{72} D_{xx} D_{yy} \right) u
$$

$$
= F + \frac{h^2}{12} (D_{xx} + D_{yy}) F + O(h^4).
$$

The term $\frac{h^4}{72}$ is fourth order, and can be multiplied by an arbitrary scalar, chosen as $\gamma$. The same treatment can be done on the right hand side of (2.24), choosing $\left( \frac{h^2}{12} (D_{xx} + D_{yy}) + \delta \frac{h^4}{744} D_{xx} D_{yy} \right) F$ (for
arbitrary $\delta$) instead of $\frac{h^2}{12} (D_{xx} + D_{yy}) F$, yielding

\begin{align*}
A_0 &= -\frac{10}{3} + (kh)^2 \left( \frac{2}{3} + \frac{\gamma}{36} \right) \\
A_s &= \frac{2}{3} + (kh)^2 \left( \frac{1}{12} - \frac{\gamma}{36} \right), \quad A_c = \frac{1}{6} + (kh)^2 \frac{\gamma}{144} \\
B_0 &= \left( \frac{2}{3} + \frac{\delta}{36} \right) h^2, \quad B_s = \left( \frac{1}{12} - \frac{\delta}{72} \right) h^2, \quad B_c = \frac{\delta}{144} h^2.
\end{align*}

This approximation is the same as the Pad approximation developed in [29], which is also valid for non-constant $k$. We call this formulation HO.

This scheme is of the form (2.21) with

$$\alpha = \frac{5}{6}, \quad \beta_s = \frac{1}{12} - \frac{\gamma}{72}, \quad \beta_c = \frac{\gamma}{144}$$

Hence, when used for problems with variable coefficients the scheme will be second order accurate. In regions where the coefficients are constant the scheme will increase to fourth order accuracy.

Using these values of $\alpha = \frac{5}{6}$ and $\beta_s + 2\beta_c = \frac{1}{12}$ in (2.20) we get

\begin{align*}
\Theta h^2 &= \frac{1}{12} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \left( (Au)_x + (Bu)_y + Cu \right) + \\
&\quad \frac{1}{24} \left( (Au_{xxx})_x - (Au_{xx})_x + (Bu_{yyy})_y - (Bu_{yy})_y \right) \quad (2.26)
\end{align*}

We get two important results from this equation. The first one is that if $A, B, C$ are constants and $u$ is the exact solution of (2.16) we get $\Theta h^2 = 0$, and so the approximation (2.19) is of order $O(h^4)$. The second result is that if we use this scheme to solve the Helmholtz equation with non-constant value of $k$, but $A = B = 1$, and $u$ is the exact solution we also get $\Theta h^2 = 0$. This result is consistent with the scheme development for non-constant $k$ in [29].
2.4. Sixth order divergence form. Rearranging (2.14) we get

\[
\left( D_{xx} + D_{yy} + \frac{h^2}{6} \left(1 + \frac{(k h)^2}{30}\right) D_{xx} D_{yy}\right) u = -k^2 u - \frac{k^2 h^2}{12} (u_{xx} + u_{yy}) + \frac{k^4 h^4}{360} (u_{xx} + u_{yy}) + F
\]

\[
+ \frac{h^2}{12} \left(1 - \frac{k^2 h^2}{30}\right) (F_{xx} + F_{yy}) + \frac{h^4}{360} (F_{xxx} + 4F_{xxyy} + F_{yyyy}) + O \left( h^6 \right).
\]

To preserve the sixth order accuracy we need an fourth order approximation for the term \( \frac{k^2 h^2}{12} (u_{xx} + u_{yy}) \). Using (2.4.2.8) we get

\[
u_{xx} + u_{yy} = (D_{xx} + D_{yy}) u - \frac{h^2}{12} (F_{xx} + F_{yy} - k^2 (u_{xx} + u_{yy}) - 2u_{xxyy}) + O \left( h^4 \right)
\]

or

\[
u_{xx} + u_{yy} =
\]

\[
(D_{xx} + D_{yy}) u - \frac{h^2}{12} (F_{xx} + F_{yy} - k^2 (D_{xx} + D_{yy}) u - 2D_{xx} D_{yy} u) + O \left( h^4 \right)
\]

\[
= (D_{xx} + D_{yy}) \left(1 + \frac{k^2 h^2}{12}\right) u + \frac{h^2}{6} D_{xx} D_{yy} u - \frac{h^2}{12} (F_{xx} + F_{yy}) + O \left( h^4 \right).
\]

Inserting this fourth order approximation and \( (D_{xx} + D_{yy} - \frac{2}{3} h^2 D_{xx} D_{yy}) u \) as a second order accurate approximation to the term \( \frac{k^4 h^4}{360} (u_{xx} + u_{yy}) \) in (2.27) we get

\[
\left( D_{xx} + D_{yy} + \frac{h^2}{6} \left(1 + \frac{(k h)^2}{30}\right) D_{xx} D_{yy}\right) u =
\]

\[
- k^2 u - \frac{(k h)^2}{12} \left(D_{xx} + D_{yy}\right) \left(1 + \frac{k^2 h^2}{12}\right) u + \frac{h^2}{6} D_{xx} D_{yy} u - \frac{h^2}{12} (F_{xx} + F_{yy})
\]

\[
+ \frac{(k h)^4}{360} \left(D_{xx} + D_{yy} - \frac{2}{3} h^2 D_{xx} D_{yy}\right) u
\]

\[
+ F + \frac{h^2}{12} \left(1 - \frac{k^2 h^2}{30}\right) (F_{xx} + F_{yy}) + \frac{h^4}{360} (F_{xxx} + 4F_{xxyy} + F_{yyyy}) + O \left( h^6 \right).
\]
or

\[
\left(1 + \frac{(kh)^2}{12} + \frac{(kh)^4}{240}\right)(D_{xx} + D_{yy})u + \\
\frac{h^2}{6}\left(1 + \frac{7}{60}(kh)^2 + \frac{\gamma}{120}(kh)^4\right)D_{xx}D_{yy}u + k^2u = \\
F + \frac{h^2}{12}\left(1 + \frac{(kh)^2}{20}\right)(F_{xx} + F_{yy}) + \frac{h^4}{360}(F_{xxxx} + 4F_{xxyy} + F_{yyyy}) + O(h^6)
\]

Assuming \( F = 0 \), we achieve

\[
A_0 = -\frac{10}{3} + \frac{67}{90}(kh)^2 + \frac{\gamma - 3}{180}(kh)^4 \\
A_s = \frac{2}{3} + \frac{2}{45}(kh)^2 + \frac{3 - 2\gamma}{720}(kh)^4 \\
A_c = \frac{1}{6} + \frac{7}{360}(kh)^2 + \frac{\gamma}{720}(kh)^4.
\]

This approximation is in the form (2.21) with

\[
\alpha = \frac{5}{6}, \quad \beta_s = \frac{2}{45} + \frac{3 - 2\gamma}{720}(kh)^2, \quad \beta_c = \frac{7}{360} + \frac{\gamma}{720}(kh)^2
\]

and is sixth-order accurate for all values of \( \gamma \). We call this formulation HO-6. As in the HO scheme we get \( \alpha = \frac{5}{6} \) and \( \beta_s + 2\beta_c = \frac{1}{12} + O(k^2h^2) \), which yields similar results to (2.26).

In [10] a scheme which is also sixth order of accuracy was developed using minimization of the Green function. Choosing \( \gamma = \frac{11}{12} \) and using the notations in [10], we get

\[
a_1 = \frac{A_c}{A_0} = \frac{1440 + 168(kh)^2 + 11(kh)^4}{28800 - 6432(kh)^2 + 100(kh)^4} \\
a_2 = \frac{A_s}{A_0} = \frac{2880 + 192(kh)^2 + 7(kh)^4}{14400 - 3216(kh)^2 + 50(kh)^4}.
\]
Expanding in a Taylor series we get

\[
a_1 = \frac{1}{20} + \frac{17}{1000} (kh)^2 + \frac{801}{200000} (kh)^4 + O(h^6)\\
a_2 = \frac{1}{5} + \frac{29}{500} (kh)^2 + \frac{2549}{200000} (kh)^4 + O(h^6)
\]

which is the same approximation, up to sixth order accuracy, achieved in [10].

2.5. Approximations of the boundary. When we solve the Helmholtz equation in a closed bounded space \(- S \), we wish to approximate the boundary conditions, on the boundary \( \partial S \), with the same accuracy used in the interior. For the Dirichlet boundary condition

\[
u = f(x) \quad \text{for } x \in \partial S
\]

we simply use the known function \( f(x) \) on the boundary and set

\[
\phi_{i,j} = f(x_i, y_j).
\]

This exact value is suitable for all the schemes.

For the Neumann boundary condition

\[
u_n = f(x) \quad \text{for } x \in \partial S, \text{ } n \text{ the normal to } \partial S
\]

the task is more complicated, and is beyond the scope of this work. In [24, 348-350] we developed a suitable approximation for the \( O(h^4) \) HO scheme.

3. Numerical Results

We examine the behavior of the schemes developed on model problems. Both model problems are the Helmholtz equation in a square coupled with a Dirichlet boundary condition. Even though the numerical examples use the model problems, they illustrate the benefits
2. HIGH ORDER ACCURATE FINITE DIFFERENCE SCHEMES

to be expected for more realistic problems. Numerical results of the combined PML-Helmholtz equation are found in the next chapter.

The model problem used for both cases is

\[ \Delta u + k^2 u = F(x, y) \]

in the square \([0, \pi] \times [0, \pi]\) with the boundary conditions

\[ u(0, y) = u(\pi, y) = u(x, 0) = u(x, \pi) = 0. \]

We choose two different functions \(F(x, y)\), which are tailored to have explicit exact solutions. We choose the exact solutions as

\[ u(x, y) = \sin ix \sin jy \] (2.29)

where \(i, j\) are positive integers, chosen as 1 and 2 respectively. In this model problem

\[ F(x, y) = \left( k^2 - (i^2 + j^2) \right) \sin ix \sin jy \]

In the second problem we choose the solution as

\[ u(x, y) = \sinh \alpha x \sinh \alpha (\pi - x) \sinh \beta y \sinh \beta (\pi - y) \]

\[ = \left( \cosh \alpha \pi - \cosh \alpha (2x) \right) \left( \cosh \beta \pi - \cosh \beta (2y) \right) \]

where \(\alpha, \beta\) are real numbers, chosen as 0.5, 0.7 respectively. For this second problem

\[ F(x, y) = k^2 \sinh \alpha x \sinh \alpha (\pi - x) \sinh \beta y \sinh \beta (\pi - y) \]

\[ - \alpha^2 \cosh \alpha (\pi - 2x) \sinh \beta y \sinh \beta (\pi - y) \]

\[ - \beta^2 \cosh \beta (\pi - 2y) \sinh \alpha x \sinh \alpha (\pi - x) \]

The problems are solved using the second order accurate PT scheme (2.6), the fourth order accurate schemes EB (2.11), and HO (2.25) with the parameter \(\gamma = 2\), and the sixth order accurate schemes EB-6 (2.15)
and HO-6 (2.28) with $\gamma = 2$ and $\gamma = -1$. We use a uniform grid size $h = \frac{2}{n}$ where $n$ is the number of gridpoints in both $x, y$ directions.

For the sixth-order schemes we need high order approximations for the function $F(x, y)$ and its derivatives. In order to simplify the schemes we use the exact values for $F_{xx} + F_{yy}$, and $F_{xxxx} + 4F_{xxyy} + F_{yyyy}$. We compare the numerical solution $\phi_{i,j}$ to $u(x_i, y_j)$ where $u$ is the exact solution (2.29) or (2.30). The error vector $e$ is

$$e_{i,j} = u(x_i, y_j) - \phi_{i,j}.$$  

We measure the error in the $l_\infty$ norm:

$$\| e \|_\infty = \max_{0 \leq i,j \leq n} e_{i,j}.$$ 

To solve the linear system in this section, we use a banded $LU$ factorization.

In Table 1 we present the $l_\infty$ norm of the error in the örst problem (2.29) with $k = 6.4$. In all the schemes the norm of the error decreases, as the number of gridpoints increases. We clearly see that the norm of the error behaves proportional to the order of the scheme. As we multiply $n$ by 2 (divide $h$ by 2) the norm of the error decreases by
2. HIGH ORDER ACCURATE FINITE DIFFERENCE SCHEMES

<table>
<thead>
<tr>
<th>$n$</th>
<th>PT</th>
<th>EB</th>
<th>HO $\gamma = 2$</th>
<th>EB-6</th>
<th>HO-6 $\gamma = 2$</th>
<th>HO-6 $\gamma = -1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4.34E-03</td>
<td>2.66E-04</td>
<td>5.03E-03</td>
<td>6.88E-04</td>
<td>5.74E-02</td>
<td>2.21E-01</td>
</tr>
<tr>
<td>8</td>
<td>1.90E-03</td>
<td>9.82E-04</td>
<td>4.14E-03</td>
<td>8.44E-05</td>
<td>2.43E-03</td>
<td>2.96E-03</td>
</tr>
<tr>
<td>16</td>
<td>2.74E-03</td>
<td>1.32E-05</td>
<td>1.21E-04</td>
<td>8.58E-07</td>
<td>5.25E-05</td>
<td>3.48E-05</td>
</tr>
<tr>
<td>32</td>
<td>2.78E-04</td>
<td>8.30E-07</td>
<td>7.77E-06</td>
<td>1.31E-09</td>
<td>9.06E-07</td>
<td>5.61E-08</td>
</tr>
<tr>
<td>64</td>
<td>6.51E-05</td>
<td>5.09E-08</td>
<td>4.87E-07</td>
<td>2.02E-10</td>
<td>1.43E-08</td>
<td>8.97E-09</td>
</tr>
<tr>
<td>128</td>
<td>1.61E-05</td>
<td>3.19E-09</td>
<td>3.04E-08</td>
<td>3.12E-12</td>
<td>2.24E-10</td>
<td>1.40E-10</td>
</tr>
</tbody>
</table>

Table 2. $\|e\|_\infty$ in the second problem with $k = 6.4$

4 in the PT scheme, by 16 in the EB and HO schemes and by 64 in the EB-6 and HO-6 schemes. Similar results are found for the second problem (2.30) in Table 2.

We wish to computationally verify the order of the schemes, so we assume that for small values of $h$,

$$\|e\|_\infty \simeq C\left(\frac{k}{\pi}\right) n^{-r} = \frac{C\left(\frac{k}{\pi}\right)}{h^r}$$

where $r$ is the order of the scheme, i.e.

$$r = \begin{cases} 
2 & \text{for PT scheme} \\
4 & \text{for the schemes EB,HO} \\
6 & \text{for the schemes EB-6,HO-6} 
\end{cases}$$

Hence, if $n = 2^l$

$$-\log_2 (\|e\|_\infty) \simeq l \cdot r - \log_2 C\left(\frac{k}{\pi}\right). \quad (2.31)$$

We calculate $r$, the order of accuracy, by measuring the slope of the curve. We verify (2.31) using Figures 2-6 (in all the HO schemes $\gamma = 2$). We see that for small values of $k$ our assumption (2.31) is correct, but as $k$ grows the schemes begins to act differently and the assumption
3. NUMERICAL RESULTS

Figure 2. $-\log_2 (\|e\|_\infty)$ in EB schemes, ørst problem,
$k = 1.6$

Figure 3. $-\log_2 (\|e\|_\infty)$ in HO schemes, ørst problem,
$k = 1.6$

is correct only for small values of $h$, see also [24]. For larger $h$ the
solution is no longer reasonable as $k$ grows.
Figure 4. $-\log_2 (\|e\|_\infty)$ in HO & EB schemes, second problem, $k = 1.6$

Figure 5. $-\log_2 (\|e\|_\infty)$ in HO & EB schemes, second problem, $k = 6.4$
Figure 6. $-\log_2 (\|e\|_\infty)$ in HO & EB schemes, orst problem, $k = 12.8$
CHAPTER 3

The Infinite Strip Problem

In this chapter we present an infinite strip problem for the Helmholtz equation. To solve this problem we truncate the unbounded space and introduce the PML we developed in chapter 1. We then analyze the error governed by the use of the artificial boundary condition on the exact solution of the Helmholtz equation. This analysis yields heuristics for choosing the decaying function inside the PML. The physical PML-Helmholtz scheme which is developed is then solved by directly discretizing, using high order finite difference schemes which we specially design for the Helmholtz equation in chapter 2. We analyze the effect of the use of these schemes on the solution and we find conditions required for convergence. We also verify the assumption made in [28] that the overall accuracy of the Helmholtz-PML scheme depends only of the order of accuracy inside the physical domain where we solve the pure Helmholtz equation. We conclude the chapter with numerical results to verify our analytical conclusions.

1. Introduction to the Strip Problem

We consider a semi-infinite x-aligned waveguide from $x = 0$ to $x = \infty$ with width $\pi$ in the $y$ direction, see Figure 1. We wish to solve the Helmholtz equation in two dimensions (2.2) in this semi-infinite strip with the boundary conditions:

$$u(x, 0) = u(x, \pi) = 0,$$
Figure 1. The x-aligned semi-infinite waveguide

and

\[ u(0, y) = f(y), \quad 0 \leq y \leq \pi \]
\[ f(0) = f(\pi) = 0, \]

and specify that \( u \) is outgoing at \(+\infty\). Using the sine Fourier series expansion in the \( y \) direction we get

\[ u(x, y) = \sum_{n=1}^{\infty} \hat{u}(x, n) \sin ny, \quad (3.1) \]

where

\[ \hat{u}(x, n) = \frac{2}{\pi} \int_{0}^{\pi} u(x, y) \sin(ny) \, dy. \]

Inserting this expansion into the Helmholtz equation we get an ordinary differential equation for \( \hat{u}(x, n) \):

\[ \frac{d^2}{dx^2} \hat{u}(x, n) + (k^2 - n^2) \hat{u}(x, n) = 0 \quad (3.2) \]

with the initial condition

\[ \hat{u}(0, n) = \frac{2}{\pi} \int_{0}^{\pi} f(y) \sin(ny) \, dy. \quad (3.3) \]
To guarantee uniqueness of the solution of this equation in the waveguide, we require that only right-traveling waves be present in the solution at infinity.

In the case $n < k$ (traveling wave), the unique solution of (3.2), (3.3) is

$$
\hat{u} (x, n) = \frac{2}{\pi} \left( \int_0^\pi f(y) \sin (ny) \, dy \right) e^{-i \sqrt{k^2 - \omega^2} x}
$$

and so

$$
u (x, y) = \sum_{n=1}^{\infty} \frac{2}{\pi} \left( \int_0^\pi f(y) \sin (ny) \, dy \right) e^{-i \sqrt{k^2 - \omega^2} x} \sin ny.
$$

To simplify the analysis of the PML layer we assume the boundary condition at the entrance to the waveguide is

$$
f(y) = \sin (my), \ m \in \mathbb{Z}
$$

and then

$$
\hat{u} (x, n) = \begin{cases} 
    e^{-i \sqrt{k^2 - \omega^2} x} & n = m \\
    0 & n \neq m
\end{cases}
$$

$$
u (x, y) = e^{-i \sqrt{k^2 - \omega^2} x} \sin my.
$$

We will label this, the pure Helmholtz solution, $u_{exact}$:

$$
u_{exact} (x, y) = e^{-i \sqrt{k^2 - \omega^2} x} \sin my.
$$

In the case $m > k$ (evanescent wave) we get

$$
u_{exact} (x, y) = e^{-\sqrt{m^2 - k^2} x} \sin my
$$
Figure 2. The PML layer in the strip

2. Constructing the PML

Since we cannot solve the semi-infinite problem on a computer we instead solve in a bounded domain: \([0, L_1] \times [0, \pi]\), and construct a PML layer. We first analyze an infinite PML and then a PML with width \(L_2 - L_1\), see Figure 2. In the interior of the strip (vacuum) we solve the Helmholtz equation and in the PML layer we solve equation (1.10). In this case we choose \(\sigma_y = 0\) in the PML layer and so \(S_y = 1\) and hence (1.10) becomes

\[
\frac{\partial}{\partial x} \left( \frac{1}{S_x} u_x \right) + \frac{\partial}{\partial y} (S_x u_y) + k^2 S_x u = 0, \quad S_x = S_x(x). \tag{3.6}
\]

Using the Fourier series expansion (3.1) in \(y\) we get the ordinary differential equation:

\[
\frac{d}{dx} \left( \frac{1}{S_x} \frac{d \hat{u}(x, n)}{dx} \right) + S_x (k^2 - n^2) \hat{u}(x, n) = 0. \tag{3.7}
\]

If we set

\[
S_x(x) = \begin{cases} 
1 & 0 \leq x \leq L_1 \\
1 + \frac{2k}{n} & x > L_1 
\end{cases} \tag{3.8}
\]
equation (3.7) represents \( \hat{u} \) in the entire space (vacuum + PML). Substituting

\[
t = \int_0^x S_x(r) \, dr
\]

and labeling

\[
Y_n(t) = \hat{u}(x, n),
\]

we get by the chain rule

\[
\frac{d\hat{u}(x, n)}{dx} = \frac{dY_n}{dt} \frac{dt}{dx} = S_x \frac{d}{dt} Y_n(t)
\]

and

\[
\frac{1}{S_x} \frac{d}{dx} \left( \frac{1}{S_x} \frac{d\hat{u}(x, n)}{dx} \right) = \frac{d^2}{dt^2} Y_n(t).
\]

And so \( Y_n(t) \) satisfies an ordinary differential equation with constant coefficients

\[
\frac{d^2}{dt^2} Y_n(t) + \left(k^2 - n^2\right) Y_n(t) = 0.
\]

The solution of this equation is

\[
Y_n(t) = c_+ e^{i \sqrt{k^2 - n^2} t} + c_- e^{-i \sqrt{k^2 - n^2} t}.
\]

Thus, the solution for (3.7) is

\[
\hat{u}(x, n) = c_+ e^{i \sqrt{k^2 - n^2} \int_0^x S_x(r) \, dr} + c_- e^{-i \sqrt{k^2 - n^2} \int_0^x S_x(r) \, dr}
\]  

(3.9)

where \( c_+ \) and \( c_- \) are arbitrary constants. The unique solution for (3.7) in an infinite PML is found when we add the condition that only the right-traveling waves are present in addition to the condition at \( x = 0 \).
Using (3.4) we get the unique solution for (3.6) which we label $u_{I-PML}$ (Infinite-PML):

$$
  u_{I-PML}(x,y) = \begin{cases} 
  e^{-i\sqrt{k^2-m^2} \int_0^x S_x(r) dr} \sin my & \text{if } m < k \\
  e^{-i\sqrt{\varpi^2-k^2} \int_0^x S_x(r) dr} \sin my & \text{if } m > k 
\end{cases}.
$$

(3.10)

Comparing (3.5) with (3.10) we get for $0 \leq x \leq L_1$:

$$
  u_{I-PML}(x,y) = u_{\text{exact}}(x,y)
$$

which shows that the PML is perfectly non-reflecting.

When solving the problem on a computer we need to truncate the semi-infinite domain at $L_2$ and we choose the boundary condition as $u = 0$ at $x = L_2$. Choosing other types of boundary conditions at $x = L_2$ does not significantly change the solution. Hence, instead of $c_+ = 0$ in (3.9) we need to solve (3.7) with the boundary condition $\hat{u}(L_2,n) = 0$. For our choice of $f(y) = \sin(my)$, we get for $n \neq m$ in (3.1) $\hat{u}(x,n) = 0$ and

$$
  \hat{u}(x,m) = c_+ e^{i\sqrt{k^2-m^2} \int_0^x S_x(r) dr} + c_- e^{-i\sqrt{k^2-m^2} \int_0^x S_x(r) dr}
$$

with the boundary conditions

$$
  \hat{u}(0,m) = 1
$$

$$
  \hat{u}(L_2,m) = 0.
$$

Solving for $c_+$ and $c_-$ we get

$$
  c_+ = \frac{-1}{\eta - 1}
$$

(3.11)

$$
  c_- = \frac{\eta}{\eta - 1}
$$
where

\[ \eta = e^{2i\sqrt{k^2 - m^2} \int_0^{L_2} S_x(r) \, dr}. \tag{3.12} \]

Uniqueness is not guaranteed when \( \eta = 1 \). Otherwise

\[ 2i\sqrt{k^2 - m^2} \int_0^{L_2} S_x(r) \, dr = 2i\sqrt{k^2 - m^2} \left( L_2 + \frac{1}{ik} \int_{L_1}^{L_2} \sigma_x(r) \, dr \right) \]

since \( \sigma_x(x) > 0 \) for \( x \in (L_1, L_2] \), then for \( k \neq m \),

\[ \text{Re} \left( 2i\sqrt{k^2 - m^2} \int_0^{L_2} S_x(r) \, dr \right) \neq 0 \text{ and } \eta \neq 1. \]

From now on we assume that \( k \neq m \).

Labeling this unique solution \( u_{F-pml} \) (Finite PML) we get

\[ u_{F-pml} = \left( \frac{-1}{\eta - 1} e^{i\sqrt{k^2 - m^2} \int_0^{x} S_x(r) dr} + \frac{\eta}{\eta - 1} e^{-i\sqrt{k^2 - m^2} \int_0^{x} S_x(r) dr} \right) \sin(my) \tag{3.13} \]

and for the case \( m < k \)

\[
\text{error}(x, y) = u_{I-pml} - u_{F-pml} = \frac{1}{\eta - 1} \left( e^{i\sqrt{k^2 - m^2} \int_0^{x} S_x(r) dr} - e^{-i\sqrt{k^2 - m^2} \int_0^{x} S_x(r) dr} \right) \sin(my) = \frac{2i}{\eta - 1} \sin \left( \sqrt{k^2 - m^2} \int_0^{x} S_x(r) \, dr \right) \sin(my). \]

We are interested in the solution in the interior and so we examine the error where \( 0 \leq x \leq L_1 \)

\[ \| \text{error}(x, y) \|_\infty = \max_{0 \leq x \leq L_1, \, 0 \leq y \leq \pi} \left| \frac{2i}{\eta - 1} \sin \left( \sqrt{k^2 - m^2} x \right) \sin (my) \right| \]

\[ = \frac{2}{|\eta - 1|}. \tag{3.14} \]

Our goal is to minimize the error and thus choose a suitable function \( \sigma_x(x) \) that will minimize the value of \( \left| \frac{2}{\eta - 1} \right| \). We wish \( \eta \) in (3.12) to
satisfy $|\eta| >> 1$. For the case $m > k$ we get

$$error(x, y) = u_{l-pml} - u_{F-pml}$$

$$= \frac{\eta}{\eta - 1} \left( e^{\sqrt{k^2 - m^2} \int_0^r S_x(r)dr} - e^{-\sqrt{k^2 - m^2} \int_0^r S_x(r)dr} \right) \sin (my)$$

$$= \frac{2\eta}{\eta - 1} \sinh \left( \sqrt{k^2 - m^2} \int_0^r S_x(r)dr \right) \cdot \sin (my)$$

and inside the interior

$$\|error(x, y)\|_\infty \simeq \left| \frac{\eta}{\eta - 1} e^{\sqrt{m^2 - k^2} L_1} \right| . \tag{3.15}$$

In this case we wish in (3.12) that $|\eta| << 1$.

3. Minimizing the Error

The function $\sigma_x : [L_1, L_2] \rightarrow \mathbb{R}$ is chosen in such a way so that it has the following properties:

$$\sigma_x(x) > 0 \quad \text{for } x \in (L_1, L_2]$$

$$\sigma_x(L_1) = 0$$

$$\sigma_x(x) \text{ is smooth in } [L_1, L_2].$$

A suitable choice for this function is

$$\sigma_x(x) = \sigma \left( \frac{x - L_1}{L_2 - L_1} \right)^p \tag{3.16}$$

where $\sigma$ is a positive constant and $p \geq 1$. Integrating we get

$$\int_{L_1}^x \sigma_x(r) dr = \frac{\sigma (L_2 - L_1)}{p + 1} \left( \frac{x - L_1}{L_2 - L_1} \right)^{p+1}$$
and

\[ u_{F_{pml}} = \sin my \cdot \]

\[
\begin{cases} 
  \frac{-1}{\eta} e^{i \sqrt{k^2 - m^2}x} + \frac{\eta}{\eta} e^{-i \sqrt{k^2 - m^2}x} & 0 < x \leq L_1 \\
  \frac{-1}{\eta} e^{i \sqrt{k^2 - m^2}L_1 + \sqrt{1 - \varepsilon} \frac{\sigma(L_2 - L_1)}{L_2 - L_1} (\frac{x}{L_2})^{p+1}} + \frac{\eta}{\eta} e^{-i \sqrt{k^2 - m^2}L_1 - \sqrt{1 - \varepsilon} \frac{\sigma(L_2 - L_1)}{L_2 - L_1} (\frac{x}{L_2})^{p+1}} & L_1 < x \leq L_2 
\end{cases}
\]

where \( \varepsilon = \left( \frac{m}{k} \right)^2 \) and

\[
\eta = e^{2i \sqrt{k^2 - m^2} \int_0^{L_2} s_x(r) \, dr} = e^{2i \sqrt{k^2 - m^2} L_2 + \frac{2\sqrt{1 - \varepsilon} \sigma(L_2 - L_1)}{p+1}}. \tag{3.18}
\]

For \( m < k \) we get

\[
|\eta| = e^{\frac{2\sqrt{1 - \varepsilon} \sigma(L_2 - L_1)}{p+1}} >> 1
\]

and for \( m > k \)

\[
|\eta| = e^{-2\sqrt{m^2 - k^2} L_2} << 1,
\]

which are the desired results for (3.14) and (3.15). Using estimate (3.14) we get for \( m < k \)

\[
\|error(x, y)\|_\infty \approx \frac{2}{|\eta| - 1} \sim \frac{2}{|\eta|} = 2e^{-\frac{2\sqrt{1 - \varepsilon} \sigma(L_2 - L_1)}{p+1}} \tag{3.19}
\]

and for \( m > k \) using (3.15) we get

\[
\|error(x, y)\|_\infty \approx \frac{\eta e^{\sqrt{m^2 - k^2} L_1}}{|\eta| - 1} \sim |\eta e^{\sqrt{m^2 - k^2} L_1}| = e^{-\sqrt{m^2 - k^2} (2L_2 - L_1)}.
\]

We see that if \( m > k \) then the norm of the error does not depend on \( \sigma_x \). To decrease the error we can only increase the length of the PML region, i.e. \( 2L_2 - L_1 \). The harder problem arises where \( m < k \) and then the error depends exponentially on the value of \( \frac{2\sigma(L_2 - L_1)}{p+1} \). From this continuous analysis we conclude that the parameters should be chosen
to satisfy: A large value of $\sigma$, a wide PML (extend $L_2 - L_1$) and $p = 1$. Further analysis (below) for the numerical algorithm demonstrates that one should be more careful when choosing these parameters.

We need a scheme to approximate the $u_{F-PML}$ solution. For efficiency we use high-order finite differences schemes in the interior. To reduce the size of the matrices we wish to minimize the number of points in the artificial PML layer. In the next sections we present these schemes and analyze their influence on the error. We will also see that the choice a large value of $\sigma$ and a small value of $p$ is inaccurate.

4. Exploring the Numerical Error

We start with (2.19) taking $A = \frac{1}{S_x}$, $B = S_x$, $C = k^2 S_x$ and get

$$
[ Au_x ]_x + [Bu_y]_y + [Cu]
= \frac{\partial}{\partial x} \left( \frac{u_x}{S_x} \right) + \frac{\partial}{\partial y} (S_x u_y) + k^2 S_x u + \Theta_h h^2 + O(h^4). \tag{3.20}
$$

The value $\Theta_{h^2}$ in (2.20) satisfies

$$
\Theta_{h^2} = \frac{1}{24} \left( \frac{\partial^3}{\partial x^3} \left( \frac{u_x}{S_x} \right) + \frac{\partial}{\partial x} \left( \frac{u_{xxx}}{S_x} \right) + 2 S_x u_{yy} \right) + \frac{1-\alpha}{2} \left( \frac{\partial^3}{\partial y^3} \frac{u_y}{S_x} \right) + \frac{\partial^2}{\partial y^2} (S_x u_{yy}) \right) + k^2 (\beta_s + 2\beta_c) \left( (S_x u)_{xx} + S_x u_{yy} \right). \tag{3.21}
$$

In order for the scheme to be accurate we require that $h^2 \Theta_{h^2} \ll 1$. 

Using \( u = u_{F-pml} \) in (3.21) we get

\[
\Theta_h^2 = c_\pm e^{\pm \sqrt{k^2 - m^2} f_0} s_x(r) dr
\]

(3.22)

\[
\frac{1}{24} \left( \begin{array}{c}
\left( 2 (k^2 - m^2)^2 S_x^3 + 8i (k^2 - m^2)^\frac{3}{2} S_x S'_x \right) \\
-5 (k^2 - m^2) S''_x \pm i \sqrt{k^2 - m^2} \left( \frac{S''_x}{S'_x} \right) + 2m^4 S_x
\end{array} \right) +
\frac{1-a}{2} m^2 \left( \begin{array}{c}
(k^2 - m^2) (S_x + S_x^3) \\
+3i \sqrt{k^2 - m^2} S_x S'_x - S''_x
\end{array} \right) +
k^2 (\beta_s + 2\beta_c) \left( \begin{array}{c}
- (k^2 - m^2) S_x^3 \\
\pm 3i \sqrt{k^2 - m^2} S_x S'_x + S''_x - m^2 S
\end{array} \right)
\]

Assuming \( m \ll k \) and \( k \gg 1 \) and collecting the \( O(k^4) \) terms we get

\[
\Theta_h^2 \simeq c_\pm (k^2 - m^2)^2 \left( \frac{1}{12} - \mu \right) S_x^3 e^{\pm \sqrt{k^2 - m^2} f_0} s_x(r) dr
\]

(3.23)

where

\[
\mu = \frac{(\beta_s + 2\beta_c)}{1 - \varepsilon}, \quad \varepsilon = \left( \frac{m}{k} \right)^2.
\]

Using the choice of \( \sigma_x \) (3.16), the exact values of the constants \( c_\pm \) in the solution \( u_{F-pml} \) (3.17) and denoting

\[
z = \frac{x - L_1}{L_2 - L_1}, \quad 0 \leq z \leq 1,
\]

(3.24)

we get inside the PML

\[
|\Theta_h^2| \simeq \frac{(k^2 - m^2)^2 \left( \frac{1}{12} - \mu \right)}{1 - \eta} 
\cdot e^{i \sqrt{k^2 - m^2} L_1 + \sqrt{1 - \varepsilon} \frac{(L_2 - L_1)}{p+1} z^{p+1}} + \eta e^{-i \sqrt{k^2 - m^2} L_1 - \sqrt{1 - \varepsilon} \frac{(L_2 - L_1)}{p+1} z^{p+1}} 
\cdot \left| 1 + \frac{\sigma}{i k} z^{p} \right|^3.
\]
Figure 3. Value of $|C_{h^2}|$ for $k = 8$, $m = 1$, $\sigma = 50$, $L_2$—

$$L_1 = \frac{\pi}{4}$$

Inserting the definition of $\eta$ in (3.18) and rearranging we get

$$|\Theta_{h^2}| \sim \frac{(k^2 - m^2)^2 \left( \frac{1}{12} - \mu \right)}{|1 - \eta|} \cdot$$

$$\left| -e^{\sqrt{1 - \frac{2\sigma(l_{z+1} - l_1)}{p+1}}} + e^{\sqrt{1 - \frac{2\sigma(l_{z+1} - l_1)}{p+1}}} (2 - z^{p+1}) \right| \left| 1 + \frac{\sigma}{ik} z^p \right|^3$$

$$= \frac{(k^2 - m^2)^2 \left( \frac{1}{12} - \mu \right) e^{\sqrt{1 - \frac{2\sigma(l_{z+1} - l_1)}{p+1}}} (1 - z^{p+1})}{|1 - \eta|} \cdot$$

$$\left| e^{\sqrt{1 - \frac{2\sigma(l_{z+1} - l_1)}{p+1}}} (1 - z^{p+1}) - 1 \right| \left| 1 + \frac{\sigma}{ik} z^p \right|^3.$$  

Using the approximations $\left| e^{\sqrt{1 - \frac{2\sigma(l_{z+1} - l_1)}{p+1}}} (1 - z^{p+1}) - 1 \right| \simeq e^{\sqrt{1 - \frac{2\sigma(l_{z+1} - l_1)}{p+1}}} (1 - z^{p+1})$ and $|1 - \eta| \simeq e^{\sqrt{1 - \frac{2\sigma(l_{z+1} - l_1)}{p+1}}}$ we get

$$|\Theta_{h^2}| \simeq (k^2 - m^2)^2 \left( \frac{1}{12} - \mu \right) e^{-\sqrt{1 - \frac{2\sigma(l_{z+1} - l_1)}{p+1}}} \left( 1 + \left( \frac{\sigma}{k} \right) z^{2p} \right)^{\frac{3}{2}}.$$  

(3.25)

We can see in figures 3-7 the behavior of $|\Theta_{h^2}|$ for various values of $p$. In all these figures the x-axis of the graph is the variable $z$ denoted in (3.24). The value of $\beta_s + 2\beta_c$ is set equal to 0 which is what we use with the standard pointwise representation (2.6). One concludes
Figure 4. Value of $|C_{h^2}|$ for $k = 8$, $m = 1$, $\sigma = 150$, $L_2 - L_1 = \frac{\pi}{4}$

Figure 5. Value of $|C_{h^2}|$ for $k = 8$, $m = 1$, $\sigma = 50$, $L_2 - L_1 = \frac{\pi}{8}$

that in most cases $p$ should be set equal to 2 or 3. Increasing the value of $\sigma$ increase also the value of $|\Theta_{h^2}|$. These two facts contradict our desire to decrease the error (3.19). Increasing $L_2 - L_1$ decreases the value of $|\Theta_{h^2}|$ as well as the error, however, this has its ëEaw. A thick PML means more storage and CPU and also means a harder task for the linear solvers. Another diëiculty arises if we use iterative
Figure 6. Value of $|C_{h^2}|$ for $k = 12$, $m = 1$, $\sigma = 50$, $L_2 - L_1 = \frac{\pi}{4}$

Figure 7. Value of $|C_{h^2}|$ for $k = 8$, $m = 3$, $\sigma = 50$, $L_2 - L_1 = \frac{\pi}{4}$

Techniques to solve the linear system. One popular possibility is to have an algorithm which is based on a preconditioned Krylov subspace. We analyze algorithms of this kind in chapter 4. These algorithms are built especially for the combined Helmholtz-PML problem and prove to work better when we use a thin PML layer.
4. EXPLORING THE NUMERICAL ERROR

To get (3.23) we take only the $O(k^4)$ term and neglect the others. We have to prove that

$$\left| \frac{S_x S'_x}{S_x^3} \right| = o(k),$$

$$\left| \frac{S_x S'_x}{S_x^3} \right| = \left| \frac{S'_x}{S_x^2} \right| = \frac{\sigma p}{k (L_2 - L_1)} \cdot \frac{z^{p-1}}{1 + \left( \frac{\alpha}{k} \right)^2 z^{2p}}$$

This function of $z$ attains its maxima when

$$z_{\text{max}} = 2^{p} \left( \frac{k}{\sigma} \right)^{2 \frac{p-1}{p+1}}.$$

For $p = 1$

$$\left| \frac{S_x S'_x}{S_x^3} \right| \leq \frac{\sigma}{k} \frac{1}{L_2 - L_1}$$

and for $p > 1$

$$\left| \frac{S_x S'_x}{S_x^3} \right| \leq \sqrt{\frac{\sigma}{k}} \sqrt{(p-1)^{p-1}(p+1)^{p+1}} \cdot \frac{1}{2(L_2 - L_1)}.$$

Because $\sigma$ is taken larger than $k$ the last inequality indicates that as $p$ increases estimate (3.23) becomes more accurate. However, for small values of $p$, as long as $\frac{\sigma}{k}$ is not too large, this estimate yields accurate approximations. The same analysis for the $O(k^2)$, $O(k)$ and $O(1)$ terms in (3.22) yields similar results.

An interesting result is found when examining the value of $|\Theta_{h^2}|$ near the intersection between the vacuum and the PML layer, i.e. $z \to 0^+$. At this point we can end a lower bound of $|\Theta_{h^2}|$ and get

$$|\Theta_{h^2}| \simeq \left( k^2 - m^2 \right)^2 \left( \frac{1}{12} - \mu \right).$$

(3.26)

Combining this result with Figures 6 and 7 we conclude that as $(k^2 - m^2)^2$ increases, the schemes become more inaccurate independent of the values of the other parameters, $\sigma$, $p$ and $L_2 - L_1$. A feasible way to work on these problems is to rene the grid size choosing smaller values of $h$. 
Figure 8. Value of $|\Theta_{h^2}|$ in the high-order schemes for 
\[ k = 8, \ m = 1, \ \sigma = 50, \ L_2 - L_1 = \frac{\pi}{4} \]

We can also decrease the value of $|\Theta_{h^2}|$ by choosing $\mu \to \frac{1}{12}$. In fact when we examine the value of $\mu$ in the high-order schemes (2.25, 2.28) we get
\[
\frac{1}{12} - \mu = \frac{1}{12} \frac{m^2}{k^2 - m^2}.
\]

Hence, assumption (3.23) is not valid and we have to take the next term in (3.22), of order $O(k^3)$:
\[
\Theta_{h^2} \simeq -i c_{\pm} \left( k^2 - m^2 \right)^{\frac{1}{2}} \left( \frac{1}{3} - 3\mu \right) S_{x,r} S_{z} e^{\pm \sqrt{k^2 - m^2} \int_{0}^{r} S_{z}(r) dr},
\]
when $\frac{1}{3} - 3\mu = \frac{k^2 - 4m^2}{k^2 - m^2}$. Using the same method that was applied to get (3.25) yields
\[
|\Theta_{h^2}| \simeq \frac{\sigma p}{12 (L_2 - L_1)} \sqrt{1 - \varepsilon} \left( k^2 - 4m^2 \right) e^{-\sqrt{1 - \varepsilon} \frac{L_2 - L_1}{p+1}} z^{p+1} z^{-1} \sqrt{1 + \left( \frac{\sigma}{k} \right)^{2} z^{2p}}.
\]

(3.27)

In Figures 8-11 we obtain similar results to the ones we found in the case of the second order schemes. We see that the dependence on the value of $(k^2 - m^2)$ is reduced, which means better performance for large values of $k$ (compare Figures 6 and 11). Another quality, which we can easily
Figure 9. Value of $|C_h^2|$ in the high-order schemes for
$k = 8, \ m = 1, \ \sigma = 150, \ L_2 - L_1 = \frac{\pi}{4}$

Figure 10. Value of $|C_h^2|$ in the high-order schemes for
$k = 8, \ m = 1, \ \sigma = 50, \ L_2 - L_1 = \frac{\pi}{8}$

see from these figures and also from (3.27) is that $p = 1$ is a very poor choice, because it achieves its maximum when $z \to 0^+$, unlike the case when $p > 1$. 
Figure 11. Value of $|C_h^2|$ in the high-order schemes for 
$k = 12$, $m = 1$, $\sigma = 50$, $L_2 - L_1 = \frac{\pi}{4}$

5. The Modified Equation

Another approach to deal with the numerical solution is to analyze the modified equation when using one of the finite difference schemes (2.6, 2.25 or 2.28). We start with (3.20) and (3.21), and try to find the analytical solution for the equation

$$
\frac{\partial}{\partial x} \left( \frac{u_x}{S_x} \right) + \frac{\partial}{\partial y} (S_x u_y) + k^2 S_x u +
$$

$$
h^2 \left( \frac{1}{24} \left( \frac{\partial^2}{\partial x^2} \left( \frac{u_x}{S_x} \right) + \frac{\partial}{\partial x} \left( \frac{u_{xx}}{S_x} \right) + 2S_x u_{yyy} \right) + \frac{1-\nu}{2} \left( \frac{\partial^2}{\partial y^2} \frac{u_x}{S_x} + \frac{\partial^2}{\partial y^2} (S_x u_y) \right) + k^2 (\beta_s + 2\beta_e) ((S_x u)_{xx} + S_x u_{yy}) \right) = 0
$$
When the input boundary condition is \( \sin(my) \) then the only Fourier coefficient that does not vanish is \( U = \hat{u}(x,m) \). We recover the modified ordinary differential equation

\[
\left( \frac{U'}{S_x} \right)' + \left( k^2 - m^2 \right) S_x U + \frac{1}{2k} \left( \left( \frac{U'}{S_x} \right)' + 2m^4 S_x U \right) - \frac{1-n}{2} m^2 \left( \left( \frac{U'}{S_x} \right)' + (S_x U)'' \right) + k^2 (\beta_s + 2\beta_c) \left( (S_x U)^{\#} - m^2 S_x U \right) = 0,
\]

with the boundary conditions

\[ U(0) = 1, \ U(L_2) = 0. \]

Because our main interest is in the high-order schemes we take

\[ \alpha = \frac{5}{6}, \ \beta_s + 2\beta_c = \frac{1}{12} \]

(for the \( O(h^6) \) we take \( \beta_s + 2\beta_c = \frac{1}{12} + O(k^2 h^2) \)). So (3.28) becomes

\[
\left( \frac{U'}{S_x} \right)' + \left( k^2 - m^2 \right) S_x U
\]

\[ = -\frac{k^2}{24} \left( 2m^2 \left( \frac{U'}{S_x} \right)' + 2 \left( k^2 - m^2 \right) \left( (S_x U)' - m^2 S_x U \right) \right). \]

If \( p = 1 \) then \( S_x \) is not differentiable near \( x = L_1 \). Hence, we assume that \( p \geq 2 \). Inside the interior we set \( S_x = 1 \) and thus the right hand side of the equation becomes

\[
-\frac{k^2}{12} \left( U^{(4)} - m^2 U'' + \left( k^2 - m^2 \right) \left( U'' - m^2 U \right) \right) \]

\[ = -\frac{k^2}{12} \left( \frac{d^2}{dx^2} \left( U'' + \left( k^2 - m^2 \right) U \right) - m^2 \left( U'' + \left( k^2 - m^2 \right) U \right) \right) \]
and we can rewrite (3.29)

\[
\left(1 + \frac{h^2}{12 \left( \frac{d^2}{dx^2} - m^2 \right)} \right) \left( U'' + \left( k^2 - m^2 \right) U \right) = 0
\]

which leads to

\[
U'' + \left( k^2 - m^2 \right) U = 0.
\]

This is in fact the Fourier expansion of the Helmholtz equation. Thus, in the interior, we solve the Helmholtz equation up to terms of order \( h^2 \). This is consistent with the results obtained for the pure Helmholtz equation in (2.26). Unlike (3.7) we have not found the exact solution to (3.29). When \( h^2 \ll 1 \) we look for a perturbed solution in the form

\[
U = e^{\pm i \sqrt{\kappa^2 - m^2} \int_0^r S_x(r) (1 + h^2 g_\pm(r)) dr}.
\]

Inserting this into (3.29) and collecting all the terms with \( O(h^2) \) yields a differential equation for \( q_\pm \)

\[
\pm i \sqrt{\kappa^2 - m^2} q_\pm' - 2 \left( k^2 - m^2 \right) q_\pm S_x
= \frac{k^2 - m^2}{24} \left[ \pm 2 i \sqrt{\kappa^2 - m^2} S_x S_x' + 3 S_x'' + \frac{1}{\pm i \sqrt{\kappa^2 - m^2}} \left( \frac{S_x''}{S_x} \right)' \right].
\]

Inside the interior the right hand side of this equation vanishes. This emphasizes that only the \( O(h^2) \) factor comes from the PML approximation. Labeling \( g_\pm (x) = \pm i \sqrt{\kappa^2 - m^2} \int_0^x S_x (r) dr \) and multiplying the equation with the integrating factor \( e^{2g_\pm(x)} \) we get

\[
(q_\pm e^{2g_\pm})' = -\frac{1}{24} \left( 2 g_\pm'' q_\pm' + 3 g_\pm'' + \left( \frac{g_\pm''}{q_\pm} \right)' \right) e^{2g_\pm}
\]

Using integration by parts we get

\[
q_\pm = \frac{-1}{24} \left( \pm i \sqrt{\kappa^2 - m^2} S_x' + \frac{S_x''}{S_x} \right) + d_\pm e^{\pm 2 i \sqrt{\kappa^2 - m^2} \int_0^r S_x(r) dr}
\]
for some constants $d_{\pm}$. Inserting into (3.30) we obtain an estimate for the numerical solution using the high-order schemes

$$
\hat{u} (x, m) = c_{+} e^{i \sqrt{k^2 - m^2} J_0} S_\pi (r) dr - \frac{k^2}{48} \Psi_+ (x) + c_{-} e^{-i \sqrt{k^2 - m^2} J_0} S_\pi (r) dr - \frac{k^2}{48} \Psi_- (x).
$$

(3.31)

where

$$
\Psi_+ (x) = - \left( k^2 - m^2 \right) S_x^2 (r) \left| \frac{V}{6} + i \sqrt{k^2 - m^2} 2 S_x' (r) \right| + \delta_+ e^{-2i \sqrt{k^2 - m^2} J_0} S_\pi (r) dr
$$

$$
\Psi_- (x) = - \left( k^2 - m^2 \right) S_x^2 (r) \left| \frac{V}{6} - i \sqrt{k^2 - m^2} 2 S_x' (r) \right| + \delta_- e^{2i \sqrt{k^2 - m^2} J_0} S_\pi (r) dr
$$

and $\delta_{\pm}$ are constants.

Since in the interior $S_x = 1$ it follows that $(S_x^2)'$ and $S_x^2$ vanish. So for $x \in [0, L_1]$

$$
\hat{u} (x, m) = c_{+} e^{i \sqrt{k^2 - m^2} x - \frac{k^2}{48} \delta_+} e^{-2i \sqrt{k^2 - m^2} x} + c_{-} e^{-i \sqrt{k^2 - m^2} x - \frac{k^2}{48} \delta_-} e^{2i \sqrt{k^2 - m^2} x}.
$$

(3.32)

To find the constants $c_{\pm}$ and $\delta_{\pm}$ we compare the coefficients of $h^2$ using the boundary conditions

$$
\hat{u} (0, m) = 1 + 0 \cdot h^2
$$

$$
\hat{u} (L_2, m) = 0 + 0 \cdot h^2
$$

and the approximation

$$
e^{a(x) + b(x)} \simeq e^{a(x)} \left( 1 + h^2 b(x) \right).
$$

(3.33)

Using (3.31, 3.32) near the boundaries

$$
\hat{u} (0, m) = c_{+} e^{-\frac{k^2}{48} \delta_+} + c_{-} e^{-\frac{k^2}{48} \delta_-} \simeq c_{+} \left( 1 - \frac{h^2}{48} \delta_+ \right) + c_{-} \left( 1 - \frac{h^2}{48} \delta_- \right)
$$
and
\[
\hat{u}(L_2, m) = c_- e^{i\sqrt{k^2 - m^2} \int_0^{L_2} S_x(r) dr - \frac{k^2}{4i} \Psi_+(L_2)} + c_- e^{-i\sqrt{k^2 - m^2} \int_0^{L_2} S_x(r) dr - \frac{k^2}{4i} \Psi_-(L_2)}
\]
\[
= c_+ e^{i\sqrt{k^2 - m^2} \int_0^{L_2} S_x(r) dr \left(1 - \frac{\eta^2}{48}\right)} + \eta e^{-i\sqrt{k^2 - m^2} \int_0^{L_2} S_x(r) dr \left(1 - \frac{\eta^2}{48}\right)}.
\]
Solving these equations we end that the values of $c_+$ and $c_-$ are the same as in the formula for $u_{F-pml}$ (3.11), and
\[
\delta_+ = 4i\sqrt{k^2 - m^2} \sqrt{L_2} \frac{\eta}{\eta - 1}
\]
\[
\delta_- = 4i\sqrt{k^2 - m^2} \sqrt{L_2} \frac{\eta}{\eta - 1}
\]
where the value of $\eta$ is given in (3.18). Inserting into (3.32), approximating using (3.33) and rearranging, we get the approximate solution of the high-order schemes inside the interior, which we label $u_{N-pml}$
\[
\left|u_{N-pml} - u_{F-pml}\right| = \frac{\eta}{\eta - 1} \left|e^{i\sqrt{k^2 - m^2} x} + e^{-i\sqrt{k^2 - m^2} x}\right| \sin \eta \tau y
\]
\[
= u_{F-pml} + \frac{\eta}{6} \sqrt{k^2 - m^2} \sqrt{L_2} \frac{\eta}{(\eta - 1)^2} \sin \left(\sqrt{k^2 - m^2} x\right) \sin \eta \tau y.
\]
Using our choice for $S_x$:
\[
\left|u_{N-pml} - u_{F-pml}\right| = \frac{\eta}{\eta - 1} \left|e^{i\sqrt{k^2 - m^2} x} + e^{-i\sqrt{k^2 - m^2} x}\right| \sin \eta \tau y
\]
\[
= \frac{\eta}{6} \sqrt{k^2 - m^2} \frac{\eta}{(\eta - 1)^2} \sin \left(\sqrt{k^2 - m^2} x\right) \sin \eta \tau y.
\]
One of our goals was to show that the use of the second order approximation inside the PML layer does not corrupt the high-order accuracy.
inside the interior. We can prove this for the $O(h^4)$ scheme (2.25) from the error estimate (3.35). Taking a large value of $\frac{2\pi \sqrt{1 - \alpha(L_2 - L_1)}}{p+1}$ then the term $2e^{-\frac{2\pi \sqrt{1 - \alpha(L_2 - L_1)}}{p+1}}$ is exponentially small and is negligible in respect to $O(h^r)$ for every integer $r$. Thus inside the interior

$$u_{N-pml} = u_{F-pml} + O\left(h^4\right) = u_{exact} + O\left(h^4\right) + O\left(h^2\right)e^{-\frac{2\pi \sqrt{1 - \alpha(L_2 - L_1)}}{p+1}}$$

where the $O(h^4)$ comes from the accuracy of the scheme. In the $O(h^6)$ scheme numerical computations demonstrate a similar error estimate (like (3.35)) which yields,

$$u_{N-pml} = u_{exact} + O\left(h^6\right).$$

An approach yielding a modified equation including both the $O(h^2)$ and $O(h^4)$ factors may be carried out to obtain similar estimates. We may also need in the $O(h^6)$ scheme analysis more derivatives for the function $S_x$. Thus we need $p \geq 4$, instead of $p \geq 2$ in the $O(h^4)$ scheme.

6. Numerical Results

We present several computational results for the Helmholtz equation in the semi-infinite strip. The region to the right of $L_1$ is replaced by a PML equation which is then truncated at $L_2$. In each computation we wish to demonstrate one of the properties that has been analyzed. For all the results we use a uniform grid-size, where $n$ denotes the number of gridpoints along the $y$ axis, so $h = \frac{\pi}{n}$. We measure the error in the maximum norm and in the square $[0, \pi] \times [0, \pi]$ and so $L_1 \geq \pi$. The numerical approximation is denoted by $\phi$. We use a standard LU
factorization to solve the linear systems that arise from the finite difference schemes. For one grids a LU solver is not efficient. We deal with iterative algorithms in chapter 4.

6.1. The case \( m > k \): evanescent waves. We wish to check our error estimate \( e^{-\sqrt{m^2-k^2(2L_2-L_1)}} \). For the test case we consider

\[
m = 5, \quad k = 4.5, \quad L_1 = \pi, \quad L_2 = \frac{6}{5}\pi, \quad p = 2, \quad \sigma = 20, \quad n = 32. \quad (3.36)
\]

In this case

\[
e^{-\sqrt{m^2-k^2(2L_2-L_1)}} \simeq 6.87 \times 10^{-5}
\]

We solve the problem with the \( O(h^6) \) solver (2.28) with the parameter \( \gamma = 1 \). The computed solution \( \phi \) approximates \( u_{N-pml} \) and satisfies:

\[
err = \| u_{\text{exact}} - \phi \| = 6.53 \times 10^{-5}
\]

Reusing the gridsize, taking \( n = 64 \) we get

\[
err = 7.19 \times 10^{-5}
\]

which shows that we cannot improve the accuracy of the computation because we reached our desired level of accuracy.

Choosing instead \( L_2 = 1.6\pi \) in (3.36) we get

\[
e^{-\sqrt{m^2-k^2(2L_2-L_1)}} \simeq 4.29 \times 10^{-7}
\]

and with \( n = 32 \) we get

\[
err = \pm 1 \times 10^{-6},
\]

with \( n = 64 \)

\[
err = 2.01 \times 10^{-7},
\]
and with $n = 128$

$$err = 2.13 \times 10^{-7}.$$

Now, the first refinement of the grid improves the result, but we cannot improve it further, as we can see in the second refinement. That is because we reached our lower bound.

Taking for a second test case

$$m = 5, \ k = 1, \ L_1 = \pi, \ L_2 = \frac{3}{2}\pi, \ p = 2, \ \sigma = 20,$$

$$n = 32, \ O(h^6) \ scheme \ with \ \gamma = 1$$

we have

$$e^{-\sqrt{m^2 - \sigma^2(2L_2 - L_1)}} \approx 4.28 \times 10^{-14}.$$

and

$$err_{base} = 8.18 \times 10^{-7}$$

we are far from the lower error bound. In Table 1 we can see the effect of the parameter change from the basic test case. We verify our results in Table 1:

- The independence of the error on the values of $p$ (results 1,2), $\sigma$ (result 3), $2L_2 - L_1$(result 10).

We see (result 9) that if we change $m, k$ in such a way that we preserve the value of $m^2 - k^2$, the error changes. We can explain this by the assumption that as $m$ increases the accuracy of the numerical schemes decreases.

- The efficiency of our high-order scheme for these problems. We can clearly verify the $O(h^6)$ behavior versus the $O(h^4)$ one (results 4-8)
3. THE INFINITE STRIP PROBLEM

<table>
<thead>
<tr>
<th>no.</th>
<th>parameter changed</th>
<th>value of $\text{err}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$p = 1$</td>
<td>$\text{err}_{\text{base}}$</td>
</tr>
<tr>
<td>2</td>
<td>$p = 4$</td>
<td>$\text{err}_{\text{base}}$</td>
</tr>
<tr>
<td>3</td>
<td>$\sigma = 100$</td>
<td>$\text{err}_{\text{base}}$</td>
</tr>
<tr>
<td>4</td>
<td>$n = 64$</td>
<td>$1.28 \times 10^{-8} \approx \left(\frac{1}{2}\right)^6 \cdot \text{err}_{\text{base}}$</td>
</tr>
<tr>
<td>5</td>
<td>$n = 128$</td>
<td>$1.99 \times 10^{-10} \approx \left(\frac{1}{2}\right)^6 \cdot \text{err}_{\text{base}}$</td>
</tr>
<tr>
<td>6</td>
<td>$O(h^6)$ scheme with $\gamma = 3$</td>
<td>$8.06 \times 10^{-7}$</td>
</tr>
<tr>
<td>7</td>
<td>$O(h^4)$ scheme with $\gamma = 1$, $n = 32$</td>
<td>$6.16 \times 10^{-6}$</td>
</tr>
<tr>
<td>8</td>
<td>$O(h^4)$ scheme with $\gamma = 1$, $n = 64$</td>
<td>$3.46 \times 10^{-7} \approx \left(\frac{1}{2}\right)^4 \cdot #7$</td>
</tr>
<tr>
<td>9</td>
<td>$m = 7$, $k = 5$</td>
<td>$3.58 \times 10^{-6}$</td>
</tr>
<tr>
<td>10</td>
<td>$L_1 = \frac{5}{4\pi}$, $L_2 = \frac{13}{5\pi}$</td>
<td>$\text{err}_{\text{base}}$</td>
</tr>
<tr>
<td>11</td>
<td>$k = 0.0000001$</td>
<td>$8.51 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Table 1. The Case $m > k$

- The value of $\gamma$ in the high order schemes is of minor importance and controls the parameter $c$ in the leading order of the error $ch^4$ or $ch^6$.

- We maintain the same behavior even for $k \to 0$ (result 11)

In a real problem we cannot control the value of $m$ (the input boundary condition). In the Fourier series expansion we have a complete set of non-zero values. We concentrate on the leading mode value.

6.2. The case $m < k$: traveling waves.

6.2.1. The dependence on the accuracy of the scheme. We wish to numerically verify our proof that the behavior in the interior depends only on the accuracy used in the interior. We examine the high-order schemes as well as the standard $O(h^2)$ scheme on a test case. To check this behavior we have to exclude errors that arise from other
approximations. Thus, we take in (3.19) $2e^{\frac{-2\sigma\sqrt{1-\varepsilon p}}{p+1}} \rightarrow 0$ and make sure that $\frac{h^2 \sigma \sqrt{1-\varepsilon p}}{6 (L_2 - L_1)} e^{\frac{-2\sigma\sqrt{1-\varepsilon p}}{p+1}}$ in (3.35) is negligible.

We take as our base test case

$$m = 6, \ k = 8, \ L_1 = \frac{5}{4} \pi, \ L_2 = \frac{7}{4} \pi, \ p = 2, \ \sigma = 25$$

In this case

$$2e^{\frac{-2\sigma\sqrt{1-\varepsilon p}}{p+1}} \simeq 6.03 \times 10^{-8}$$

and

$$\frac{\sigma \sqrt{1-\varepsilon p}}{6 (L_2 - L_1)} \simeq 3.5$$

and if $n \geq 16$

$$h^2 = \left( \frac{\pi}{n} \right)^2 \leq \left( \frac{\pi}{16} \right)^2 \simeq 3.8 \times 10^{-2}$$

and

$$h^2 \frac{\sigma \sqrt{1-\varepsilon p}}{6 (L_2 - L_1)} \leq 0.14$$

Thus, our high-order schemes should yield good results. In Table 2 we see the value of $err = \| u_{exact} - \phi \|_{\infty}$ resulting from the use of the schemes: PT (2.6), HO-4 (2.25) and HO-6 (2.28). Increasing the value of $n$ decreases the value of the gridsize $h$ in both directions. We assume that for small values of $h$,

$$err \simeq C(k) n^{-r} = \frac{C(k)}{\pi^r} h^r$$

where $r$ is the order of the scheme, i.e.

$$r = \begin{cases} 
2 & \text{for PT scheme} \\
4 & \text{for the schemes EB,H} \\
6 & \text{for the schemes EB-6,H} 
\end{cases}$$
Table 2. The value of \( err = \| u_{\text{exact}} - \phi \|_{\infty} \) for \( m = 6, \ k = 8, \ L_1 = \frac{5}{4} \pi, \ L_2 = \frac{7}{4} \pi, \ p = 2, \ \sigma = 25 \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>PT</th>
<th>HO-4 ( \gamma = 0 )</th>
<th>HO-4 ( \gamma = 2 )</th>
<th>HO-6 ( \gamma = 0 )</th>
<th>HO-6 ( \gamma = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>( 1.72 \times 10^0 )</td>
<td>( 5.91 \times 10^{-2} )</td>
<td>( 3.15 \times 10^{-1} )</td>
<td>( 4.25 \times 10^{-2} )</td>
<td>( 1.46 \times 10^{-1} )</td>
</tr>
<tr>
<td>32</td>
<td>( 4.85 \times 10^{-1} )</td>
<td>( 4.21 \times 10^{-3} )</td>
<td>( 1.97 \times 10^{-2} )</td>
<td>( 6.86 \times 10^{-4} )</td>
<td>( 2.32 \times 10^{-3} )</td>
</tr>
<tr>
<td>48</td>
<td>( 2.17 \times 10^{-1} )</td>
<td>( 8.54 \times 10^{-4} )</td>
<td>( 3.93 \times 10^{-3} )</td>
<td>( 6.98 \times 10^{-5} )</td>
<td>( 2.07 \times 10^{-4} )</td>
</tr>
<tr>
<td>64</td>
<td>( 1.23 \times 10^{-1} )</td>
<td>( 2.73 \times 10^{-4} )</td>
<td>( 1.25 \times 10^{-3} )</td>
<td>( 1.49 \times 10^{-5} )</td>
<td>( 3.72 \times 10^{-5} )</td>
</tr>
<tr>
<td>80</td>
<td>( 7.86 \times 10^{-2} )</td>
<td>( 1.13 \times 10^{-4} )</td>
<td>( 5.12 \times 10^{-4} )</td>
<td>( 4.74 \times 10^{-6} )</td>
<td>( 9.89 \times 10^{-6} )</td>
</tr>
<tr>
<td>96</td>
<td>( 5.46 \times 10^{-2} )</td>
<td>( 5.45 \times 10^{-5} )</td>
<td>( 2.47 \times 10^{-4} )</td>
<td>( 1.93 \times 10^{-6} )</td>
<td>( 3.38 \times 10^{-6} )</td>
</tr>
<tr>
<td>112</td>
<td>( 4.02 \times 10^{-2} )</td>
<td>( 2.95 \times 10^{-5} )</td>
<td>( 1.34 \times 10^{-4} )</td>
<td>( 9.07 \times 10^{-7} )</td>
<td>( 1.38 \times 10^{-6} )</td>
</tr>
<tr>
<td>128</td>
<td>( 3.08 \times 10^{-2} )</td>
<td>( 1.73 \times 10^{-5} )</td>
<td>( 7.85 \times 10^{-5} )</td>
<td>( 4.71 \times 10^{-7} )</td>
<td>( 6.47 \times 10^{-7} )</td>
</tr>
</tbody>
</table>

Figure 12. \(-\log(err)\) by \(\log(n)\) for the finite differences schemes

Hence, if \( n = 2^k \)

\[-\log_2(err) \simeq l \cdot r - \log_2 C(k) .\]

We calculate \( r \), the order of accuracy, by measuring the slope of the curve (in figure 12). This computationally verifies our hypothesis. The computing time required for all of these schemes, applied on a given
Table 3. Convergence towards $u_{ho}$ for $m = 1$, $k = 5$, $L_1 = \frac{5}{4}\pi$, $L_2 = \frac{3}{2}\pi$, $p = 2$, $\sigma = 20$ solved with the HO-4 scheme problem, is the same when we use Gaussian elimination. This fact emphasizes the benefit of using the high order schemes.

6.2.2. Convergence towards the modiﬁed solution. We wish to confirm the modiﬁed equation’s solution (3.34) which we labeled $u_{N-pml}$. We start with the $O(h^4)$ approximation with $\gamma = 0$, and solve the problem with

$$m = 1, \quad k = 5, \quad L_1 = \frac{5}{4}\pi, \quad L_2 = \frac{3}{2}\pi, \quad p = 2, \quad \sigma = 20.$$ 

In this case the lower error bound is rather poor and we get

$$2e^{-2\pi\sqrt{\pi L_1 L_2}} \approx 7 \times 10^{-5}.$$

We measure two kinds of errors

$$err_{ex} = \|u_{exact} - \phi\|_\infty$$

and

$$err_{ho} = \|u_{N-pml} - \phi\|_\infty$$

our goal is to show that the scheme converges to the $u_{N-pml}$ solution with $O(h^4)$. The results are collected in Table 3. We see that the
3. THE INFINITE STRIP PROBLEM

<table>
<thead>
<tr>
<th>( n )</th>
<th>( err_{ex} )</th>
<th>( err_{ho} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>( 6.19 \times 10^{-3} )</td>
<td>( 6.12 \times 10^{-3} )</td>
</tr>
<tr>
<td>32</td>
<td>( 5.64 \times 10^{-4} )</td>
<td>( 4.93 \times 10^{-4} )</td>
</tr>
<tr>
<td>64</td>
<td>( 9.91 \times 10^{-5} )</td>
<td>( 2.88 \times 10^{-5} )</td>
</tr>
<tr>
<td>128</td>
<td>( 7.18 \times 10^{-5} )</td>
<td>( 1.76 \times 10^{-6} )</td>
</tr>
</tbody>
</table>

Table 4. Convergence towards \( u_{ho} \) for \( m = 1, k = 5 \), \( L_1 = \frac{5}{4} \pi \), \( L_2 = \frac{3}{2} \pi \), \( p = 2 \), \( \sigma = 20 \) solved with the HO-6 scheme

<table>
<thead>
<tr>
<th>( n )</th>
<th>( err_{ex} )  = ( err_{ho} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>( 1.81 \times 10^{-3} )</td>
</tr>
<tr>
<td>32</td>
<td>( 6.29 \times 10^{-5} )</td>
</tr>
<tr>
<td>64</td>
<td>( 3.33 \times 10^{-6} )</td>
</tr>
<tr>
<td>128</td>
<td>( 1.99 \times 10^{-7} )</td>
</tr>
</tbody>
</table>

Table 5. Convergence towards \( u_{ho} \) for \( m = 1, k = 5 \), \( L_1 = \frac{5}{4} \pi \), \( L_2 = 2 \pi \), \( p = 2 \), \( \sigma = 20 \) solved with the HO-6 scheme

The HO-6 scheme approximates the \( u_{N-pml} \) solution and

\[
\frac{err_{ho}(n = 32)}{err_{ho}(n = 16)} \sim \frac{err_{ho}(n = 64)}{err_{ho}(n = 32)} \sim \frac{err_{ho}(n = 128)}{err_{ho}(n = 64)} \sim \left( \frac{1}{2} \right)^{-4}
\]

which is consistent with our analysis. The actual solution is \( u_{exact} \), so in order to converge towards this solution we set the parameters such that the exponent \( 2e^{-2\pi\sqrt{\frac{\sigma(L_2-L_1)}{p+1}}} \) is small enough.

We solve the same problem with the \( O(h^6) \) scheme with \( \gamma = 0 \) (results in Table 4). Again we see that we approach \( u_{N-pml} \) and not \( u_{exact} \), but this time we do not get a \( O(h^6) \) approximation. For instance

\[
\frac{err_{ho}(n=64)}{err_{ho}(n=32)} \sim \left( \frac{1}{2} \right)^{-4} \text{ and not } \left( \frac{1}{2} \right)^{-6}.
\]

In order to avoid the cause we change
6. NUMERICAL RESULTS

<table>
<thead>
<tr>
<th>$n$</th>
<th>$err_{ex} = err_{ho}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>$4.77 \times 10^{-3}$</td>
</tr>
<tr>
<td>32</td>
<td>$1.15 \times 10^{-5}$</td>
</tr>
<tr>
<td>64</td>
<td>$1.87 \times 10^{-7}$</td>
</tr>
<tr>
<td>128</td>
<td>$2.98 \times 10^{-9}$</td>
</tr>
</tbody>
</table>

Table 6. Convergence towards $u_{ho}$ for $m = 1$, $k = 5$, $L_1 = \frac{5}{4} \pi$, $L_2 = \frac{7}{3} \pi$, $p = 4$, $\sigma = 200$ solved with the HO-6 scheme

$L_2$ to $2\pi$, to achieve better accuracy (Table 5). This time $err_{ho}(n=64) \approx \left(\frac{1}{2}\right)^{-4.48}$, and in other problems (last subsection) we do get $O(h^6)$. The problem is that the evaluated $u_{N-pmt}$ is not what we are approximating in the $O(h^6)$ scheme. In the analysis of the modied equation we approximate the $O(h^2)$ term and neglect the $O(h^4)$ term. However, this high order term is signiicant in some problems and in some is negligible. $p = 2$ can be accurate for some problems (see Table 2), but we should choose, in general, $p \geq 4$ for the $O(h^6)$ scheme. For instance in Table 6, we do get the desired approximation

$$\frac{err_{ho}(n=64)}{err_{ho}(n=32)} \approx \frac{err_{ho}(n=128)}{err_{ho}(n=64)} \approx \left(\frac{1}{2}\right)^{-6}.$$ 

6.2.3. Selecting the parameters in the function $\sigma_x$. We wish to check the influence of the parameters we choose in the approximation, $L_1$, $L_2$, $L_1$, $p$ and $\sigma$, on a given problem.

Value of $L_1$: The error estimates we obtained imply that the value of $L_1$ is relatively unimportant (we approximated $|\frac{2}{n-1}|$ that depends on $L_1$ by $|\frac{2}{n}|$). We check this with the test case
\[ \begin{array}{|c|c|c|}
\hline
L_1 & \frac{2}{\eta - 1} & \text{err} \\
\hline
1 & 4.91 \times 10^{-14} & 3.01 \times 10^{-3} \\
1.25 & 4.73 \times 10^{-14} & 3.33 \times 10^{-3} \\
1.5 & 4.53 \times 10^{-14} & 2.85 \times 10^{-3} \\
1.75 & 4.30 \times 10^{-14} & 3.44 \times 10^{-3} \\
2 & 4.03 \times 10^{-14} & 2.68 \times 10^{-3} \\
2.25 & 3.75 \times 10^{-14} & 3.54 \times 10^{-3} \\
2.5 & 3.44 \times 10^{-14} & 2.50 \times 10^{-3} \\
\hline
\end{array} \]

Table 7. The error in respect to the value of \( L_1 \)

\[ \begin{align*}
m & = 1, \ k = 10, \ L_1 - L_2 = \frac{\pi}{4}, \ p = 4, \ \sigma = 100, \\
n & = 32, \ O(h^6) \ 	ext{scheme with } \gamma = 0,
\end{align*} \]

so the value of \( \left| \frac{2}{\eta} \right| = 2e^{-2\sigma \sqrt{\frac{\pi}{p+1} (L_2 - L_1)}} \simeq 5.36 \times 10^{-14} \). We summarize the results in Table 7.

\textbf{Width of the PML, } \( L_2 - L_1 \): To support our theory we show that the accuracy increases as we take a thicker PML layer. Taking

\[ \begin{align*}
m & = 1, \ k = 10, \ L_1 = \frac{5\pi}{4}, \ p = 4, \ \sigma = 100, \ O(h^6) \ 	ext{scheme with } \gamma = 0,
\end{align*} \]

we summarize the results in Table 8. We can clearly verify that as the order of accuracy increases we maintain the \( O(h^6) \) accuracy if the PML is thick enough. Increasing the PML width over a certain limit does not help us and we maintain the same accuracy while having more storage and computational time. For small values of \( k \) we can obtain
6. NUMERICAL RESULTS

<table>
<thead>
<tr>
<th>$L_2 - L_1$</th>
<th>$\frac{\pi}{16}$</th>
<th>$8.08 \times 10^{-4}$</th>
<th>$3.13 \times 10^{-1}$</th>
<th>$8.53 \times 10^{-3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{\pi}{8}$</td>
<td>$3.26 \times 10^{-7}$</td>
<td>$5.92 \times 10^{-2}$</td>
<td>$6.08 \times 10^{-5}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{\pi}{4}$</td>
<td>$5.31 \times 10^{-14}$</td>
<td>$3.33 \times 10^{-3}$</td>
<td>$3.34 \times 10^{-5}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{\pi}{2}$</td>
<td>$1.41 \times 10^{-27}$</td>
<td>$2.16 \times 10^{-3}$</td>
<td>$3.33 \times 10^{-5}$</td>
<td></td>
</tr>
<tr>
<td>$\pi$</td>
<td>$9.98 \times 10^{-55}$</td>
<td>$2.16 \times 10^{-3}$</td>
<td>$3.33 \times 10^{-5}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 8. The error in the HO-6 scheme in respect to the width of the PML.

<table>
<thead>
<tr>
<th>$L_2 - L_1$</th>
<th>$k = 2.5$</th>
<th>$k = 5$</th>
<th>$k = 13$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{\pi}{16}$</td>
<td>$1.68 \times 10^{-1}$</td>
<td>$2.42 \times 10^{-1}$</td>
<td>$4.21 \times 10^{-1}$</td>
</tr>
<tr>
<td>$\frac{\pi}{8}$</td>
<td>$1.34 \times 10^{-2}$</td>
<td>$2.16 \times 10^{-2}$</td>
<td>$1.10 \times 10^{-1}$</td>
</tr>
<tr>
<td>$\frac{\pi}{4}$</td>
<td>$5.16 \times 10^{-5}$</td>
<td>$1.78 \times 10^{-4}$</td>
<td>$1.26 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\frac{\pi}{2}$</td>
<td>$1.03 \times 10^{-7}$</td>
<td>$1.23 \times 10^{-5}$</td>
<td>$1.44 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\pi$</td>
<td>$1.80 \times 10^{-8}$</td>
<td>$1.24 \times 10^{-5}$</td>
<td>$1.44 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Table 9. The error in the HO-6 scheme in respect to the width of the PML and to the value of $k$ for $n = 32$.

more accurate results, but we then need a thicker PML as seen in Table 9.

Our main interest is to take a thin PML, while maintaining the accuracy. We control this by enlarging the value of $\sigma$. In Table 10 we set $2\sigma \sqrt{1+n(L_2 - L_1)} = 1.41 \times 10^{-27}$ for the test problem (3.37). We conclude that as long as the PML is not too thin and we have enough points in the PML, we get the desired accuracy.

**Fixed number of points in the PML:** Computational practice, is to set a fixed number of points inside the PML. Thus, when
3. THE INFINITE STRIP PROBLEM

<table>
<thead>
<tr>
<th>$L_2 - L_1$</th>
<th>$\sigma$</th>
<th>$\text{err}$</th>
<th>$\text{err}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{\pi}{8}$</td>
<td>400</td>
<td>$1.7 \times 10^{-1}$</td>
<td>$9.73 \times 10^{-4}$</td>
</tr>
<tr>
<td>$\frac{\pi}{4}$</td>
<td>200</td>
<td>$5.04 \times 10^{-3}$</td>
<td>$3.36 \times 10^{-5}$</td>
</tr>
<tr>
<td>$\frac{\pi}{2}$</td>
<td>100</td>
<td>$2.16 \times 10^{-3}$</td>
<td>$3.33 \times 10^{-5}$</td>
</tr>
<tr>
<td>$\pi$</td>
<td>50</td>
<td>$2.16 \times 10^{-3}$</td>
<td>$3.33 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Table 10. The error in the HO-6 scheme for a constant value of $2\varepsilon \frac{2\sqrt{1-\sigma^2(L_2 - L_1)}}{\rho + 1}$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\sigma$</th>
<th>$L_2 - L_1$</th>
<th>HO-4</th>
<th>HO-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>$\frac{\pi}{2}$</td>
<td>$9.66 \times 10^{-1}$</td>
<td>$1.40 \times 10^{-1}$</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>$\frac{\pi}{4}$</td>
<td>$5.15 \times 10^{-2}$</td>
<td>$3.02 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>$75$</td>
<td>$3.62 \times 10^{-4}$</td>
<td>$9.16 \times 10^{-4}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 11. Results for exact 8 points inside the PML with $m = 2$, $k = 10$, $L_1 = \frac{5\pi}{4}$, $p = 2$, $\frac{\sigma(L_2 - L_1)}{\rho + 1} = \frac{25\pi}{8}$

we decrease the gridsize, we change the physical width of the PML. In the above computations we choose the physical width of the PML as constant. We want to see if this influences our results. To check the behavior we set a problem with

$$m = 2, \ k = 10, \ L_1 = \frac{5\pi}{4}$$

and for different values of gridsize $h$, use the same number of points inside the PML. In the first test (tables 11 - 13) we use 8 points in the PML, and in the second test (Table 14 and 15) 16 points. We see that we lose the $O(h^6)$ behavior when $p = 2$ (tables 11 and 14), but maintain the $O(h^4)$ behavior. This is because for the $O(h^6)$ scheme we have to take $p \geq 4$ and have sufficient width of PML (compare with
6. NUMERICAL RESULTS

<table>
<thead>
<tr>
<th>n</th>
<th>$\sigma$</th>
<th>$L_2 - L_1$</th>
<th>HO-4</th>
<th>HO-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>31.25</td>
<td>$\frac{\pi}{2}$</td>
<td>$1.06 \times 10^0$</td>
<td>$1.08 \times 10^{-1}$</td>
</tr>
<tr>
<td>32</td>
<td>62.5</td>
<td>$\frac{\pi}{4}$</td>
<td>$5.14 \times 10^{-2}$</td>
<td>$1.79 \times 10^{-3}$</td>
</tr>
<tr>
<td>64</td>
<td>125</td>
<td>$\frac{\pi}{8}$</td>
<td>$3.14 \times 10^{-3}$</td>
<td>$7.83 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Table 12. Results for $\xi$ed 8 points inside the PML with $m = 2$, $k = 10$, $L_1 = \frac{5\pi}{4}$, $p = 4$, $\frac{\sigma(L_2 - L_1)}{p+1} = \frac{25\pi}{8}$

<table>
<thead>
<tr>
<th>n</th>
<th>$\sigma$</th>
<th>$L_2 - L_1$</th>
<th>HO-4</th>
<th>HO-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>18.75</td>
<td>$\frac{\pi}{2}$</td>
<td>$1.01 \times 10^0$</td>
<td>$1.34 \times 10^{-1}$</td>
</tr>
<tr>
<td>32</td>
<td>37.5</td>
<td>$\frac{\pi}{4}$</td>
<td>$5.16 \times 10^{-2}$</td>
<td>$1.64 \times 10^{-3}$</td>
</tr>
<tr>
<td>64</td>
<td>75</td>
<td>$\frac{\pi}{8}$</td>
<td>$3.17 \times 10^{-3}$ (2.55 $\times 10^{-5}$ towards $u_{\text{N-pml}}$)</td>
<td></td>
</tr>
</tbody>
</table>

Table 13. Results for $\xi$ed 8 points inside the PML with $m = 2$, $k = 10$, $L_1 = \frac{5\pi}{4}$, $p = 4$, $\frac{\sigma(L_2 - L_1)}{p+1} = \frac{25\pi}{8}$

<table>
<thead>
<tr>
<th>n</th>
<th>$\sigma$</th>
<th>$L_2 - L_1$</th>
<th>HO-4</th>
<th>HO-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>9.375</td>
<td>$\pi$</td>
<td>$9.94 \times 10^{-1}$</td>
<td>$1.23 \times 10^{-1}$</td>
</tr>
<tr>
<td>32</td>
<td>18.75</td>
<td>$\frac{\pi}{2}$</td>
<td>$5.16 \times 10^{-2}$</td>
<td>$1.67 \times 10^{-3}$</td>
</tr>
<tr>
<td>64</td>
<td>37.5</td>
<td>$\frac{\pi}{4}$</td>
<td>$3.21 \times 10^{-3}$</td>
<td>$1.19 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 14. Results for $\xi$ed 16 points inside the PML with $m = 2$, $k = 10$, $L_1 = \frac{5\pi}{4}$, $p = 2$, $\frac{\sigma(L_2 - L_1)}{p+1} = \frac{25\pi}{8}$

<table>
<thead>
<tr>
<th>n</th>
<th>$\sigma$</th>
<th>$L_2 - L_1$</th>
<th>HO-4</th>
<th>HO-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>15.625</td>
<td>$\pi$</td>
<td>$9.94 \times 10^{-1}$</td>
<td>$1.24 \times 10^{-1}$</td>
</tr>
<tr>
<td>32</td>
<td>31.25</td>
<td>$\frac{\pi}{2}$</td>
<td>$5.17 \times 10^{-2}$</td>
<td>$1.65 \times 10^{-3}$</td>
</tr>
<tr>
<td>64</td>
<td>62.5</td>
<td>$\frac{\pi}{4}$</td>
<td>$3.17 \times 10^{-3}$</td>
<td>$2.52 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Table 15. Results for $\xi$ed 16 points inside the PML with $m = 2$, $k = 10$, $L_1 = \frac{5\pi}{4}$, $p = 4$, $\frac{\sigma(L_2 - L_1)}{p+1} = \frac{25\pi}{8}$
3. THE INFINITE STRIP PROBLEM

the results in the last section). For thin PML layers (Table 12) we cannot maintain the $O(h^6)$ behavior if we choose a large $\sigma$. However, by taking smaller value of $\sigma$ we lose the approximation to $u_{\text{exact}}$ and instead maintain the $O(h^6)$ accuracy only with the modified equation solution $u_{N-pml}$. Thus, if we wish to work with a fixed number of points inside the PML we could set the parameters such that $2e^{-\frac{3\sqrt{2^p - 1}}{p+1}}$ is small enough and choose $p \geq 4$ for HO-6 and $p = 2$ for HO-4.

We denote $n'_{\text{pml}}$ as the number of points in the PML

$$2e^{-\frac{3\sqrt{2^p - 1}}{p+1}} = 2e^{-\frac{2h_{\text{pml}} \cdot \sigma \sqrt{1-\gamma}}{p+1}}.$$ 

If we know the minimal value of $h$ priori (In tables 11-15 it is $\frac{\pi}{64}$), we can set $n'_{\text{pml}}$ and $\sigma$ to satisfy a desired estimate for $2e^{-\frac{2h_{\text{pml}} \cdot \sigma \sqrt{1-\gamma}}{p+1}}$.

Value of $p$ : We solve the test problem with various values of $p$ (Table 16) with

$m = 1, k = 10, L_1 = \frac{5\pi}{4}, L_2 = \frac{7\pi}{4}, \sigma = 100, O(h^6)$ scheme with $\gamma = 0$.

We clearly see the bad behavior of the case $p = 1$ and the need for sufficient derivatives in $S_x$ as seen in Figures 8-11. In the $O(h^6)$ case we should choose $p \geq 4$. Moreover, we do not benefit from taking larger values of $p$ than four.

Value of $\sigma$ : Taking as a test problem with various values of $\sigma$,

$m = 1, k = 10, L_1 = \frac{5\pi}{4}, L_2 = \frac{3\pi}{2}, p = 4, O(h^6)$ scheme with $\gamma = 0$.

We present the results in Table 17.
## 6. Numerical Results

| $p$  | $\text{estimate}$ (3.19) | $err$  
$n = 32$ | $err$  
$n = 64$ | $err$  
$n = 128$ | $\frac{err_{32}}{err_{64}}$ | $\frac{err_{64}}{err_{128}}$ |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$1.32 \times 10^{-68}$</td>
<td>$1.19 \times 10^{-1}$</td>
<td>$2.08 \times 10^{-2}$</td>
<td>$4.86 \times 10^{-3}$</td>
<td>5.7</td>
<td>2.1</td>
</tr>
<tr>
<td>2</td>
<td>$1.12 \times 10^{-45}$</td>
<td>$1.98 \times 10^{-3}$</td>
<td>$7.70 \times 10^{-5}$</td>
<td>$7.59 \times 10^{-6}$</td>
<td>25.7</td>
<td>25.1</td>
</tr>
<tr>
<td>3</td>
<td>$2.30 \times 10^{-34}$</td>
<td>$2.22 \times 10^{-3}$</td>
<td>$3.50 \times 10^{-5}$</td>
<td>$6.21 \times 10^{-7}$</td>
<td>63.4</td>
<td>56.3</td>
</tr>
<tr>
<td>4</td>
<td>$1.41 \times 10^{-27}$</td>
<td>$2.16 \times 10^{-3}$</td>
<td>$3.33 \times 10^{-5}$</td>
<td>$5.21 \times 10^{-7}$</td>
<td>64.9</td>
<td>63.9</td>
</tr>
<tr>
<td>5</td>
<td>$4.74 \times 10^{-23}$</td>
<td>$2.16 \times 10^{-3}$</td>
<td>$3.33 \times 10^{-5}$</td>
<td>$5.21 \times 10^{-7}$</td>
<td>64.9</td>
<td>63.9</td>
</tr>
<tr>
<td>10</td>
<td>$4.56 \times 10^{-13}$</td>
<td>$2.02 \times 10^{-3}$</td>
<td>$3.33 \times 10^{-5}$</td>
<td>$5.21 \times 10^{-7}$</td>
<td>61.2</td>
<td>63.9</td>
</tr>
<tr>
<td>30</td>
<td>$8.35 \times 10^{-5}$</td>
<td>$7.23 \times 10^{-2}$</td>
<td>$5.24 \times 10^{-4}$</td>
<td>$7.69 \times 10^{-4}$</td>
<td>138.0</td>
<td>&lt; 1</td>
</tr>
</tbody>
</table>

Table 16. Results for different values of $p$ for $m = 1, k = 10, L_1 = \frac{5\pi}{4}, L_2 = \frac{3\pi}{2}, \sigma = 100, O(h^6)$ scheme with $\gamma = 0$

| $\sigma$ | $\text{estimate}$ (3.19) | $err$  
$n = 32$ | $err$  
$n = 64$ | $err$  
$n = 128$ | $\frac{err_{32}}{err_{64}}$ | $\frac{err_{64}}{err_{128}}$ |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$8.77 \times 10^{-2}$</td>
<td>$9.54 \times 10^{-2}$</td>
<td>$9.23 \times 10^{-2}$</td>
<td>$9.16 \times 10^{-2}$</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>20</td>
<td>$3.85 \times 10^{-3}$</td>
<td>$4.59 \times 10^{-3}$</td>
<td>$3.94 \times 10^{-3}$</td>
<td>$3.88 \times 10^{-3}$</td>
<td>1.2</td>
<td>1.0</td>
</tr>
<tr>
<td>40</td>
<td>$7.43 \times 10^{-6}$</td>
<td>$2.03 \times 10^{-3}$</td>
<td>$3.39 \times 10^{-5}$</td>
<td>$7.43 \times 10^{-6}$</td>
<td>59.8</td>
<td>4.6</td>
</tr>
<tr>
<td>80</td>
<td>$2.75 \times 10^{-11}$</td>
<td>$3.24 \times 10^{-3}$</td>
<td>$3.34 \times 10^{-5}$</td>
<td>$5.21 \times 10^{-7}$</td>
<td>97</td>
<td>64.1</td>
</tr>
<tr>
<td>160</td>
<td>$3.81 \times 10^{-22}$</td>
<td>$9.28 \times 10^{-3}$</td>
<td>$3.35 \times 10^{-5}$</td>
<td>$5.21 \times 10^{-7}$</td>
<td>277</td>
<td>64.3</td>
</tr>
<tr>
<td>320</td>
<td>$7.25 \times 10^{-44}$</td>
<td>$1.20 \times 10^{-2}$</td>
<td>$3.42 \times 10^{-5}$</td>
<td>$5.30 \times 10^{-7}$</td>
<td>350</td>
<td>64.5</td>
</tr>
<tr>
<td>$10^3$</td>
<td>$1.31 \times 10^{-136}$</td>
<td>$3.63 \times 10^{-2}$</td>
<td>$4.21 \times 10^{-5}$</td>
<td>$5.89 \times 10^{-7}$</td>
<td>86.6</td>
<td>71.5</td>
</tr>
</tbody>
</table>

Table 17. Results for different values of $\sigma$ for $m = 1, k = 10, L_1 = \frac{5\pi}{4}, L_2 = \frac{3\pi}{2}, p = 4, O(h^6)$ scheme with $\gamma = 0$
We conclude that the value of $\sigma$ does not influence the accuracy of the approximation as long as $2e^{-\frac{2\pi \sqrt{L_2 - L_1}}{p+1}}$ is small enough. Thus, a natural choice of $\sigma$ should be $\sigma > \hat{k}$ such that

$$2e^{-\frac{2\pi \sqrt{L_2 - L_1}}{p+1}} \sim 10^{-q}$$

(3.38)

where $q$ is the number of the desired significant digits accuracy. We choose the lowest $\sigma$ that satisfies this.
CHAPTER 4

Krylov Subspace Methods for the Strip Problem

In this chapter we consider suitable algorithms to solve the linear system resulting from the combined PML-Helmholtz problem with the schemes developed in Chapters 2, 3. We start with describing some direct methods, and then describe several iterative Krylov subspace algorithms. We want to examine the rate of convergence to the steady state of the different algorithms and show how the parameters of the PML: its width, placement and the decaying function affect them. Because the standard methods converge slowly we consider preconditioned Krylov subspace methods. We start with diagonal scaling, and examine the benefit we get from using it. We then search for preconditioners designed specially for the Helmholtz equation and the combined problem.

We concentrate in this chapter on algorithms for solving the model problem represented in section 6 in the last chapter. Throughout this chapter we denote \( n \) as the number of the gridpoints in the \( y \) direction, so \( h = \frac{\pi}{n} \). The PML stretches from \( L_1 \) to \( L_2 \) so the matrix \( A \) in the resulting linear system \( Ax = b \) is of order \( N \times N \) where

\[
N = (n - 1) (L_2 n - 1) \simeq L_2 n^2.
\]

The matrix \( A \) is banded with 9 diagonals. We denote \( \varphi \) as the exact solution to the linear system, meaning \( A \varphi = b \). In our calculations we
used the Euclidean inner product:

$$\langle u, w \rangle = \sum_{i=1}^{N} u_i \overline{w_i} \quad \text{for } u, w \in \mathbb{C}^N.$$ 

In the calculations of any algorithm cost we assume one unit of cost per arithmetic operation. Thus, the inner product costs about $2N$ units, matrix vector multiplication $17N$ units (in each row of $A$ there are at most 9 non-zero entries so we have 9 multiplications and 8 summations) and linear combinations

$$v = v + \alpha w, \quad \text{for } u, w \in \mathbb{C}^N, \alpha \in \mathbb{C}$$

cost $2N$ units.

1. Direct Methods for Linear Systems

Direct methods, like the LU decomposition and other factorizations that use the fact that the matrix $A$ is banded, are usually not efficient when trying to solve the linear systems obtained from solving PDEs by any kind of discretization, including the combined Helmholtz-PML problem. There are two major problems with such methods. The first is that they require storing the matrix $A$ and not just its non-zero diagonals, and the second is that for very large matrices these methods need a lot of CPU time. These difficulties increase in three dimensions. The numerical results shown in Chapter 3 are calculated with a band LU factorization algorithm without pivoting coupled with forward and backward substitution [15, 152-153].
Algorithm 1. Band LU Factorization

Given a square $N \times N$ matrix $A$ with upper and lower bandwidth $L = n$ the following algorithm replaces the matrix $A$ with its $LU$ factorization without pivoting

\[
\begin{align*}
\text{do } & \ k = 1, N - 1 \\
\text{do } & \ i = k + 1, \min(k + L, N) \\
\quad & A_{i,k} = \frac{A_{i,k}}{A_{k,k}} \\
\text{endo} \\
\text{do } & \ j = k + 1, \min(k + L, N) \\
\quad & \text{do } i = k + 1, \min(k + L, N) \\
\quad & A_{i,j} = A_{i,j} - A_{i,k}A_{k,j} \\
\quad & \text{endo} \\
\text{endo} \\
\text{endo}
\end{align*}
\]

We have never seen the need for pivoting even when the matrix was not diagonally dominant. The cost of this algorithm is approximately

\[2 (L)^3 n^4. \tag{4.1}\]

In order to solve $LUx = b$ we need two substitutions, forward and backward:

Algorithm 2. Band Forward and Backward substitutions

The algorithm replaces the vector $b$ in $LUx = b$ with its solution

\[
\begin{align*}
\text{do } & \ j = 1, n \\
\text{do } & \ i = j + 1, \min(j + L, N) \\
\quad & b_i = b_i - L_{i,j}b_j \\
\text{endo} \\
\text{endo}
\end{align*}
\]
4. KRYLOV SUBSPACE METHODS FOR THE STRIP PROBLEM

\[ \text{do } j = n, 1, -1 \]
\[ b_j = \frac{b_j}{U_{jj}} \]
\[ \text{do } i = \max(1, j - L_2n), j - 1 \]
\[ b_i = b_i - U_{i,j} b_j \]
\[ \text{enddo} \]
\[ \text{enddo} \]

The algorithm requires approximately \( 4 (L_2)^2 n^3 \) computations, which is negligible in respect to the cost of the \( LU \) factorization (4.1). The standard \( LU \) decomposition cost is \( \frac{2}{3} N^3 \) \[15, 100\] and in our case \( \frac{2}{3} (L_2)^3 n^6 \) which is clearly inferior compare to (4.1). Thus, the usage of the banded property of the matrix is very important.

2. Iterative Methods for Sparse Linear Systems

For linear systems of small size the standard approach is to use a direct method, as described in the last section. In practical problems, particularly when solving partial PDEs, the linear system is large and sparse and it is very useful in terms of storage requirements and computing time to apply an iterative technique.

In an iterative method we start with an initial vector \( x_0 \) and generate a sequence of vectors

\[ x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \ldots \]

which (hopefully) converge toward the solution of the numerical approximation. In each iteration the basic step is a matrix-vector multiplication with the matrix \( A \). Since the matrix is sparse this multiplication is relatively cheap. The efficiency of the iterative method depends on the cost of each iteration and on the rate of convergence i.e. how many iterations are needed to converge towards a given tolerance. Roughly, we can define the total cost of the algorithm as the
multiplication of the two quantities

\[ \text{total cost} = \text{number of iterations} \times \text{cost per iteration}. \]

There exists many iterative techniques, such as the Jacobi and Gauss-Seidel methods and the more complicated ADI and multigrid techniques. We shall discuss the solution using Krylov subspace methods.

We choose an initial guess \( x_0 \), compute the residual \( r_0 = b - Ax_0 \) and choose \( x_1 \)

\[ x_1 \in x_0 + \text{span} \{ r_0 \}. \]

We want to choose \( x_1 \) to minimize the error in a given norm. We continue and construct the next \( x_k \), using multiplications with a matrix \( C \) which is related to \( A \),

\[ x_k \in \text{span} \{ r_0, Cr_0, C^2 r_0, C^3 r_0, \ldots, C^{k-1} r_0 \} = K_k (r_0, C) \quad k = 1, 2, \ldots \tag{4.2} \]

The subspace defined in (4.2) is called a Krylov subspace spanned by \( C \) and \( r_0 \). We wish the algorithm to converge as we take more iterations.

2.1. Conjugate gradient. The Conjugate Gradient method (CG) was first developed by Hestenes and Steifel for Hermitian positive definite matrices (1952). The idea of the method comes from the observation that the exact solution \( \phi \) minimizes the functional \( F : \mathbb{R}^n \rightarrow \mathbb{R} \)

\[ F(z) = \frac{1}{2} (b - Az)^T A^{-1} (b - Az) \]

Clearly \( F(z) \geq 0 \) and \( F(z) = 0 \iff z = \phi \). Using the method of steepest descent to minimize the functional, starting with an initial guess \( x_0 \), gives the standard CG algorithm which converges for Hermitian
matrices, which satisfy
\[ A_{i,j} = \overline{A_{j,i}} \quad 1 \leq i, j \leq N, \]
and also are positive definite.

Algorithm 3. Conjugate Gradient Method (CG)

The following algorithm computes the iteration \( x_k \) to the solution of the system \( Ax = b \) where \( A \) is positive definite Hermitian matrix.

choose an initial guess \( x_0 \) and set \( p_0 = r_0 = b - Ax_0 \)
do \( k = 1, 2, \ldots \)
compute \( Ap_k \)
\[ \alpha_{k-1} = \frac{\langle r_{k-1}, r_{k-1} \rangle}{\langle p_{k-1}, Ap_{k-1} \rangle} \]
\[ x_k = x_{k-1} + \alpha_{k-1} p_{k-1} \]
\[ r_k = r_{k-1} - \alpha_{k-1} Ap_{k-1} \]
if \( \| r_k \| < \varepsilon \) then the computed solution is \( x_k \) and stop
\[ \beta_{k-1} = \frac{\langle r_k, r_k \rangle}{\langle r_{k-1}, r_{k-1} \rangle} \]
\[ p_k = r_k + \beta_{k-1} p_{k-1} \]
endo

Here \( \langle \cdot, \cdot \rangle \) is any inner product. For instance, we choose the standard Euclidean inner product. \( \varepsilon \) is a suitably small stopping criteria. The iteration \( x_k \) to the exact solution is chosen from
\[ x_0 + \text{span} \{ r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0 \} = x_0 + \mathcal{K}_k(A, r_0) \]
so that
\[ \langle x^k - \varphi, A(x^k - \varphi) \rangle \]
is minimized. For our problem convergence is not guaranteed because the matrix is not positive definite. Furthermore, in the PML it is complex and not Hermitian. Thus, we are not surprised that the CG algorithm does not converge.
2. ITERATIVE METHODS FOR SPARSE LINEAR SYSTEMS

<table>
<thead>
<tr>
<th>$n$</th>
<th>$err$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>$1.24 \times 10^{-1}$</td>
</tr>
<tr>
<td>32</td>
<td>$8.33 \times 10^{-4}$</td>
</tr>
<tr>
<td>64</td>
<td>$5.22 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Table 1. The error $err$ in $L_\infty$ norm between the exact solution to the test case and the numerical solution with respect to the number of gridpoints. The system is solved with the $LU$ factorization.

Hence, to solve the problem we need other algorithms that converge for our system. In the following subsections we examine standard Krylov-subspace iterative algorithms for non-Hermitian matrices. We apply these algorithms on the following model problem:

$$m = 2, \ k = 4, \ L_1 = \frac{5\pi}{4}, \ L_2 = \frac{3\pi}{2}, \ p = 4, \ \sigma = 300, \ O(h^6) \ scheme \ with \ \gamma = 0.$$ 

The solution to the linear system for the test case is calculated with the algorithms 1, 2. The error is given up to 3 significant digits in Table 1. In order to be sure that the error with an iterative algorithm is not affected by the number of iterations we choose $\varepsilon = 10^{-12}$ as the stopping criteria in all the iterative algorithms. This is more than needed for engineering accuracy. However, it is necessary for our applications of comparing the influence of the various parameters.

2.2. Conjugate gradient for the normal equation. Instead of solving the given system $Ax = b$ we apply the CG algorithm to the system

$$A^H A x = A^H b$$
where the superscript $^H$ denotes the complex conjugate transpose.
The matrix $A^H A$ is guaranteed to be Hermitian and positive definite, so the CG algorithm must converge. In practice we can use some inner-product identities to simplify this version of CG, resulting in the conjugate gradient normal-equation residual - CGNR algorithm, having

$$x_k \in x_0 + \mathcal{K}_k \left( A^H r_0, A^H A \right).$$

Algorithm 4. CGNR

Given a square $N \times N$ nonsingular complex matrix $A$ and $b \in \mathbb{C}^N$. The following algorithm computes the iteration $x_k$ to $x$ the solution of $Ax = b$.

choose an initial guess $x_0$ and set $r_0 = b - Ax_0$

compute $A^H r_0$ and set $p_0 = A^H r_0$

do $k = 1, 2, \ldots$

compute $A p_{k-1}$

$$\alpha_{k-1} = \frac{\langle A^H r_{k-1}, A^H r_{k-1} \rangle}{\langle p_{k-1}, A p_{k-1} \rangle}$$

$$x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$$

$$r_k = r_{k-1} - \alpha_{k-1} A p_{k-1}$$

if $\|r_k\| < \varepsilon$ then the computed solution is $x_k$ and stop

compute $A^H r_k$

$$\beta_{k-1} = \frac{\langle A^H r_k, A^H r_k \rangle}{\langle A^H r_{k-1}, A^H r_{k-1} \rangle}$$

$$p_k = A^H r_k + \beta_{k-1} p_{k-1}$$

endo

As expected, the algorithm converges as seen in Table 2. We shall now designate the iteration level as $l$ to distinguish from the wave number $k$. 
The large amount of iterations is not surprising because a well-known fact is that the rate of convergence in the CG algorithm depends on the condition number of the matrix, having

$$\frac{\|r_l\|}{\|r_0\|} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t,$$

where $\kappa$ is the condition number of $A$ [21, 187]. In the CGNR method we apply the CG algorithm on the matrix $A^HA$. Since $\kappa (A^HA) \sim \kappa (A)^2$ we get for the CGNR

$$\frac{\|r_l\|}{\|r_0\|} \leq 2 \left( \frac{\kappa - 1}{\kappa + 1} \right)^t. \quad (4.4)$$

Discretization methods for the Helmholtz equation yield in general ill-conditioned matrices. A standard approximation for the condition number of the matrix $A$ which is a result of a pure Helmholtz problem discretization [11] is

$$\kappa (A) = O (h^{-2}).$$

Using this estimate in (4.4) we get

$$\frac{\|r_l\|}{\|r_0\|} \leq 2 \left( \frac{c - h^2}{c + h^2} \right)^t \sim 2 \left( 1 - \frac{2}{c} h^2 \right)^t.$$

solving

$$2 \left( 1 - \frac{2}{c} h^2 \right)^t \leq \varepsilon$$

we get

$$l = O (h^{-2}) = O (n^2). \quad (4.5)$$

Unfortunately the results obtained in Table 2 roughly satisfy

$$l \sim O (n^{3.5}).$$

We can explain this by the fact that the condition number of the matrix for the combined problem is deeply influenced by the PML layer.
4. KRYLOV SUBSPACE METHODS FOR THE STRIP PROBLEM

<table>
<thead>
<tr>
<th>$n$</th>
<th>$N$</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>345</td>
<td>877</td>
</tr>
<tr>
<td>32</td>
<td>1457</td>
<td>12823</td>
</tr>
<tr>
<td>64</td>
<td>5985</td>
<td>116744</td>
</tr>
</tbody>
</table>

Table 2. Number of iterations required until convergence in the CGNR algorithm

The conclusion from this analysis is that the CGNR algorithm is not applicable.

We want to compare the algorithms also by calculating the cost per iteration. In this algorithm we have 2 matrix-vector multiplications, 2 inner-products and 3 linear combinations. Thus, the total cost per iteration is about $44N$.

Another approach to deal with normal equation is to solve

$$AA^Hy = b \quad x = A^Hy.$$  

This approach is called conjugate gradient normal-equation error - CGNE. The algorithm of this methods is almost the same as the CGNR algorithm. We just need to change in Algorithm 4 $\alpha_{k-1}$ and $\beta_{k-1}$ to

$$\alpha_{k-1} = \frac{\langle r_{k-1}, r_{k-1} \rangle}{\langle p_{k-1}, p_{k-1} \rangle}$$

$$\beta_{k-1} = \frac{\langle r_k, r_k \rangle}{\langle r_{k-1}, r_{k-1} \rangle}.$$  

In Table 3 we end the number of iterations using CGNE which are similar to the CGNR results.

2.3. GMRES. We now concentrate on more efficient Krylov-subspace methods for non-Hermitian matrices. In the General Minimum Residual (GMRES) method of Saad and Schultz [23] the Arnoldi process is used for construction of an orthonormal basis for the Krylov subspace
$K_n(r_0, A)$. The algorithm computes the exact solution if exact arithmetic is used. The following algorithm is based on [16, 38, 41].

Algorithm 5. GMRES

Given a square $N \times N$ nonsingular complex matrix $A$ and $b \in \mathbb{C}^N$. The following algorithm computes an approximation $x_k$ to $x$ the solution of $Ax = b$.

choose an initial guess $x_0$ and set $r_0 = b - Ax_0$, $q_1 = \frac{r_0}{\|r_0\|}$, $\xi_1 = 1$
do $k = 1, 2, \ldots$
begin - The Arnoldi algorithm for computing $q_{k+1}$, $h_{i,k}$
compute $\tilde{q} = Aq_k$
do $i = 1, k$
$\quad h_{i,j} = \langle \tilde{q}, q_j \rangle$, $\tilde{q} = \tilde{q} - h_{i,j}q_i$
enddo
$\quad h_{k+1,k} = \|\tilde{q}\|$, $q_{k+1} = \frac{\tilde{q}}{\|\tilde{q}\|}$
end of Arnoldi algorithm

do $i = 1, k - 1$
\[
\begin{pmatrix}
  h_{i,k} \\
  h_{i+1,k}
\end{pmatrix}
= \begin{pmatrix}
  c_i & s_i \\
  -s_i & c_i
\end{pmatrix}
\begin{pmatrix}
  h_{i,k} \\
  h_{i+1,k}
\end{pmatrix}
\]
enddo
\[\alpha = \frac{h_{k+1,k}}{h_{k,k}}\]
\[c_k = \frac{1}{\sqrt{1+|\alpha|^2}}, \quad s_k = \alpha c_k\]
\[
\begin{pmatrix}
\xi_k \\
\xi_{k+1}
\end{pmatrix} =
\begin{pmatrix}
c_k & s_k \\
-s_k & c_k
\end{pmatrix}
\begin{pmatrix}
\xi_k \\
0
\end{pmatrix}
\]
\[h_{k,k} = c_k h_{k,k} + s_k h_{k+1,k}, \quad h_{k+1,k} = 0\]

if \(|\xi_{k+1}| < \varepsilon\) then

solve the upper triangular system \(Hy = \|r_0\| |\xi_{k+1}|\)

where the elements of the matrix \(H\) are \(h_{i,j}\)

compute the solution \(x_k = x_0 + Qy\)

where the columns of the matrix \(Q\) are \(q_i\)

enddo

The columns of the matrix \(Q\) are the orthonormal basis for the Krylov subspace, while the matrix \(H\) is an upper Hessenberg (\(h_{i,j} = 0\) for any pair \(i, j\) such that \(i > j + 1\)). This is the full GMRES version and its major fault is that if we make \(k\) iterations we need to store the \(k\) columns of \(Q\) and the upper Hessenberg matrix \(H\). These memory requirements do not exist for the other methods we examine. Moreover, because of the Arnoldi algorithm, iteration number \(k\) involves \(O(kN)\) computations. Thus, if the total number of iterations is too large the method is not practical.

In Table 4 we end the number of iterations in this full GMRES method. The number of iterations seems much better than the normal equations results. But we have to remember that each iteration costs much more compared with the cost of the normal equation iteration. Thus, this algorithm is not applicable even for \(n = 64\) (the last iteration is \(O(1200N)\) computations!).

In order to improve the efficiency we put a limit on the number of iterations in the full GMRES - \(M\). After \(M\) iteration if we do not get
the desired solution, we start over again with the computed solution as the first guess $x_0$ for the next GMRES loop. Our experience with this idea, as seen in Table 5 indicates that in this problem we need a very large value of $M$, almost the total number of iterations. This fact makes this a noncompetitive idea.

In conclusion, because the GMRES method for non-Hermitian problems requires increasing amounts of computations and storage per iteration, we consider other methods with fixed storage. We examine two methods in the next subsections.

2.4. Biconjugate gradient. The idea of the Biconjugate gradient method (BiCG) is the same as the GMRES with the difference that instead of using the Arnoldi process to compute the orthonormal
basis for the Krylov subspace we use the two-sided Lanczos recursion algorithm (see for instance \cite{16,78}). The advantage of this approach is that it is possible to develop recursions so that it is not necessary to store the entire orthonormal basis, and the algorithm resembles the CG type algorithms. As with the restarted GMRES, because we do not use a full set of orthonormal basis, we no longer have a proof of convergence.

Algorithm 6. BiCG

Given a square $N \times N$ nonsingular complex matrix $A$ and $b \in \mathbb{C}^N$. The following algorithm computes an approximation $x_k$ to $x$ the solution of $Ax = b$.

choose an initial guess $x_0$ and set $p_0 = r_0 = b - Ax_0$

choose $\hat{r}_0$ in such a way that $\langle r_0, \hat{r}_0 \rangle \neq 0$ (for practical purposes $\hat{r}_0 = r_0$) and set $\hat{p}_0 = \hat{r}_0$

do $k = 1, 2, \ldots$

compute $Ap_k$

\[ \alpha_{k-1} = \frac{\langle r_{k-1}, \hat{r}_{k-1} \rangle}{\langle \tilde{p}_{k-1}, \tilde{p}_{k-1} \rangle} \]

\[ x_k = x_{k-1} + \alpha_{k-1}p_{k-1} \]

\[ r_k = r_{k-1} - \alpha_{k-1}Ap_{k-1} \]

if $\left| \frac{\langle r_k, \hat{r}_{k-1} \rangle}{\langle r_0, \hat{r}_0 \rangle} \right| < \varepsilon$ then the computed solution is $x_k$ and stop

compute $A^H\hat{p}_k$

\[ \tilde{r}_k = \tilde{r}_{k-1} - \alpha_{k-1}A^H\tilde{p}_k \]

\[ \beta_{k-1} = \frac{\langle r_k, \tilde{r}_k \rangle}{\langle r_{k-1}, \tilde{r}_{k-1} \rangle} \]

\[ p_k = r_k + \beta_{k-1}p_{k-1} \]

\[ \hat{p}_k = \tilde{r}_k + \beta_{k-1}\hat{p}_{k-1} \]

enddo

In each iteration we have 2 matrix-vector multiplications, 2 inner-products and 5 linear combinations. Thus the total cost per iteration
is about $48N$. In Table 6 we find the number of iterations required for convergence in the BiCG algorithm. We conclude that BiCG is much better than the normal equation algorithms. Although the BiCG requires more iterations than the GMRES, it is also superior to the latter because each iteration, after the first few, costs much less.

There are two difficulties in the BiCG algorithm that can cause it not to converge. The first one is that BiCG is based on the Lanczos method which can cause a breakdown (division by zero)\[16, 79\]. The second problem is that wide oscillations can occur in the residual norm because BiCG does not minimize a norm. In order to overcome the second problem we sometimes decrease the stopping criteria $\varepsilon$.

2.5. Quasi minimal residual. In order to overcome some of the problems in the BiCG algorithm, Freund and Nachtigal [13] suggested the Quasi Minimal Residual (QMR) Algorithm, which is also based on the two sided Lanczos algorithm but with smoother convergence in the residual norm. The following algorithm is found in [16, 83].

Algorithm 7. QMR

Given a square $N \times N$ nonsingular complex matrix $A$ and $b \in \mathbb{C}^N$. The following algorithm computes an approximation $x_k$ to $x$ the solution of $Ax = b$. 

<table>
<thead>
<tr>
<th>$n$</th>
<th>$N$</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>345</td>
<td>383</td>
</tr>
<tr>
<td>32</td>
<td>1457</td>
<td>2587</td>
</tr>
<tr>
<td>64</td>
<td>5985</td>
<td>14062</td>
</tr>
</tbody>
</table>

Table 6. Number of iterations in the BiCG algorithm
choose an initial guess \( x_0 \) and set \( r_0 = b - Ax_0 \), set \( v_1 = \frac{r_0}{\|r_0\|} \), \( \bar{v}_0 = 0 \)
choose \( \bar{r}_0 \) in such a way that \( \langle r_0, \bar{r}_0 \rangle \neq 0 \) (for practical purposes \( \bar{r}_0 = r_0 \)) and set \( \bar{v}_1 = \frac{\bar{r}_0}{\|\bar{r}_0\|} \), \( \bar{v}_0 = 0 \), \( \xi_1 = 1 \)
do \( k = 1, 2, \ldots \)
begin - The two-sided Lanczos algorithm for computing \( v_{k+1}, \bar{v}_{k+1} \), and the entries \( T_{k,k}, T_{k,k+1}, T_{k+1,k} \) of the tridiagonal matrix \( T \)
compute \( Av_k \) and \( A^H \bar{v}_k \)
\[
T_{k,k} = \langle Av_k, \bar{v}_k \rangle
\]
\[
v_{k+1} = Av_k - T_{k,k}v_k - T_{k-1,k}v_{k-1}
\]
\[
\bar{v}_{k+1} = A^H \bar{v}_k - \overline{T_{k,k}} \bar{v}_k - T_{k-1,k}\bar{v}_{k-1}
\]
\[
T_{k+1,k} = \|v_{k+1}\|, \quad v_{k+1} = \frac{v_{k+1}}{T_{k+1,k}}
\]
\[
T_{k,k+1} = \langle v_{k+1}, \bar{v}_{k+1} \rangle, \quad \bar{v}_{k+1} = \frac{\bar{v}_{k+1}}{T_{k,k+1}}
\]
end of Lanczos algorithm

if \( k > 2 \)
\[
\begin{pmatrix}
T_{k-2,k} \\
T_{k-1,k}
\end{pmatrix} =
\begin{pmatrix}
c_{k-2} & s_{k-2} \\
-s_{k-2} & c_{k-2}
\end{pmatrix}
\begin{pmatrix}
0 \\
T_{k-1,k}
\end{pmatrix}
\]
if \( k > 1 \)
\[
\begin{pmatrix}
T_{k-1,k} \\
T_{k,k}
\end{pmatrix} =
\begin{pmatrix}
c_{k-1} & s_{k-1} \\
-s_{k-1} & c_{k-1}
\end{pmatrix}
\begin{pmatrix}
T_{k-1,k} \\
T_{k,k}
\end{pmatrix}
\]
\[
\alpha = \frac{T_{k+1,k}}{T_{k,k}}
\]
\[
c_k = \frac{1}{\sqrt{1 + |\alpha|}}, \quad s_k = \alpha c_k
\]
\[
\begin{pmatrix}
\xi_k \\
\xi_{k+1}
\end{pmatrix} =
\begin{pmatrix}
c_k & s_k \\
-s_k & c_k
\end{pmatrix}
\begin{pmatrix}
\xi_k \\
0
\end{pmatrix}
\]
\[
T_{k,k} = c_k T_{k,k} + s_k T_{k+1,k}, \quad T_{k+1,k} = 0
\]
\[
p_{k-1} = \frac{v_{k-1} - T_{k-1,k}p_{k-2} - T_{k-2,k}p_{k-3}}{T_{k,k}} \quad \text{(for} \quad k \leq 3 \text{the unde ned terms are set to zero)}
\]
\[
x_k = x_{k-1} + \|r_0\| \xi_{k+1} p_{k-1}
\]
if \( |\xi_{k+1}| < \varepsilon \) then the computed solution is \( x_k \) and stop enddo
3. PRECONDITIONED ITERATIONS

<table>
<thead>
<tr>
<th>$n$</th>
<th>$N$</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>345</td>
<td>529</td>
</tr>
<tr>
<td>32</td>
<td>1457</td>
<td>5104</td>
</tr>
<tr>
<td>64</td>
<td>5985</td>
<td>43866</td>
</tr>
</tbody>
</table>

Table 7. Number of iterations in the QMR algorithm

Figure 1. The residual in logarithmic scale using the BiCG and QMR algorithms for the problem with $n = 32$

In Table 7 we found the results of the QMR algorithm. The total cost per iteration is about $70N$. Comparison of the total cost in this algorithm and the BiCG shows that QMR is less efficient than BiCG. In Figure 1 we see the wide oscillations in the residual in the BiCG algorithm compared with the smooth behavior of the residual in the QMR algorithm.

3. Preconditioned Iterations

As seen, the standard Krylov subspace methods are not efficient for the combined PML-Helmholtz problem. The main reason for this is that the convergence rate depends on the condition number of the matrix $A$ in the system $Ax = b$. Thus, in order the accelerate the
convergence we solve a system $A'x = b'$ which is equivalent to the original system and having $\kappa(A') < \kappa(A)$.

3.1. Diagonal scaling. The first approach is to apply diagonal scaling on the matrix $A$, in a way that conserves the self adjoint property of the schemes.

Algorithm 8. Diagonal Scaling

Given a square $N \times N$ Matrix $A$ with 9 diagonals at

$$A_{i,i}, A_{i,i\pm 1}, A_{i,i\pm (L_2 n - 1)}, A_{i,i\pm (L_2 n)}, A_{i,i\pm (L_2 n - 2)}$$

and a vector $b \in \mathbb{C}^N$, the following algorithm rewrites $A$ by $A'$ and $b$ by $b'$. The system $A'x = b'$ is equivalence to the system $Ax = b$ and for $1 \leq i \leq n, A'_{i,i} = 1$.

do $i = 1, N$

$$d(i) = \sqrt{A_{i,i}}$$
enddo

do $i = 1, N$

for $j \in \{i, i \pm 1, i \pm (L_2 n - 2), i \pm (L_2 n - 2), i \pm (L_2 n - 2)\} \cap \{1, 2, \ldots, N\}$

$$A_{i,j} = \frac{A_{i,j}}{d(i)d(j)}$$
$$b_j = \frac{b_j}{d(i)}$$
enddo

The resulting matrix $A'$ is (hopefully) close to the identity matrix and thus we expect $\kappa(A') < \kappa(A)$. So we expect all the Krylov subspace methods described in the last section to converge more rapidly. Table 8 shows the number of iterations required to convergence with
We clearly see, in compare with Tables 2-7, the advantage of applying this simple technique.

We can see that the number of iterations required is about $O(n^{2.3})$ for the CGNR and CGNE algorithms (compare with (4.5)), $O(n^2)$ for the GMRES and QMR and $O(n^{1.5})$ for the BiCG. Except GMRES each iteration cost is $c \cdot N$ for some constant $c$. Since $N = O(n^2)$ the total work required is $O(n^{4.3})$ for the CGNR and CGNE algorithms and similarly for the other algorithms. Since the LU algorithm requires $O(n^4)$ operations we see that the only competitive methods are the QMR and BiCG. In three dimensions, the advantage of the iterative methods increases. All these algorithms (except GMRES) are superior to LU in terms of storage requirements.

3.2. Left preconditioning. We look for the solution of the preconditioned system

$$A'x = b'$$

where

$$A' = M^{-1}Ax, \quad b' = M^{-1}b.$$  

Assuming that $M$ is a symmetric positive definite matrix which is similar in some sense to $A$. We wish that the condition number of $A'$ is
small and thus the iterative methods described should converge faster.

Rewriting Algorithm 4 we get

Algorithm 9. Left Preconditioned CGNR (LPCGNR)

Given a square $N \times N$ nonsingular complex matrix $A$, $b \in \mathbb{C}^N$ and a symmetric positive definite $N \times N$ matrix $M$. The following algorithm computes an approximation $x_k$ to $x$ the solution of $Ax = b$.

choose an initial guess $x_0$ and set $r_0 = b - Ax_0$
compute $A^H r_0$, $M^{-1} A^H r_0$ and set $p_0 = M^{-1} A^H r_0$
do $k = 1, 2, \ldots$

compute $A p_{k-1}$

\[ \alpha_{k-1} = \frac{\langle M^{-1} A^H r_{k-1}, A^H r_{k-1} \rangle}{\langle A p_{k-1}, A p_{k-1} \rangle} \]

\[ x_k = x_{k-1} + \alpha_{k-1} p_{k-1} \]

\[ r_k = r_{k-1} - \alpha_{k-1} A p_{k-1} \]

if $\|r_k\| < \varepsilon$ then the computed solution is $x_k$ and stop
compute $A^H r_k$ and $M^{-1} A^H r_k$

\[ \beta_{k-1} = \frac{\langle M^{-1} A^H r_{k-1}, A^H r_k \rangle}{\langle M^{-1} A^H r_{k-1}, A^H r_{k-1} \rangle} \]

\[ p_k = M^{-1} A^H r_k + \beta_{k-1} p_{k-1} \]
enddo

A similar approach can be made with right preconditioning, with no significant difference in the rate of convergence [22, 255]. We will use only versions of left preconditioners. A similar left-preconditioning can be applied on the CGNE, GMRES, BiCG and QMR algorithms (getting LPCGNR, LPGMRES, LPBiCG and LPQMR). In all these algorithms we solve (4.6) with (4.7). Versions of these algorithms is to be found in [22] and [2].
3. PRECONDITIONED ITERATIONS

3.3. Split preconditioning. Because $M$ is a real symmetric positive definite matrix, it has the decomposition

$$M = GG^T$$

(4.8)

We then solve the preconditioned system

$$A'x' = b'$$

where

$$A' = G^{-1}AG^{-T}, \quad x' = G^T x, \quad b' = G^{-1}b.$$

Again we apply CGNR on this problem resulting in another preconditioned CGNR algorithm:

Algorithm 10. Split Preconditioned CGNR (SPCGNR)

Given a square $N \times N$ nonsingular complex matrix $A$, $b \in \mathbb{C}^N$ and a symmetric positive definite $N \times N$ matrix $M$. The following algorithm computes an approximation $x_k$ to $x$ the solution of $Ax = b$.

choose an initial guess $x_0$ and set $r_0 = b - Ax_0$
compute $A^HM^{-1}r_0$, $M^{-1}A^HM^{-1}r_0$ and set $p_0 = M^{-1}A^HM^{-1}r_0$
do $k = 1, 2, \ldots$

compute $Ap_{k-1}$

$$\alpha_{k-1} = \frac{(G^{-1}A^HM^{-1}r_{k-1}, G^{-1}A^HM^{-1}r_{k-1})}{(G^{-1}Ap_{k-1}, G^{-1}Ap_{k-1})} = \frac{(M^{-1}A^HM^{-1}r_{k-1}, A^HM^{-1}r_{k-1})}{(M^{-1}Ap_{k-1}, Ap_{k-1})}$$

$$x_k = x_{k-1} + \alpha_{k-1}p_{k-1}$$

$$r_k = r_{k-1} - \alpha_{k-1}Ap_{k-1}$$

if $\frac{\|r_k\|}{\|r_0\|} < \varepsilon$ then the computed solution is $x_k$ and stop

compute $A^HM^{-1}r_k$ and $M^{-1}A^HM^{-1}r_k$

$$\beta_{k-1} = \frac{(G^{-1}A^HM^{-1}r_k, G^{-1}A^HM^{-1}r_k)}{(G^{-1}Ap_{k-1}, G^{-1}Ap_{k-1})} = \frac{(M^{-1}A^HM^{-1}r_k, A^HM^{-1}r_k)}{(M^{-1}Ap_{k-1}, Ap_{k-1})}$$

$$p_k = M^{-1}A^HM^{-1}r_k + \beta_{k-1}p_{k-1}$$
endo
4. Choosing the Preconditioned Matrix

4.1. Inverse Laplacian preconditioner. For the pure Helmholtz equation one of the standard preconditioners is the inverse of the discretize Laplacian operator with the standard PT approximation [3]. We do not actually compute the inverse of the operator, but instead if we need to calculate \( z = M^{-1}y \) as the solution of

\[
Mz = y
\]

we apply few sweeps of a simple iterative solver e.g. SSOR and damped Jacobi as an approximation to \( z \).

4.1.1. Symmetric successive over relaxation (SSOR). Each sweep of SSOR includes one sweep of SOR and one of backward SOR with the relaxation constant \( 0 \leq \omega \leq 2 \), as can be seen in the following algorithm.

Algorithm 11. SSOR sweep

\[
x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left( y_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right) + (1 - \omega) x_i^{(k)}
\]

endo

do \( i = N, 1, -1 \)

\[
x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left( y_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right) + (1 - \omega) x_i^{(k)}
\]

endo

Bayliss, Goldstein and Turkel showed [3] that this approach is efficient for the pure Helmholtz equation coupled with Dirichlet or Neumann boundary conditions. In theory, if we apply the exact inverse of the Laplacian, the algorithm results (in the SP CGNR algorithm) in a preconditioned method with a rate of convergence of \( O(1) \) iterations for small values of \( k \). This is only true to the pure Helmholtz equation.
4. CHOOSING THE PRECONDITIONED MATRIX

<table>
<thead>
<tr>
<th></th>
<th>$n$</th>
<th>CGNR</th>
<th>LPCGNR</th>
<th>SPCGNR</th>
<th>CGNR + diagonal scaling</th>
<th>LPCGNR + diagonal scaling</th>
<th>SPCGNR + diagonal scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>877</td>
<td>604</td>
<td>382</td>
<td>351</td>
<td>288</td>
<td>313</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>12823</td>
<td>5434</td>
<td>1916</td>
<td>1732</td>
<td>1162</td>
<td>1389</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>116744</td>
<td>34901</td>
<td>10693</td>
<td>8646</td>
<td>5837</td>
<td>8926</td>
<td></td>
</tr>
</tbody>
</table>

Table 9. Number of iterations in the CGNR and preconditioned CGNR with one sweep of SSOR with $\omega = 1.72$ applied on the Laplacian operator

It is shown [25] that if we choose one sweep of SOR as the preconditioner $G$ in (4.8) (a choice that means that $M$ is one sweep of SSOR), the SPCGNR algorithm has the number of iterations proportional to $O(n)$. It is also shown there that the PT approximation is sufficient and we do not need high order approximations for the inverse of the Laplacian. We save computations by using the lower order PT scheme in the preconditioner.

Table 9 shows the number of iterations in CGNR, LPCGNR and SPCGNR, with and without diagonal scaling. In all the computations we take one sweep of SSOR for the preconditioner. We set $a_{ii} = 1$, so in each SSOR sweep we have about $14N$ operations and thus the total number of operations in the LPCGNR is about $1.3$ as much as the standard CGNR iteration. With this in mind we find that the preferred strategy is LPCGNR with diagonal scaling. If we do not do diagonal scaling the SPCGNR is a much better scheme than the LPCGNR. Doing more sweeps of SSOR adds much computational work without significantly decreasing the number of iterations (if at all) as seen in Table 10. Table 11 shows the dependence of the operator $\omega$ in
4. Krylov Subspace Methods for the Strip Problem

<table>
<thead>
<tr>
<th>( n )</th>
<th>1 SS OR sweep</th>
<th>2 SS OR sweeps</th>
<th>5 SS OR sweeps</th>
<th>50 SS OR sweeps</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>288</td>
<td>289</td>
<td>293</td>
<td>255</td>
</tr>
<tr>
<td>32</td>
<td>1162</td>
<td>1172</td>
<td>1155</td>
<td>1155</td>
</tr>
<tr>
<td>64</td>
<td>5837</td>
<td>5830</td>
<td>5846</td>
<td>6312</td>
</tr>
</tbody>
</table>

Table 10. Number of iterations in the LPC GNR with diagonal scaling using SS OR with \( \omega = 1.72 \) applied on the Laplacian operator

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \omega = 1.5 )</th>
<th>( \omega = 1.72 )</th>
<th>( \omega = 1.9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>297</td>
<td>288</td>
<td>303</td>
</tr>
<tr>
<td>32</td>
<td>1236</td>
<td>1162</td>
<td>1257</td>
</tr>
<tr>
<td>64</td>
<td>6641</td>
<td>5837</td>
<td>7000</td>
</tr>
</tbody>
</table>

Table 11. Number of iterations in the LPC GNR with diagonal scaling using one sweep of SS OR with various values of \( \omega \) applied on the Laplacian operator

the SS OR algorithm. For SS OR the rate of convergence depends on the value of the relaxation constant \( \omega \). Because this algorithm is not optimal a search for an optimal value of \( \omega \) is omitted.
4. CHOOSING THE PRECONDITIONED MATRIX

<table>
<thead>
<tr>
<th>n</th>
<th>CGNR</th>
<th>LPCGNR</th>
<th>SP CGNR</th>
<th>CGNR +</th>
<th>LPCGNR +</th>
<th>SP CGNR +</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>diagonal</td>
<td>diagonal</td>
<td>diagonal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>scaling</td>
<td>scaling</td>
<td>scaling</td>
</tr>
<tr>
<td>16</td>
<td>877</td>
<td>640</td>
<td>538</td>
<td>351</td>
<td>270</td>
<td>227</td>
</tr>
<tr>
<td>32</td>
<td>12823</td>
<td>9245</td>
<td>6937</td>
<td>1732</td>
<td>1389</td>
<td>1166</td>
</tr>
<tr>
<td>64</td>
<td>116744</td>
<td>86895</td>
<td>65001</td>
<td>8646</td>
<td>7077</td>
<td>5994</td>
</tr>
</tbody>
</table>

Table 12. Number of iterations in the CGNR and preconditioned CGNR with two sweeps of DJ with $\omega = 0.7$

applied on the Laplacian operator

4.1.2. Damped Jacobi (DJ). We want to compare the SSOR preconditioner with the damped Jacobi preconditioner (DJ). Each sweep of DJ consists of the Jacobi iteration linearly interpolated with the last iteration. This yields

Algorithm 12. DJ sweep

do $i = 1, N$

$$x_i^{[k+1]} = \omega \left( y_i - \sum_{j=1}^{i-1} a_{ij} x_j^{[k]} - \sum_{j=i+1}^{N} a_{ij} x_j^{[k]} \right) + (1 - \omega) x_i^{[k]}$$
enddo

To keep the cost of the two methods comparable we take 2 sweeps of DJ for every sweep of SSOR. Table 12 shows the number of iterations using the DJ preconditioner applied on the Laplacian operator with $\omega = 0.7$. We see again that the main factor is the diagonal scaling and the use of this preconditioner reduces the number of iterations, but each iteration costs more. The fact that we need fewer iterations in the SP CGNR in compare to LPCGNR is also deceiving because in each iteration of the SP CGNR we use twice the number of DJ sweeps as the number of sweeps in the LPCGNR iteration. Tables 13 and 14 shows that increasing the number of DJ sweeps decreases the number
4. KRYLOV SUBSPACE METHODS FOR THE STRIP PROBLEM

<table>
<thead>
<tr>
<th></th>
<th>2 DJ sweeps</th>
<th>4 DJ sweeps</th>
<th>10 DJ sweeps</th>
<th>100 DJ sweeps</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>270</td>
<td>228</td>
<td>210</td>
<td>201</td>
</tr>
<tr>
<td>32</td>
<td>1389</td>
<td>1123</td>
<td>870</td>
<td>721</td>
</tr>
<tr>
<td>64</td>
<td>7077</td>
<td>5684</td>
<td>4174</td>
<td>3070</td>
</tr>
</tbody>
</table>

Table 13. Number of iterations in the LPCG NR with diagonal scaling using DJ with \( \omega = 0.7 \) applied on the Laplacian operator

<table>
<thead>
<tr>
<th></th>
<th>2 DJ sweeps</th>
<th>5 DJ sweeps</th>
<th>50 DJ sweeps</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>227</td>
<td>174</td>
<td>159</td>
</tr>
<tr>
<td>32</td>
<td>1166</td>
<td>704</td>
<td>612</td>
</tr>
<tr>
<td>64</td>
<td>5994</td>
<td>3381</td>
<td>3388</td>
</tr>
</tbody>
</table>

Table 14. Number of iterations in the SPCG NR with diagonal scaling using DJ with \( \omega = 0.7 \) applied on the Laplacian operator

of iterations, unlike the SSOR results. This fact suggests the question what is the optimal number DJ sweeps in order to decrease the total cost of the algorithm. If we have \( s \) sweeps and \( i \) iterations the total cost is about \((44 + 7s) \cdot iN\) in the LPCG NR and \((44 + 14s) \cdot iN\) for the SPCG NR. Figure 2 shows that for \( n = 32 \) the optimal algorithm is SPCG NR with \( s = 4 \). It is also fair to apply the LPCG NR with \( s = 3 \). Table 15 shows the number of iterations when we change the value of \( \omega \) in the DJ algorithm. For this problem the optimal value of
4. CHOOSING THE PRECONDITIONED MATRIX

Figure 2. Total cost of the LPCG NR and SPCG NR algorithms depending on the number of DJ sweeps, for $n = 32$.

<table>
<thead>
<tr>
<th></th>
<th>$\omega = 0.5$</th>
<th>$\omega = 0.7$</th>
<th>$\omega = 0.9$</th>
<th>$\omega = 1.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>246</td>
<td>288</td>
<td>218</td>
<td>230</td>
</tr>
<tr>
<td>32</td>
<td>1237</td>
<td>1123</td>
<td>1035</td>
<td>1012</td>
</tr>
<tr>
<td>64</td>
<td>6317</td>
<td>5684</td>
<td>5146</td>
<td>4966</td>
</tr>
</tbody>
</table>

Table 15. Number of iterations in the LPCGNR with diagonal scaling using three sweeps of DJ with various values of $\omega$ applied on the Laplacian operator

$\omega$ is 1.0 i.e. the standard Jacobi. When we use other algorithms e.g. preconditioned BiCG, other values of $\omega$ are optimal.

4.2. Symmetric inverse PML-Helmholtz like preconditioner.

The major difficulty of the inverse of the Laplacian operator as a preconditioner is that we have a PML layer which introduces a different equation. $k = 0$ in the PML does not give the Laplace equation. Thus, we wish to construct a preconditioner, that will act as the inverse Laplacian operator in the interior, and behave as an approximate inverse of the equation in the PML. If the preconditioner is $\tilde{M}$ we do not actually
calculate the inverse $M^{-1}$ but we approximate the solution of $My = z$, by applying few sweeps of SSOR or DJ.

In the PML we cannot choose $k = 0$ because of the deñition of the function $S_x (3.8)$. Thus, to end such a preconditioner $\hat{M}$ we apply the standard approximation of the combined problem with a small value of $k$ which we denote $\tilde{k}$

$$\alpha D_x (Au_x)_j + \frac{1 - \alpha}{2} (D_x (Au_x)_{j+1} + D_x (Au_x)_{j-1}) +$$
$$\alpha D_y (Bu_y)_i + \frac{1 - \alpha}{2} (D_y (Au_y)_{i+1} + D_y (Au_y)_{i-1}) +$$
$$\left(1 - 4\beta_s - 4\beta_c\right) C_{i,j} u_{i,j}$$

$$+ \beta_s \begin{pmatrix} C_{i+1,j} u_{i+1,j} + C_{i-1,j} u_{i-1,j} \\ C_{i,j+1} u_{i,j+1} + C_{i,j-1} u_{i,j-1} \end{pmatrix} +$$
$$\beta_s \begin{pmatrix} C_{i+1,j+1} u_{i+1,j+1} + C_{i-1,j+1} u_{i-1,j+1} \\ C_{i+1,j-1} u_{i+1,j-1} + C_{i-1,j-1} u_{i-1,j-1} \end{pmatrix}.$$  

Where $A = \frac{1}{S_x}$, $B = S_x$, $C = \tilde{k}^2 S_x$. The resulting matrix $\hat{M}$ is generally complex and cannot be applied as a preconditioner for two reasons: it is not symmetric and it is not real and positive deñite. It does have these two properties inside the interior, but not inside the PML.

To overcome these two problems we set the PT values: $\alpha = 1$, $\beta_s = \beta_c = 0$ in the preconditioner. This choice makes the approximation complex-symmetric, it is also very useful because it involves only 5 unknowns in each equation instead of 9. This choice should be useful in terms of computation cost as seen in the last subsection. To change the matrix to a real symmetric positive deñite matrix we make the
4. CHOOSING THE PRECONDITIONED MATRIX

<table>
<thead>
<tr>
<th>n</th>
<th>$\tilde{\sigma} = 10$</th>
<th>$\tilde{\sigma} = 10$</th>
<th>$\tilde{\sigma} = 1$</th>
<th>$\tilde{\sigma} = 10$</th>
<th>$\tilde{\sigma} = 100$</th>
<th>$\tilde{\sigma} = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\tilde{\rho} = 1$</td>
<td>$\tilde{\rho} = 2$</td>
<td>$\tilde{\rho} = 4$</td>
<td>$\tilde{\rho} = 4$</td>
<td>$\tilde{\rho} = 4$</td>
<td>$\tilde{\rho} = 6$</td>
</tr>
<tr>
<td>16</td>
<td>211</td>
<td>206</td>
<td>208</td>
<td>200</td>
<td>204</td>
<td>204</td>
</tr>
<tr>
<td>32</td>
<td>832</td>
<td>811</td>
<td>927</td>
<td>807</td>
<td>807</td>
<td>871</td>
</tr>
<tr>
<td>64</td>
<td>4022</td>
<td>3911</td>
<td>4683</td>
<td>3766</td>
<td>3857</td>
<td>4421</td>
</tr>
</tbody>
</table>

Table 16. Number of iterations in the LPC GNR with diagonal scaling with 4 sweeps of DJ with $\omega = 0.7$ applied on the combined preconditioner with $k = 0.1$

change:

\[
M_{i,j} = \begin{cases} 
  |\hat{M}_{i,j}| & i = j \\
  -|\hat{M}_{i,j}| & i \neq j 
\end{cases} \quad \text{for every } 1 \leq i, j \leq N.
\]

Inside the interior $M_{i,j} = \hat{M}_{i,j}$. Because $\tilde{k}$ is small in the interior, the approximation is a small perturbation of the Laplacian. We call this preconditioner the combined preconditioner. Note, that as in the pure Laplacian operator preconditioner, this preconditioner is second order accurate even when the scheme inside the interior is higher order accurate.

Table 16 shows the number of iterations in the LPC GNR with the new combined preconditioner. We choose $\tilde{k} = 0.1$, and various values of $\tilde{\sigma}$ and $\tilde{\rho}$ inside the PML (as a preconditioner). It seems that the best value of $\tilde{\rho}$ is the same $p$ we use for the approximation of the problem and the value of $\tilde{\sigma}$ should be small, but not too small. A comparison of the results with the results of Table 13 shows that this preconditioner is better than the inverse of the Laplacian.
4. KRYLOV SUBSPACE METHODS FOR THE STRIP PROBLEM

<table>
<thead>
<tr>
<th>n</th>
<th>LPBiCG</th>
<th>LPQMR</th>
<th>LPGMRES</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>113</td>
<td>121</td>
<td>84</td>
</tr>
<tr>
<td>32</td>
<td>306</td>
<td>350</td>
<td>192</td>
</tr>
<tr>
<td>64</td>
<td>1269</td>
<td>1758</td>
<td>527</td>
</tr>
</tbody>
</table>

Table 17. Number of iterations in the LPBiCG, LPQMR, LPGMRES with diagonal scaling, applied with 4 sweeps of DJ with $\omega = 0.7$ applied on the combined preconditioner with $\tilde{k} = 0.1$, $\tilde{p} = 4$ and $\tilde{\sigma} = 10$

5. Choosing the Optimal Algorithm

In this section we choose the best algorithm to use with the new preconditioner which based on the combined problem. As seen, the normal-equation solvers are usually inferior in compare with the BiCG QMR and GMRES algorithms. In these algorithms we have just the left preconditioner (and a similar right one), which we build in the same way as we build LPCGNR in Algorithm 9. The resulting are the LPBiCG, LPQMR and LPGMRES.

Table 17 shows the number of iterations in these three preconditioned algorithms. As a basic test preconditioner we set $\tilde{k} = 0.1$, $\tilde{p} = 4$ and $\tilde{\sigma} = 10$, solved with 4 sweeps of DJ with $\omega = 0.7$. In all the algorithms we used diagonal scaling. Comparison with Table 16 shows that these algorithms are superior to the LPCGNR scheme. The advantage of using the preconditioner is emphasized when we compare these results to the results in Table 8. Although the number of iterations in the LPGMRES is less than the other it is still inferior to them in respect to storage requirements and even CPU time in large matrices. Increasing to $n = 128$ makes the full GMRES inapplicable. Thus, we can recommend this algorithm only for medium size matrices (up to $n = 64$).
5. CHOOSING THE OPTIMAL ALGORITHM

Figure 3. The residual in logarithmic scale using the LP-BiCG and LPQMR algorithms for the problem with $n = 32$

Figure 3 shows in logarithmic scale the residual in the LPBiCG and LPQMR, for the same problem solved in Table 17, for $n = 32$. If we accept the wide oscillations in the residual norm in the LPBiCG algorithm we recommend it over the LPQMR. The latter requires more iterations and more operations per iteration. Next we examine how the parameters of the initial problem and the preconditioner influence the total cost of the algorithm.

5.1. SSOR or DJ. Tables 18 and 19 shows the number of iterations for the LPBiCG algorithm with SSOR with 1 sweep and DJ with 4 sweeps for various values of the damping coefficient $\omega$. We see that the optimal value of $\omega$ depends on the preconditioner (compare with Tables 11 and 15). We use from now on $\omega = 1.2$ for SSOR and $\omega = 0.7$ for DJ.

Tables 20 and 21 shows the number of iterations in respect to the number of SSOR or DJ sweeps - $s$. In this case the optimal number of iterations for the SSOR is $s = 1$ while for the DJ is $s = 2$, a choice
Table 18. Number of iterations in the LPBICG with diagonal scaling with 1 sweep of SSOR applied on the combined preconditioner with \( \tilde{k} = 0.1 \tilde{\sigma} = 10 \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \omega = 0.9 )</th>
<th>( \omega = 1.1 )</th>
<th>( \omega = 1.2 )</th>
<th>( \omega = 1.3 )</th>
<th>( \omega = 1.5 )</th>
<th>( \omega = 1.7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>178</td>
<td>177</td>
<td>175</td>
<td>185</td>
<td>198</td>
<td>215</td>
</tr>
<tr>
<td>32</td>
<td>547</td>
<td>510</td>
<td>512</td>
<td>535</td>
<td>550</td>
<td>619</td>
</tr>
<tr>
<td>64</td>
<td>2259</td>
<td>2253</td>
<td>2236</td>
<td>2238</td>
<td>2471</td>
<td>3547</td>
</tr>
</tbody>
</table>

Table 19. Number of iterations in the LPBICG with diagonal scaling with 4 sweeps of DJ applied on the combined preconditioner with \( \tilde{k} = 0.1 \tilde{\sigma} = 10 \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \omega = 0.1 )</th>
<th>( \omega = 0.3 )</th>
<th>( \omega = 0.5 )</th>
<th>( \omega = 0.7 )</th>
<th>( \omega = 0.8 )</th>
<th>( \omega = 1.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>157</td>
<td>132</td>
<td>120</td>
<td>113</td>
<td>110</td>
<td>145</td>
</tr>
<tr>
<td>32</td>
<td>435</td>
<td>356</td>
<td>323</td>
<td>306</td>
<td>301</td>
<td>551</td>
</tr>
<tr>
<td>64</td>
<td>1743</td>
<td>1488</td>
<td>1299</td>
<td>1269</td>
<td>1282</td>
<td>4057</td>
</tr>
</tbody>
</table>

Table 20. Number of iterations in the LPBICG with diagonal scaling solved with \( s \) sweeps of SSOR with \( \omega = 1.2 \) applied on the combined preconditioner with \( \tilde{k} = 0.1 \tilde{\sigma} = 10 \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( s = 1 )</th>
<th>( s = 2 )</th>
<th>( s = 4 )</th>
<th>( s = 20 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>175</td>
<td>180</td>
<td>185</td>
<td>219</td>
</tr>
<tr>
<td>32</td>
<td>512</td>
<td>540</td>
<td>634</td>
<td>550</td>
</tr>
<tr>
<td>64</td>
<td>2236</td>
<td>2523</td>
<td>3679</td>
<td>&gt;10000</td>
</tr>
</tbody>
</table>

which is equal to one sweep of SSOR in terms of CPU cost. We prefer in the DJ algorithm \( s = 2 \) because it means less cost as can be seen in Figure 4. Between SSOR with \( s = 1 \) and DJ with \( s = 2 \) we choose to
5. CHOOSING THE OPTIMAL ALGORITHM

<table>
<thead>
<tr>
<th>n</th>
<th>s = 2</th>
<th>s = 4</th>
<th>s = 6</th>
<th>s = 12</th>
<th>s = 40</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>124</td>
<td>113</td>
<td>114</td>
<td>112</td>
<td>125</td>
</tr>
<tr>
<td>32</td>
<td>344</td>
<td>306</td>
<td>308</td>
<td>327</td>
<td>392</td>
</tr>
<tr>
<td>64</td>
<td>1421</td>
<td>1269</td>
<td>1256</td>
<td>1298</td>
<td>2568</td>
</tr>
</tbody>
</table>

Table 21. Number of iterations in the LPBiCG with diagonal scaling solved with s sweeps of DJ with \( \omega = 0.7 \) applied on the combined preconditioner with \( \tilde{k} = 0.1 \) \( \tilde{\sigma} = 10 \)

Figure 4. Total cost of the LPBiCG algorithm in respect to the number of DJ sweeps, for \( n = 32 \).

use the DJ because it means less iterations. We use this choice from now on.

5.2. The parameter of the Helmholtz equation \( \tilde{k} \). Tables 22 and 23 demonstrate the fact that as \( \tilde{k} \) grows we need more iterations. Comparison between the two tables demonstrates that as the value of \( \tilde{k} \) increases we benefit more from the use of the preconditioner. We can see by comparing the results in these tables that the number of iterations is reduced by about 27% if we use the preconditioned version.
for all values of $k$. We stress that the decrease in the number of required iterations (percentage wise) is independent of $k$.

5.3. Number of iterations for fixed accuracy. In the last subsection we fixed the value of $h$ and changed $k$. We also want to examine how the value of $k$ influences the rate of convergence when we keep the accuracy fixed. Here, when we increase $k$ we also decrease $h = \frac{\pi}{k}$ such that the error in $l_\infty$ norm is constant.

Tables 24 - 26 support the fact that for better accuracy we need one grids that lead to large linear systems and we need more iterations in order to solve them. A simple power law for the number of iterations as a function of $k$ was not found.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$k = 2.5$</th>
<th>$k = 4$</th>
<th>$k = 7$</th>
<th>$k = 12$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>111</td>
<td>124</td>
<td>387</td>
<td>520</td>
</tr>
<tr>
<td>32</td>
<td>308</td>
<td>344</td>
<td>624</td>
<td>1407</td>
</tr>
<tr>
<td>64</td>
<td>1476</td>
<td>1421</td>
<td>2383</td>
<td>2870</td>
</tr>
</tbody>
</table>

Table 22. Number of iterations in the LPBiCG with diagonal scaling with 2 sweeps of DJ with $\omega = 0.7$ applied on the combined preconditioner.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$k = 2.5$</th>
<th>$k = 4$</th>
<th>$k = 7$</th>
<th>$k = 12$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>141</td>
<td>171</td>
<td>473</td>
<td>507</td>
</tr>
<tr>
<td>32</td>
<td>431</td>
<td>501</td>
<td>870</td>
<td>1913</td>
</tr>
<tr>
<td>64</td>
<td>1979</td>
<td>1955</td>
<td>3292</td>
<td>3925</td>
</tr>
</tbody>
</table>

Table 23. Number of iterations in the BiCG with diagonal scaling.
5. CHOOSING THE OPTIMAL ALGORITHM

<table>
<thead>
<tr>
<th>$k$</th>
<th>$n$</th>
<th># of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>16</td>
<td>124</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>407</td>
</tr>
<tr>
<td>12</td>
<td>26</td>
<td>2897</td>
</tr>
</tbody>
</table>

Table 24. Number of iterations in the LPBiCG with diagonal scaling with 2 sweeps of DJ with $\omega = 0.7$ applied on the combined preconditioner with $\sigma_{\text{ex}}\cdot err \approx 0.13$

<table>
<thead>
<tr>
<th>$k$</th>
<th>$n$</th>
<th># of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>24</td>
<td>220</td>
</tr>
<tr>
<td>8</td>
<td>29</td>
<td>524</td>
</tr>
<tr>
<td>12</td>
<td>34</td>
<td>1406</td>
</tr>
</tbody>
</table>

Table 25. Number of iterations in the LPBiCG with diagonal scaling with 2 sweeps of DJ with $\omega = 0.7$ applied on the combined preconditioner with $\sigma_{\text{ex}}\cdot err \approx 1.1 \times 10^{-2}$

<table>
<thead>
<tr>
<th>$k$</th>
<th>$n$</th>
<th># of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>32</td>
<td>344</td>
</tr>
<tr>
<td>8</td>
<td>38</td>
<td>845</td>
</tr>
<tr>
<td>12</td>
<td>46</td>
<td>1802</td>
</tr>
</tbody>
</table>

Table 26. Number of iterations in the LPBiCG with diagonal scaling with 2 sweeps of DJ with $\omega = 0.7$ applied on the combined preconditioner with $\sigma_{\text{ex}}\cdot err \approx 8.3 \times 10^{-4}$

5.4. The parameters inside the PML. The following results show how the parameters of the function $\sigma_x$ inside the PML influence the number of iterations of the LPBiCG algorithm.
4. KRYLOV SUBSPACE METHODS FOR THE STRIP PROBLEM

<table>
<thead>
<tr>
<th>n</th>
<th>( L_1 = 1.25 )</th>
<th>( L_1 = 1.5 )</th>
<th>( L_1 = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>124</td>
<td>142</td>
<td>180</td>
</tr>
<tr>
<td>32</td>
<td>344</td>
<td>389</td>
<td>474</td>
</tr>
<tr>
<td>64</td>
<td>1421</td>
<td>1637</td>
<td>2036</td>
</tr>
</tbody>
</table>

Table 27. Number of iterations in respect to \( L_1 \) solved with LPBICG with diagonal scaling

\[
\begin{array}{c|cccc}
\backslash \n \ \ & \ L_2 - L_1 \ & \ \frac{\pi}{2} \ & \ \frac{\pi}{4} \ & \ \frac{\pi}{8} \\
\hline
n   & \backslash & \ & \ & \\
16  & 158        & 124              & 114              \\
32  & 498        & 344              & 284              \\
64  & 2485       & 1421             & 863              \\
\end{array}
\]

Table 28. Number of iterations in respect to the width of the PML \( L_2 - L_1 \) solved with LPBICG with diagonal scaling

Value of \( L_1 \): Table 27 shows that as \( L_1 \) increases, keeping the value of \( L_2 - L_1 \), the size of the matrix increases and so does the number of iterations.

Width of the PML, \( L_2 - L_1 \): Table 28 shows the number of iterations for different sizes of the PML. The total size of the matrix is kept the same. We see that a thin PML means better performance. One explanation of this phenomena is that even with the combined preconditioner we do not imitate perfectly the behavior inside the PML and in the interior we succeed better. As mentioned in Chapter 3 we cannot make our PML too thin as we lose accuracy.

Value of \( \mathcal{P} \): Table 29 shows the number of iterations as we change the parameter \( p \). In all these results we set the parameter \( \tilde{p} = p \)
6. Conclusions

\[ \begin{array}{|c|c|c|c|c|}
\hline
n & p = 2 & p = 4 & p = 6 & p = 10 \\
\hline
16 & 134 & 124 & 125 & 113 \\
32 & 347 & 344 & 321 & 290 \\
64 & 1651 & 1421 & 1292 & 1037 \\
\hline
\end{array} \]

Table 29. Number of iterations in respect to \( \tilde{p} \) solved with LPBIC G with diagonal scaling

\[ \begin{array}{|c|c|c|c|c|}
\hline
n & \sigma = 50 & \sigma = 100 & \sigma = 300 & \sigma = 1000 \\
\hline
16 & 124 & 125 & 124 & 139 \\
32 & 339 & 352 & 344 & 351 \\
64 & 1370 & 1476 & 1421 & 1372 \\
\hline
\end{array} \]

Table 30. Number of iterations in respect to \( \sigma \) solved with LPBIC G with diagonal scaling

(\( \tilde{p} \) is the corresponding parameter in the PML of the preconditioned matrix). We conclude that the rate of convergence increases as the value of \( p \) increases but not dramatically. The value of \( p \) is a key factor for accuracy, so we recommend to choose \( p = 4 \) for the \( O(h^6) \) scheme and \( p = 2 \) for the \( O(h^4) \) scheme.

Value of \( \sigma \): Table 30 shows that the value of \( \sigma \) does not influence the rate of convergence in any consistent manner. Hence, we choose \( \sigma \) as small as possible so as to maintain the error estimate (3.38).

6. Conclusions

We described how to construct a PML Layer to the Helmholtz equation and developed high-order finite difference schemes for the combined PML-Helmholtz equation. In this section we organize the results and give our recommendations.
We recommend to use a high-order scheme which is in divergence form. We have two possibilities, the HO-6 (for constant $k$) and the HO-4 schemes. In chapter 2 we describe how the second order approximation in the PML layer switches automatically to the high order approximation.

In the PML layer we solve

$$\frac{\partial}{\partial x} \left( \frac{S_y}{S_x} \frac{\partial}{\partial x} u \right) + \frac{\partial}{\partial y} \left( \frac{S_x}{S_y} \frac{\partial}{\partial y} u \right) + k^2 S_x S_y u = 0$$

where

$$S_x = 1 + \frac{\sigma_x}{ik}, \quad S_y = 1 + \frac{\sigma_y}{ik}$$

and

$$\sigma_x(x) = \sigma \left( \frac{x - L_1}{L_2 - L_1} \right)^p$$

For a waveguide $\sigma_y(y) = 0$. $L_1$ is the width of the physical interior. Inside the PML we control the parameters $p$, $\sigma$, and the width of the PML $L_2 - L_1$.

We concentrate in chapter 3 on traveling waves which is the more difficult case to solve. In this case the limiting factor of the accuracy is

$$2e^{-\frac{3\sqrt{1-(L_2-L_1)} \varepsilon}{p+1}}.$$

This is an estimate of the accuracy that can be achieved by applying the PML. The value of $\varepsilon = \left( \frac{m}{k} \right)^2$ where $m$ is the input mode. Hence, for general boundary conditions it cannot be controlled. However, we can control the other parameters.

In order to solve the linear system solved with the use of the high-order schemes we apply Krylov subspace iterative methods. Besides
the accuracy of the schemes, the parameters inside the PML influence also the rate of convergence of the iterative methods.

We recommend to use the LPBiCG method which is a left pre-conditioned BiCG. We specially tailored the pre-conditioning for the combined problem. The specific details of this method is described in the previous section. The parameters should be chosen in such a way:

- $L_1$ - As small as the physical problem permits, because it does not influence the accuracy but has significant impact on the size of the linear system.
- $L_2 - L_1$ - Our desire is to set the width of the PML as thin as possible in order to decrease the size of the linear system. Numerical tests show that we cannot make the layer too thin. For the grids used in this study one can use 16 grid - points inside the PML and still get the high order accuracy. However, if the grid is refined than one would need more points in the PML layer.
- $p$ should be set to 4 in the HO-6 scheme and 2 in the HO-4 scheme.
- $\sigma$ should be set to the lowest value which maintains the desired accuracy 
  \[ 2e^{-\frac{2\pi \sqrt{\pi (L_2 - L_1)}}{L_2 - L_1}}. \]
Bibliography


